

Deutsch



BS2000/OSD

BS2000/OSD

Performance-Handbuch

Benutzerhandbuch

Stand der Beschreibung:

BS2000/OSD-BC V9.0

OSD/XC V9.0

X2000 V5.3

Ausgabe März 2013

Kritik... Anregungen... Korrekturen...

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an manuals@ts.fujitsu.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2008

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2008 erfüllt.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright und Handelsmarken

Copyright © Fujitsu Technology Solutions GmbH 2013.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

EMC²®, Symmetrix®, CLARiiON CX, SRDF™, TimeFinder™, SnapView™, Enginuity™ und EMC Ionix ControlCenter™ sind Warenzeichen oder eingetragene Warenzeichen der Firma EMC² Corporation, Hopkinton/MA (USA).

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.

Inhalt

1	Einleitung	9
1.1	Zielsetzung und Zielgruppen des Handbuchs	10
1.2	Konzept des Handbuchs	10
1.3	Änderungen gegenüber dem Vorgänger-Handbuch	12
1.4	Darstellungsmittel	13
2	Leistungserwartungen aus Anwendersicht	15
2.1	Kenngößen für die Beschreibung der Leistung eines IT-Systems	15
2.1.1	Online-Anwendung	16
2.1.2	Batch-Verarbeitung	19
2.2	Formulieren einer Leistungserwartung	20
3	Leistungsverhalten der Server	25
3.1	CPU	25
3.1.1	Relativer Performance Faktor (RPF)	25
3.1.2	Multiprozessor-Systeme	27
3.1.3	CPU-Leistung von SQ-Servern	29
3.2	CPU-naher Speicher	31
3.2.1	Hauptspeicher	31
3.2.2	Hauptspeicher auf SQ-Servern	32
3.3	Ein-/Ausgabe	33
3.3.1	Ein-/Ausgabe auf S-Servern	33
3.3.1.1	Kanal Typ FC	33
3.3.1.2	Kanal Typ S	35

3.3.2	Ein-/Ausgabe auf SQ-Servern	36
3.3.2.1	I/O-Prozessoren (X2000)	36
3.3.2.2	SAS-Anschluss	38
3.3.2.3	FC-Anschluss	38
3.4	Migration	39
3.4.1	Migration von S-Servern auf SQ-Server	41
3.4.2	Migration von SX-Servern auf SQ-Server	43
4	Leistungsverhalten der Peripherie	45
4.1	Eigenschaften externer Plattenspeichersysteme	45
4.1.1	Komponenten von Plattenspeichersystemen	45
4.1.2	Konfigurierung von Plattenspeichersystemen	48
4.1.2.1	RAID-Gruppen und LUNs	48
4.1.2.2	RAID-Level	48
4.1.2.3	Replikationsfunktionen externer Plattenspeichersysteme	50
4.1.2.4	Thin Provisioning	51
4.1.3	Konfigurierung in BS2000/OSD	52
4.1.4	Performer Betrieb	55
4.1.5	Lasttypen	57
4.2	Plattenspeichersysteme an S-Servern	59
4.2.1	ETERNUS DX	59
4.2.1.1	Durchsatz mit einer Task	60
4.2.1.2	Durchsatz mit bis zu vier Kanälen Typ FC	61
4.2.1.3	Hardware-Bedienzeiten	62
4.2.1.4	PAV und RAID-Level	63
4.2.1.5	Grenzen von PAV	66
4.2.2	Symmetrix DMX und VMAX	70
4.2.2.1	Durchsatz mit einer Task	71
4.2.2.2	Durchsatz mit bis zu vier Kanälen Typ FC	72
4.2.2.3	Hardware-Bedienzeiten	73
4.2.2.4	PAV und RAID-Level	75
4.3	Plattenspeichersysteme an SQ-Servern	79
4.3.1	ETERNUS DX	79
4.3.1.1	Durchsatz mit einer Task	80
4.3.1.2	Durchsatz mit 32 Tasks	81
4.3.1.3	Hardware-Bedienzeiten	82
4.3.2	Symmetrix DMX, VMAX	83
4.3.2.1	Durchsatz mit einer Task	83
4.3.2.2	Hardware-Bedienzeiten	84

4.4	Net-Storage	86
4.5	LAN-Anschluss	87
4.5.1	LAN-Anschluss für S-Server	88
4.5.2	LAN-Anschluss für SQ-Server	92
4.5.3	Hilfsmittel zur Leistungskontrolle	94
5	Datensicherung	97
5.1	Logische Datensicherung mit HSMS/ARCHIVE	99
5.2	Performance-Maßnahmen im Sicherungsbetrieb (HSMS)	101
5.3	Physikalische Datensicherung mit FDDRL	105
5.4	Komponenten bei der Datensicherung	107
5.4.1	BS2000-CPU	108
5.4.2	Anschluss an den Server	108
5.4.3	Bandspeichersysteme	109
5.4.4	Plattenspeichersysteme	113
5.5	Empfehlungen	115
6	Performance-Aspekte bei VM2000-Betrieb	117
6.1	Verwaltung der CPU-Leistung mit VM2000	118
6.1.1	VM2000-Scheduling (S-Server)	118
6.1.2	VM-Gruppen (S-Server)	121
6.1.3	CPU-Pools	122
6.2	Hauptspeicher	124
6.3	Peripherie	125
6.4	Gastsystem-Planung	129
6.4.1	Anzahl und Multiprozessorgrad der VMs	129
6.4.2	Einstellung der CPU-Quoten	130
6.4.3	Optimale Konfiguration von VM2000	132
6.5	Systembetreuung im Gastsystem	135
6.5.1	Einstellung der Prioritäten	135
6.5.2	PCS unter VM2000	137
6.5.3	Gastsystem-Bedienung	138
6.6	Messwerkzeuge für VM2000	139

7	Leistungsverhalten der Software	141
7.1	Größe des Benutzeradressraums	143
7.2	Performance-Kriterien beim Einrichten von Dateien	145
7.2.1	Logische Betriebsarten von Platten	145
7.2.2	Beschleunigung der Katalogzugriffe mit SCA	150
7.2.3	Einsatz von SF-Pubsets	152
7.2.4	Einsatz von SM-Pubsets	153
7.2.5	Einrichten von Systemdateien	156
7.2.5.1	Dateikatalog (SF-Pubset)	157
7.2.5.2	Dateikatalog (SM-Pubset)	161
7.2.5.3	Seitenwechselbereiche	162
7.2.5.4	SYSEAM-Datei	167
7.2.5.5	Meldungsdateien	169
7.2.6	Einrichten von Anwenderdateien	173
7.3	Arbeiten mit HIPERFILEs	176
7.3.1	Caching-Modi	177
7.3.2	ADM-PFA-Caching	180
7.3.3	User-PFA-Caching	184
7.4	Transaktionsbetrieb und BCAM	188
7.4.1	Phasen einer OLTP-Transaktion	188
7.4.2	Optimierung der einzelnen Phasen	190
8	Systemsteuerung	197
8.1	Job-Management	197
8.1.1	Job-Klassen	197
8.1.2	Job-Streams	201
8.2	Task-Management	203
8.2.1	Einführung in das PRIOR-Konzept	203
8.2.1.1	Task-Kategorien	204
8.2.1.2	Task-Prioritäten	205
8.2.1.3	Lastanalyse	208
8.2.1.4	Einstellen der Kategorieparameter	211
8.2.1.5	Prioritätenvergabe	219
8.2.1.6	I/O-Prioritäten-Steuerung für Tasks mit IORM	221
8.2.2	Einführung in das PCS-Konzept	222
8.2.3	Einführung in das TANGRAM-Konzept	233
8.3	Data-Management	236

9	Systeminterne Abläufe	239
9.1	Unterbrechungsanalyse (Funktionszustände)	239
9.2	Erzeugen und Beenden von Tasks	243
9.3	Verwalten der Betriebsmittel Hauptspeicher und CPU	244
9.3.1	Hauptspeicher-Verwaltung	246
9.3.2	CPU-Verwaltung	258
9.4	Steuern der Tasks über Warteschlangen	261
9.5	Ablauf einer Ein-/Ausgabe	265
9.5.1	Ein-/Ausgaben auf Platten	265
9.5.2	Hardware-Bedienzeit	268
9.5.3	Software-Bedienzeit	268
10	Einsatz des Mess-Systems openSM2	271
10.1	Messmonitor SM2	272
10.2	Routinemäßige Systemüberwachung	274
10.2.1	Parameter für die Messung und Auswertung	274
10.2.2	Bewertung charakteristischer Reports	274
10.2.3	Anmerkung zu verschiedenen Reports	276
10.3	Durchführung einer Engpass-Analyse	282
10.3.1	Prinzipielle Vorgehensweise	282
10.3.1.1	Parameter für die Messung und Auswertung	284
10.3.1.2	Messwerte für die Untersuchung von systemorientierten Leistungsproblemen	285
10.3.1.3	Messwerte für die Untersuchung von anwenderorientierten Leistungsproblemen	286
10.3.1.4	Auswertung einer SM2-Messung	290
10.3.2	Untersuchen eines systemorientierten Leistungsproblems	291
10.3.2.1	Auslastung der Ein-/Ausgabe-Peripherie	292
10.3.2.2	Paging-Aktivitäten	299
10.3.2.3	CPU-Auslastung	300
10.3.3	Untersuchen eines anwenderorientierten Leistungsproblems	306
10.3.3.1	Task-Belegungszeit	307
10.3.3.2	Behinderungen durch andere Tasks	308
10.3.3.3	Hoher Betriebsmittelbedarf	310
10.3.4	Einfluss des Netzes	311
10.4	Leistungsbedarf von SM2	312
10.5	Performance-Analyse mit COSMOS	313

11	Optimierungshinweise zur Erstellung von Anwender-Software	315
11.1	Wahl der Zugriffsmethode	315
11.2	Wahl des Bearbeitungsmodus	321
11.2.1	OPEN	321
11.2.2	Zugriffe	322
11.2.3	CLOSE	322
11.3	Performantes Laden von Programmen/Produkten	323
11.3.1	Prinzipieller Ablauf beim Binden/Laden eines Programms/Produkts	324
11.3.2	Allgemeine Hinweise zur Beschleunigung des Ladevorganges	325
11.3.3	Strukturelle Maßnahmen zur Reduzierung des Betriebsmittelbedarfs	327
11.3.4	Beschleunigung des Ladevorganges von C-Programmen	330
11.3.5	Nutzung von DAB	330
12	Anhang	331
12.1	Richtwerte für die BS2000/OSD-Server	331
12.2	Performance-relevante Systemparameter	336
12.3	Puffergröße der CISC-Firmware	348
	Literatur	351
	Stichwörter	355

1 Einleitung

Das Ziel jeder IT-Abteilung ist die Erfüllung der Anforderungen der Systemanwender bei gleichzeitiger Minimierung der Kosten.

Folgende Gesichtspunkte machen die Anpassung der Systemumgebung (Hard- und Software) zu einem dynamischen Prozess:

- Die stetig steigende Zunahme der Realzeitverarbeitung mit allen Möglichkeiten der direkten Abwicklung von Kundenservice und Sachbearbeitertätigkeit.
- Die stetig steigende Zunahme der Anwendungen und Funktionen (Datennetze, Verteilte Verarbeitung).
- Die Wechselbeziehungen zwischen diesen Funktionen (als Folge funktioneller Abhängigkeiten).
- Das Wachstum bestehender Anwendungen durch Ausweitung des Geschäftsvolumens.
- Die Lastschwankungen des täglichen Betriebes.

Zur Sicherstellung eines wirtschaftlichen Systemeinsatzes sollte folgende Vorgehensweise angestrebt werden:

1. Festlegung der Leistungserwartungen.
2. Auslegung der Konfiguration.
3. Nach Aufnahme des Produktionsbetriebes **Durchführen von Messungen** zur Feststellung, ob die Leistungserwartungen erfüllt werden.
4. Bei Nichterfüllung der Leistungserwartungen Konzentration auf jene Engpässe, deren Beseitigung die größtmögliche Leistungsverbesserung verspricht.
5. Nach Beseitigung der festgestellten Engpässe **Wiederholung der Messungen**, denn öfter werden nach Eingriffen andere, bislang versteckte Engpässe sichtbar.
6. Auch nach Erfüllung der Leistungserwartungen ist die **periodische Durchführung von Messungen** erforderlich, denn nur so können sich abzeichnende Sättigungserscheinungen in den Hauptbetriebsmitteln durch wachsende Belastung erkannt und kritische Systemzustände vermieden werden.

1.1 Zielsetzung und Zielgruppen des Handbuchs

Das Performance-Handbuch soll dem Systemanwender helfen, die Leistung seines IT-Systems zu beurteilen bzw. zu verbessern. Es wendet sich im besonderen an die Mitarbeiter im Data Center und in der Systembetreuung.

Hinweise für das Tuning von Konfiguration und Software sollen den wirtschaftlichen Einsatz von BS2000/OSD erleichtern.

Das vorliegende Performance-Handbuch ist Bestandteil der Basisliteratur zu BS2000/OSD-BC und damit auch von OSD/XC. In ihm werden die Performance-Aspekte des IT-Betriebs für die BS2000/OSD-Server übersichtlich zusammengefasst.

1.2 Konzept des Handbuchs

Das Handbuch beschreibt die Möglichkeiten für das Tuning von Konfiguration und Software zur Erfüllung der Leistungserwartungen und zur Optimierung des Betriebsmitteleinsatzes. Wenn erforderlich wird zwischen den einzelnen BS2000/OSD-Servern unterschieden.

Das Handbuch gibt **keine** Einführung in die Anwendung von Business Servern.

Diese Themen sind Gegenstand der Basisliteratur der Hardware, des Betriebssystems und der systemnahen Softwareprodukte in der aktuellen Version.

Das [Kapitel „Leistungserwartungen aus Anwendersicht“](#) beschreibt die für die Leistungsbeurteilung eines IT-Systems maßgebenden Größen und ihre Bedeutung.

Die Kapitel [„Leistungsverhalten der Server“](#) und [„Leistungsverhalten der Peripherie“](#) beschreiben die Bedeutung der Leistungskennzahl RPF (Relativer Performance Faktor) für die CPU-Leistung und das Leistungsverhalten der Ein-/Ausgabe-Peripherie. Anhand von Richtwerten geben die Kapitel Hinweise zur Auslegung der Ein-/Ausgabe-Konfiguration.

Die Kapitel [„Datensicherung“](#), [„Performance-Aspekte bei VM2000-Betrieb“](#) und [„Leistungsverhalten der Software“](#) beschreiben performance-relevante Besonderheiten, die beim Einsatz der BS2000-Softwareprodukte zu beachten sind.

Die Kapitel [„Systemsteuerung“](#) und [„Systeminterne Abläufe“](#) erläutern den Einfluss von Systemparametern auf den Systemdurchsatz und beschreiben innere Vorgänge in BS2000/OSD, soweit sie zum Verständnis und zur Erleichterung der Interpretation von Messwerten beitragen.

Das [Kapitel „Einsatz des Mess-Systems openSM2“](#) gibt Empfehlungen für die Parameter-einstellung bei der Durchführung von Routine-Überwachungen und Engpass-Analysen, erläutert charakteristische Reports und beschreibt die prinzipielle Vorgehensweise bei der Untersuchung eines system- bzw. anwenderorientierten Leistungsproblems.

Das [Kapitel „Optimierungshinweise zur Erstellung von Anwender-Software“](#) gibt Empfehlungen zur Wahl der Dateizugriffsmethode und -struktur und liefert Hinweise für das performante Laden von Programmen/Produkten.

Am Ende des Handbuchs finden Sie einen Anhang sowie verschiedene Verzeichnisse, die Ihnen das Arbeiten mit diesem Handbuch erleichtern.

Readme-Datei

Funktionelle Änderungen der aktuellen Produktversion und Nachträge zu diesem Handbuch entnehmen Sie bitte ggf. der produktspezifischen Readme-Datei.

Readme-Dateien stehen Ihnen online bei dem jeweiligen Produkt zusätzlich zu den Produkthandbüchern unter <http://manuals.ts.fujitsu.com> zur Verfügung. Alternativ finden Sie Readme-Dateien auch auf der Softbook-DVD.

Informationen unter BS2000/OSD

Wenn für eine Produktversion eine Readme-Datei existiert, finden Sie im BS2000-System die folgende Datei:

```
SYSRME.<product>.<version>.<lang>
```

Diese Datei enthält eine kurze Information zur Readme-Datei in deutscher oder englischer Sprache (<lang>=D/E). Die Information können Sie am Bildschirm mit dem Kommando `/SHOW-FILE` oder mit einem Editor ansehen.

Das Kommando `/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>` zeigt, unter welcher Benutzerkennung die Dateien des Produkts abgelegt sind.

Ergänzende Produkt-Informationen

Aktuelle Informationen, Versions-, Hardware-Abhängigkeiten und Hinweise für Installation und Einsatz einer Produktversion enthält die zugehörige Freigabemitteilung. Solche Freigabemitteilungen finden Sie online unter <http://manuals.ts.fujitsu.com>.

1.3 Änderungen gegenüber dem Vorgänger-Handbuch

Das vorliegende Handbuch enthält gegenüber dem Vorgängerhandbuch die nachfolgenden wesentlichen Änderungen:

- Das Handbuch wurde an BS2000/OSD V9.0 angepasst
- Neue Business Server: S175, S210, SQ200 und SQ210
- Neue Plattenspeichersysteme: ETERNUS DX
- Neues Bandlaufwerk: LTO-5
- SX-Server werden von BS2000/OSD ab V9.0 nicht mehr bedient. Ihre Beschreibung in diesem Handbuch entfällt damit. Ausnahme: Hinweise zur Migration von SX- zu SQ-Servern.
- Das Plattenspeichersystem Symmetrix 8000 ist obsolet und wird in diesem Handbuch nicht mehr beschrieben.
- Der Anschluss über Kanal Typ S wird von den neuen Band- und Plattenspeichersystemen nicht mehr unterstützt. Kanal Typ S ist daher nicht mehr Gegenstand von Performance-Messungen. Er wird in diesem Handbuch in Vergleichen noch erwähnt.

1.4 Darstellungsmittel

Wegen der häufigen Nennung der Bezeichnungen, werden der Einfachheit und Übersichtlichkeit halber folgende Abkürzungen gebraucht:

- **S-Server** für die Business Server der S-Serie
- **SQ-Server** für die Business Server der SQ-Serie
- **BS2000** für das Betriebssystem BS2000/OSD
- **SM2** für das Mess-System openSM2, sofern **keine** Unterscheidung zwischen dem Mess-System openSM2 und dem Messmonitor SM2 gemacht werden muss

In diesem Handbuch werden folgende Darstellungsmittel verwendet:



für Hinweise auf besonders wichtige Informationen

- [] Literaturhinweise werden im Text in Kurztiteln angegeben. Der vollständige Titel jeder Druckschrift, auf die durch eine Nummer verwiesen wird, ist im Literaturverzeichnis hinter der entsprechenden Nummer aufgeführt.

Kommandos, auf die in diesem Handbuch verwiesen wird, sind im Handbuch „Kommandos“ [15], die genannten Makros im Handbuch „DVS-Makros“ [8] beschrieben. Die Metasyntax der Kommandos bzw. Makros ist in den entsprechenden Handbüchern enthalten.

Hinweis zu den verwendeten Maßeinheiten

Im Handbuch werden die deutsche und die englische Schreibweise nebeneinander verwendet. Für Durchsatz und Übertragungsrate wird jedoch immer die Schreibweise Kbyte, Mbyte bzw. Gbyte verwendet. Es bedeuten:

1 KB	= 2 ¹⁰ Bytes	= 1.024 Bytes	1 Kbyte	= 10 ³ Bytes	= 1.000 Bytes
1 MB	= 2 ²⁰ Bytes	= 1.048.576 Bytes	1 Mbyte	= 10 ⁶ Bytes	= 1.000.000 Bytes
1 GB	= 2 ³⁰ Bytes	= 1.073.741.824 Bytes	1 Gbyte	= 10 ⁹ Bytes	= 1.000.000.000 Bytes

2 Leistungserwartungen aus Anwendersicht

2.1 Kenngrößen für die Beschreibung der Leistung eines IT-Systems

Jedes IT-System kann in folgende Ebenen hierarchisch untergliedert werden:

- Anwenderprogramme
- Systemsoftware
- Hardware

Die Anwenderanforderungen – repräsentiert durch die Anwenderprogramme – werden auf der Systemsoftware-Ebene in Tasks, auf der Hardware-Ebene in Befehle und Ein-/Ausgabe-Operationen umgewandelt.

Die Systemsoftware sieht nicht mehr die Anwenderprogramme als Belastung, sondern die Menge der simultan ablaufenden Tasks, welche unterschiedliche Anforderungen stellen.

Auf Hardware-Ebene erscheint die Belastung als eine Menge von Befehlen, Ein-/Ausgabe-Operationen und Speicherbelegungen, wobei viele verschiedene Belastungen auf der Anwender Ebene zu derselben Hardwarebelastung führen können.

Für den Anwender stehen bei der Leistungsbewertung eines IT-Systems folgende Fragen im Vordergrund:

- Innerhalb welcher Zeit bearbeitet das IT-System die Anwenderanforderungen?
- Wieviele Anforderungen pro Zeiteinheit kann das IT-System abwickeln?

Die Leistungsfähigkeit auf Hardware-Ebene (z.B. Anzahl Befehle pro Sekunde) oder auf Systemsoftware-Ebene (z.B. Effizienz des Task-Managements in Form von Warteschlangentransfers pro Sekunde) berührt den Anwender erst in zweiter Linie. Die Anwendersoftware beeinflusst diese Leistungsfähigkeit durch ihr Verhalten. Deshalb kann man die Betrachtung dieser Komponenten nicht ganz vernachlässigen.

Abhängig von der Art der Anwendung (Online-Anwendung, Batch-Verarbeitung) werden für die **Leistungsbeschreibung auf Anwender Ebene** die nachfolgend beschriebenen Kenngrößen verwendet.

2.1.1 Online-Anwendung

Sowohl beim Transaktions-, als auch beim Dialogbetrieb ist die Einheit der IT-Arbeit die **Transaktion**. Die Anwenderanforderungen in Form der Eingaben werden im System durch Transaktionen repräsentiert.

Beim **TP-Betrieb** (Transaktionsbetrieb) kann der Terminalbenutzer nur mit Programmen kommunizieren, die seitens der Anwendung fest vorgegeben sind. Üblicherweise arbeiten viele Terminalbenutzer mit einer relativ geringen Anzahl von **Anwenderprogrammen** zusammen.

Beim **Dialogbetrieb** formuliert jeder Terminalbenutzer seine eigene Anwendung, mit der er im Dialog die gewünschte Aufgabe abarbeitet. Die Programme, welche die jeweilige Dialoganwendung steuern, sind i.d.R. **Systemprogramme** zum Erstellen, Testen und Ändern von Dateien bzw. Programmen.

Die Zeit zwischen der Eingabe des Benutzers und der Fertigmeldung des Systems wird als **Transaktionszeit** bezeichnet. Sie kann sich aus mehreren Antworten mit unterschiedlicher Antwortzeit zusammensetzen.

Kenngößen bei Online-Anwendung

- Transaktionsrate:
Gesamtheit der erfolgreich beendeten Transaktionen pro Zeiteinheit
- Antwortzeit:
Zeit **einer** Bearbeitung durch den Server
- Anzahl der Terminals
(gleichbedeutend mit der Anzahl aktiver Terminalbenutzer)
- Wirkungsgrad pro Terminal

Darstellung der Zeitdefinitionen

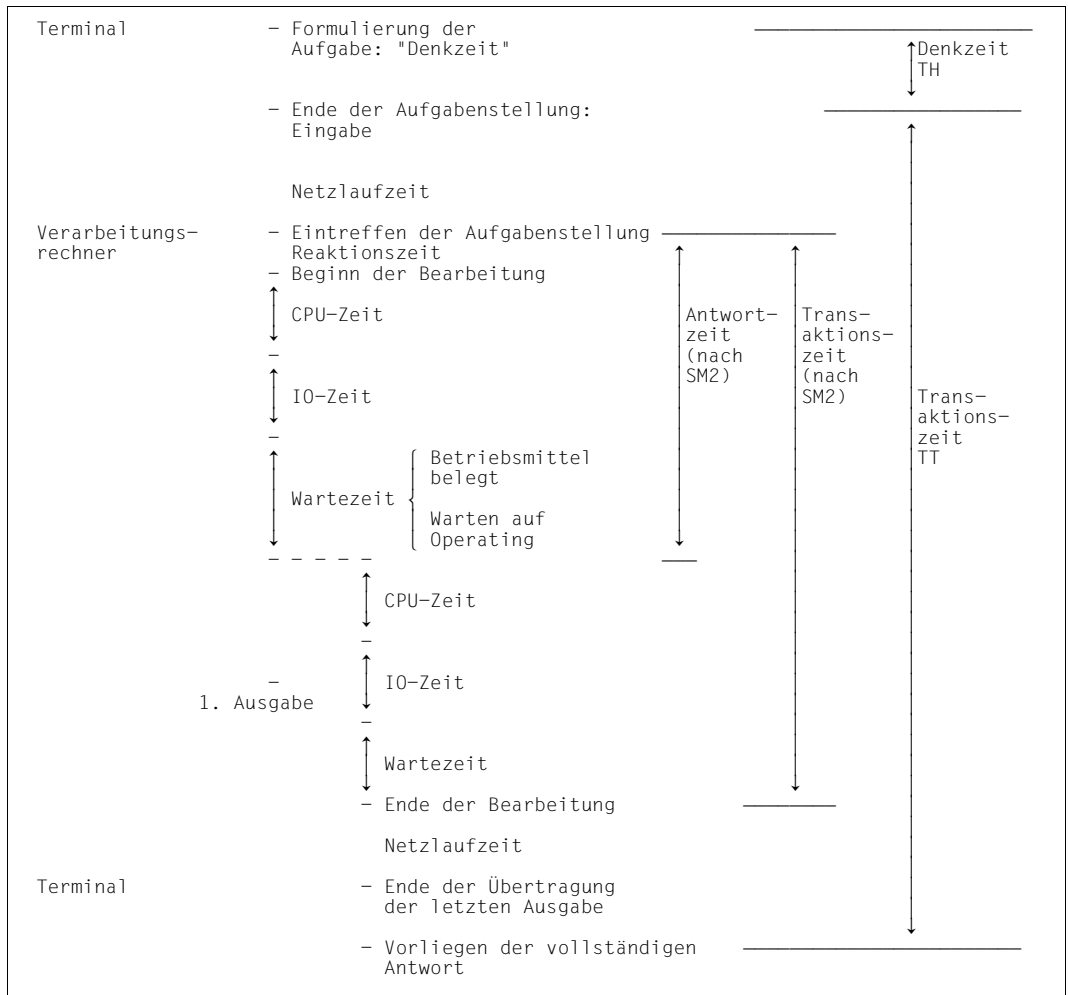


Bild 1: Zeitdefinitionen bei Online-Anwendung



Gibt es zu einer Transaktion nur **eine** Antwort (bei TP-Betrieb ist dies der Normalfall), so ist die Antwortzeit gleich der Transaktionszeit.

Die dargestellten Zeitdefinitionen (Antwortzeit, Transaktionszeit, Denkzeit) und die Transaktionsrate können durch das SM2-Messprogramm BCAM-CONNECTION erfasst werden, jedoch nur für Anwendungen, die mit dem Transportsystem BCAM über die Programmschnittstelle IDCAM oder ITIAM zusammenarbeiten.

Bei Anwendungen, die mit dem Transportsystem BCAM über die Programmschnittstelle SOCKETS zusammenarbeiten, liefert das SM2-Messprogramm BCAM-CONNECTION verbindungs-spezifische Werte über PORT-Nummern:

- **INPROC:**
Zeit vom Eintreffen einer Nachricht bei BCAM bis zum Abholen der Nachricht durch die Anwendung. Die INPROC-Zeit beinhaltet die INWAIT-Zeit, also die Zeit zwischen dem Anzeigen der Nachricht durch BCAM bei der Anwendung und dem Abholen der Nachricht durch die Anwendung.
- **REACT:**
Zeit zwischen dem Sendeaufruf und dem unmittelbar vorangegangenen Empfangsauf-ruf einer Anwendung. Bei TP-Betrieb kann diese Zeit als Antwortzeit angesehen wer-den.
- **OUTPROC:**
Zeit vom Sendeaufruf bis zur Übergabe des letzten Bytes der Nachricht an das Netz.

Der TCP-Transportservice arbeitet nicht nachrichten-orientiert, sondern stream-orientiert. Daher ist bei Transaktionen mit mehreren Antworten pro Eingabe die Ermittlung der Trans-aktionszeit durch BCAM nicht möglich. Ebenso wird die Denkzeit nicht erfasst.

Sofern die Eingabe-Nachrichtenlängen kurz sind (< 500 Bytes), kann die Anzahl empfan-gener TSDUs (Transport Service Data Units; Aufträge an BCAM) pro Sekunde als Maß für die Transaktionsrate interpretiert werden.

Die Messwerte INPROC, INWAIT, REACT und OUTPROC werden auch für Anwendungen erfasst, die mit dem Transportsystem BCAM über die Programmschnittstelle ICMX, IDCAM oder ITIAM zusammenarbeiten.

2.1.2 Batch-Verarbeitung

Die Einheit der IT-Arbeit bei Batch-Verarbeitung ist der **Auftrag** (Job).

Darstellung der Zeitdefinitionen



Bild 2: Zeitdefinitionen bei Batch-Verarbeitung

Kenngrößen bei Batch-Verarbeitung

- **Durchsatzrate:** Gesamtheit der erfolgreich ausgeführten Aufträge, bezogen auf die Zeiteinheit
- **Verweilzeit:** Zeit, die zur Verarbeitung eines Auftrages beansprucht wird

2.2 Formulieren einer Leistungserwartung

Zwischen der Transaktionszeit, der Transaktionsrate und der Anzahl Terminals besteht folgender grundlegender Zusammenhang:

$$(\text{Denkzeit} + \text{Transaktionszeit}) * \text{Transaktionsrate} = \text{Anzahl der Terminals}$$

Unter dem Begriff Transaktionszeit fasst man die Reaktion des Systems auf sehr unterschiedliche Anforderungen zusammen. Es ist deshalb schwer zu entscheiden, ob eine Transaktionszeit angemessen ist. Wegen der Kosten für den Betrieb der Terminals und die Arbeitszeit des Bedieners empfiehlt sich folgende Sicht:

Während der Transaktionszeit kann das Terminal nicht genutzt werden, da es auf den Server wartet. Während der Denkzeit arbeiten Terminal und Anwender durch Lesen der Ausgabe bzw. Eintippen einer neuen Eingabe. Der Anteil der Denkzeiten an der gesamten Belegungszeit des Terminals ist der Wirkungsgrad für Terminals E (Efficiency Value). Sinkt dieser im Mittel aller Terminals unter 75%, so ist das ein Anzeichen für Leistungsprobleme. Natürlich dürfen bei der Ermittlung der Denkzeiten „Verschnaufpausen“ nicht mitgemessen werden.

Es seien:

TH	(Think Time)	die Denkzeit
TT	(Transaction Time)	die Transaktionszeit
TR	(Transaction Rate)	die Transaktionsrate

Für den **Wirkungsgrad für Terminals (Efficiency Value, E)** gilt laut Definition:

$$E = TH / (TH + TT)$$

Die Anzahl der je Sekunde für eine Anwendung zu bewältigenden Transaktionen sei TR. Dann errechnet sich die für diese Anwendung benötigte Anzahl K von Terminals zu

$$K = (TH + TT) * TR$$

$$K = TH / E * TR$$

Nimmt man einen Wirkungsgrad für Terminals von 0,75 an, so lautet die Formel:

$$K = 1,33 * TH * TR$$

Für verschiedene Wirkungsgrade gibt die folgende Tabelle die Werte für $1/E$ an.

E [%]	70	75	80	85	90	95
$1/E$	1,43	1,33	1,25	1,18	1,11	1,05

Die Formel $E = TH / (TH + TT)$ lässt erkennen, dass für lange Denkzeiten auch eine entsprechend lange Transaktionszeit (bzw. Antwortzeit bei nur **einer** Antwort pro Transaktion) akzeptiert werden kann, während sich die Transaktionszeiten bei kurzen Denkzeiten nachhaltig auf den Wirkungsgrad und damit auf die Anzahl der benötigten Terminals auswirken. Da in der Transaktionszeit die Netzlaufzeit enthalten ist, sollten die Terminals mit kurzen Denkzeiten und langen Ausgaben über schnelle Leitungen angeschlossen werden.

Neben dem Wirkungsgrad bestimmt die Auslastung U (Utilization) des Terminals die Anzahl der benötigten Terminals. Die Auslastung ist der Anteil an der Gesamtzeit, in der das Gerät entweder auf Antworten wartet (Transaktionszeit) oder mit der Vorbereitung einer neuen Eingabe tätig ist (Denkzeit). Verschnauaufpausen z.B. senken die Auslastung, ohne dass das Gerät wirklich frei ist. Bei einer Datenerfassungsanwendung werden i.d.R. Belege von einem Stapel abgearbeitet. Man kann die Arbeitsplätze daher sehr hoch auslasten. Eine Auslastung der Terminals von 90% ist möglich. Ein kleiner Verlust von 10% wird für Verschnauaufpausen eingeplant. Für andere Anwendungen muss die Auslastung geringer angesetzt werden. Bei TP-Betrieb mit Publikumsverkehr möchte man dem Kunden lästige Wartezeiten ersparen. Diese entstehen, wenn gerade kein Terminal frei ist und der Kunde deshalb nicht sofort bedient werden kann. Ähnliche Verhältnisse findet man in der Programmentwicklung. In diesen Fällen sollte die Auslastung U der Terminals 60% nicht überschreiten.

Dann errechnet sich die **Anzahl der benötigten Terminals (K)** zu:

$$K = (TH * TR) / (E * U)$$

Beispiel

Es existieren 2 Anwendungen:

Die erste Anwendung ist eine Datenerfassung mit einer Denkzeit von 20 Sekunden und einer Transaktionszeit von 1 Sekunde. Im Mittel sind immer 90% aller Erfassungsplätze tätig. Pro Stunde müssen 7.200 Belege erfasst werden.

Die zweite Anwendung ist eine Platz-Buchungsanwendung mit Publikumsverkehr. Die Denkzeit beträgt 120 Sekunden, die Transaktionszeit 3 Sekunden. Damit keine spürbaren Wartezeiten für die buchungswilligen Kunden entstehen, werden die Terminals nur zu 60% genutzt. 360 Buchungen pro Stunde sind abzufertigen.

$$E_1 = 95\%; U_1 = 90\%; TR_1 = 2/s$$

$$(E_1 = TH / (TH + TT) = 20s / (20s + 1s) = 0.95)$$

$$E_2 = 97\%; U_2 = 60\%; TR_2 = 0.1/s$$

$$K_1 = (20 * 2) / (0.95 * 0.90) = 47 \text{ Terminals}$$

$$K_2 = (120 * 0.1) / (0.97 * 0.60) = 21 \text{ Terminals}$$

Für die beiden Anwendungen werden zusammen 68 Terminals benötigt. Es ist sicherzustellen, dass die im Wirkungsgrad für Terminals enthaltenen Vorgaben für die Transaktionszeit vom Server eingehalten werden können.

Dies ist die zu formulierende Leistungserwartung.

Sie kann nur eingehalten werden, wenn der Server durch den Betriebsmittelbedarf der Anwendung nicht überbeansprucht wird und noch gewisse Leistungsreserven hat. Dabei spielt die Belastung derjenigen Betriebsmittel eine besonders große Rolle, die von der betrachteten Anwendung stark genutzt werden. Ist die Anwendung mit vielen Ein-/Ausgaben verbunden, dann ist für eine geringe Auslastung der Plattenlaufwerke zu sorgen. Zusätzliche freie Kapazitäten im Server bringen hier keine Verbesserung. Umgekehrt ist es bei rechenintensiven Anwendungen.

Paging wirkt sich in beiden Fällen negativ aus.

Jede Transaktionszeit hat eine natürliche untere Grenze. Diese wird erreicht, wenn das Terminal seine Transaktion auf einem sonst völlig lastfreien Server mit ebenso lastfreien Datenübertragungswegen abwickelt. Eine solche Nutzung eines IT-Systems ist natürlich unwirtschaftlich. Kommen andere Terminals und ggf. andere Anwendungen hinzu, dann wird der Ablauf durch gegenseitige Behinderung der Transaktionen untereinander gedehnt.

Der **Dehnfaktor (D)** wird wie folgt definiert:

$$D = TT(\text{aktuelle Belastung}) / TT(\text{leerer Server})$$

Die Dehnung ist der Preis für die wirtschaftliche Nutzung der Installation. Welche Dehnungen akzeptabel sind, hängt wieder von der Denkzeit und von der optimalen Transaktionszeit TT (leerer Server) ab. Bei kurzen Transaktionszeiten und langen Denkzeiten sind Dehnungen von 10 noch akzeptabel. Beträgt die optimale Transaktionszeit schon 3 Sekunden bei einer Denkzeit von nur 10 Sekunden, dann ist bereits die Dehnung 3 zu groß. Die **optimale Transaktionszeit** kann man entweder auf einem sonst betriebslosen Server messen oder nach folgender Formel abschätzen:

$$TT(\text{leerer Server}) = CPU + n * IO * NL$$

mit:

CPU	für die Transaktion benötigte CPU-Zeit
IO	mittlere Zeit für einen Ein-/Ausgabe-Vorgang
n	Anzahl der für die Transaktion benötigten Ein-/Ausgaben (inkl. Paging)
NL	Netzlaufzeit für die Eingabe- und die Ausgabenachricht

Alle diese Werte sind durch Modifikationen an Hardware oder Software änderbar. Bei den CPU- und IO-Zeiten sind meist mehrere Tasks zu berücksichtigen, beim Transaktionsbetrieb etwa UTM- und DBH-Tasks.



Die von SM2 gemessene Dehnung berücksichtigt nicht die Netzlaufzeit. Es werden nur die Server-internen Daten verarbeitet.

Beispiel 1

Datenbankabfrage mit folgenden Werten:

CPU	0,1s
IO	0,004s
NL	0,1s
n	60

$$TT(\text{leerer Server}) = 0,1s + 60 * 0,004s + 0,1s = 0,44s$$

Bei einem Dehnfaktor von 3 für Server und Netz ergibt sich ein TT_{aktuell} von $3 * 0,44 = 1,32 \approx 1,3$. Mit einer Denkzeit von 40 Sekunden berechnet sich ein Wirkungsgrad für Terminals von 97%:

$$E = TH / (TH + TT_{\text{aktuell}}) = 40s / (40s + 1,3s) = 0,97$$

Beträgt die Denkzeit nur 20 s, sinkt der Wirkungsgrad für Terminals auf 94%. In diesem Fall wäre zu prüfen, ob sich die IO-Zeiten verringern ließen, etwa durch weniger Ein-/Ausgaben oder durch eine schnellere Peripherie.

Beispiel 2

Blättern im Editor mit folgenden Werten:

CPU	0,01s
IO	0,004s
NL	1s
n	2

$$TT \text{ (leerer Server)} = 0.01s + 2 * 0.004s + 1s = 1.02s$$

Hier dominiert die Übertragungszeit des Ausgabebildschirms. Selbst bei einer Überlastung der Platten im Server (z.B. 0,025 s/IO) steigt die Transaktionszeit nur unmerklich auf 1,06 s. Bei einer Denkzeit von 3 Sekunden ergibt sich ein Wirkungsgrad für Terminals von 75%. Das ist unbefriedigend.

$$E = TH / (TH + TT) = 3s / (3s + 1.02s) = 0.746$$

Eine Verdopplung der Leitungsgeschwindigkeit führt hier zu einer optimalen Transaktionszeit von 0,53 s. Nimmt man an, dass während der Übertragung der Nachrichten keine Dehnungen auftreten, dann bleibt diese Transaktionszeit auch bei realistischer Last erhalten. Der Wirkungsgrad für Terminals steigt auf 85%.

Die beiden Beispiele zeigen, dass die Ursachen für eine unzureichende Leistung vielfältig sein können. Überlastungen in der CPU, in der Peripherie und Kombinationen von Engpässen können auftreten. Der [Abschnitt „Durchführung einer Engpass-Analyse“ auf Seite 282](#) und das [Kapitel „Optimierungshinweise zur Erstellung von Anwender-Software“ auf Seite 315](#) geben Hilfen zum Aufdecken und Beseitigen solcher Probleme.

Die angeführten Formeln und Beispiele verdeutlichen, wie der Anwender seine Leistungserwartungen für den Online-Betrieb formulieren sollte. Das Ziel seines Handelns ist immer eine Kostenminimierung. Rechnet man die Einsparungen auf der einen Seite (z.B. für Personal und Terminals) gegen die Mehrkosten (z.B. für schnellere Leitungen, schnellere Rechner, schnellere Platten oder Programmänderungen) auf der anderen Seite auf, dann kommt man immer zu einer fundierten Leistungserwartung. Unter Berücksichtigung der Preisbewegungen für die Hardware sollten die gefundenen Entscheidungen in gewissen Zeitabständen überprüft werden.

Natürlich dürfen subjektive Gesichtspunkte nicht ganz vernachlässigt werden, etwa die psychologisch schädliche Wirkung gelegentlicher langer Transaktionszeiten. Die darauf aufgebauten Leistungserwartungen führen jedoch auf Irrwege. Die Beseitigung einzelner langer Transaktionszeiten ist meist sehr aufwändig und führt i.d.R. nicht zu generellen Einsparungen.

3 Leistungsverhalten der Server

Eine vergleichende Übersicht über die Leistungsdaten und Richtwerte der verschiedenen Servermodelle finden Sie im [Abschnitt „Richtwerte für die BS2000/OSD-Server“ auf Seite 331](#).

3.1 CPU

3.1.1 Relativer Performance Faktor (RPF)

In der BS2000-Welt wird die Leistung eines Servers mit der Kennzahl RPF (Relativer Performance Faktor) ausgedrückt. Der RPF-Wert gibt die Leistungsfähigkeit der CPUs unter realistischer Last an.

Zur RPF-Bestimmung werden Messungen mit zwei Benchmark-Lasten durchgeführt. Diese bestehen aus einer OLTP-Last (TP-Benchmark mit UTM- und SESAM/SQL-Applikationen) und einer Batch-Last (SKBB-Benchmark mit COBOL-Programmen).

Es wird der Leistungswert für den Batch-Betrieb (RPF_{SKBB} über das Leistungsmaß „Jobs pro CPU-Sekunde“) und den TP-Betrieb (RPF_{TP} über das Leistungsmaß „Transaktionen pro CPU-Sekunde“) ermittelt.

Das Ergebnis des TP-Betriebs geht nun mit 75%, das Ergebnis des Batch-Betriebs mit 25% in den Gesamt-RPF-Wert ein.

Beispiele für die Ermittlung des RPF-Wertes (die ermittelten Messwerte werden zu den Messwerten der Referenzmaschine C40-F (RPF=1) in Bezug gesetzt):

Modell	Anzahl CPUs	RPF _{SKBB} (RPF-Wert mit SKBB-Benchmark)	RPF _{TP} (RPF-Wert mit TP-Benchmark)	RPF _{gesamt} (gewichteter Mittelwert aus RPF _{SKBB} und RPF _{TP})
C40-F	1	1	1	1
S175-10E	1	556	325	390
S210-40	4	2.685	1.480	1.770
SQ200-10E	1	140	120	125
SQ200-80E	8	890	650	700
SQ210-10F	1	177	175	175
SQ210-160F	16	2100	1700	1750

Tabelle 1: Beispiele zum RPF/Wert



Es ist generell problematisch, das Leistungsverhalten eines Servers mit einer einzigen Zahl auszudrücken. Wie aus vorangehender Tabelle zu ersehen ist, hängt die Leistung in gewissen Grenzen vom Anwendungsprofil ab.

Auf SQ-Servern ist wegen der besonderen Architektur dieser Server und des lastabhängigen Wirkungsgrades der CISC-Firmware (siehe [Seite 29](#)) eine stärkere Abhängigkeit vom Anwendungsprofil und damit eine größere Bandbreite der Leistung zu erwarten.

3.1.2 Multiprozessor-Systeme

Multiprozessor-Systeme bestehen aus 2 bis 16 Verarbeitungsprozessoren und 1 bis 8 I/O-Prozessoren. Die Vorteile liegen in der Erhöhung der Verfügbarkeit sowie einer entsprechenden Leistungssteigerung.

Voraussetzung für eine performante Nutzung von Multiprozessor-Systemen ist die Multiprozessor-Fähigkeit der Anwendung: Dazu muss die Anwendung in der Lage sein, die Anforderungen nicht nur parallel abzuarbeiten, sondern diese auch ohne wesentliche gegenseitige Beeinträchtigung abzuwickeln.

Erhöhte Verfügbarkeit

Bei geeigneter, redundanter Auslegung der Hardware erfolgt bei Ausfall einer Hardware-Komponente die automatische Rekonfiguration durch das System ohne Unterbrechung des laufenden Betriebs.

Leistungssteigerung

Im Multiprozessor-Modus können unter der Steuerung **eines** BS2000-Betriebssystems mehrere Tasks parallel (je nach Anzahl der CPUs) bedient werden. Um dieses mehrfache Leistungsangebot zu nutzen, muss also dafür gesorgt werden, dass stets genügend Tasks zur parallelen Verarbeitung vorhanden sind.

Es ist zu beachten, dass jeder Task nur die Leistung **einer** CPU zur Verfügung steht. Daraus ergibt sich, dass das Antwortzeitverhalten nur bedingt verbessert werden kann (durch die weiteren CPUs kann lediglich die Wartezeit auf CPU-Zuteilung verkürzt werden). Analoges gilt für Batch-Anwendungen: Nur der Teil der Laufzeit einer Batch-Anwendung kann deutlich reduziert werden, der durch Parallelisierung des reinen Rechenbedarfs erreicht wird.

Zur Höhe der verbrauchten CPU-Zeit einer Task ist anzumerken, dass diese (bei gleicher Arbeit) i.d.R. etwas höher ist, als auf dem entsprechenden Monoprozessor. Je größer die Anzahl CPUs ist, desto stärker steigt der Aufwand für die Hauptspeicher-Synchronisation, verbunden mit einer Abnahme der Cache-Hit-Rate.

Die CPU-Zeit wird sich in folgendem Rahmen erhöhen:

- bei 2 CPUs um 5 bis 10%
- bei 4 CPUs um 10 bis 15%
- bei 6 CPUs um 15 bis 20%
- bei 8 CPUs um 20 bis 30%
- bei 10 CPUs um 25 bis 35%
- bei 12 CPUs um 30 bis 40%
- bei 16 CPUs um 35 bis 45%

Unter der Voraussetzung der Aufteilung der Last auf parallel laufende, möglichst voneinander unabhängige Tasks kann die Transaktionsrate wesentlich gesteigert werden. Gegenüber einem Monoprozessor können auf S-Servern folgende Durchsatzsteigerungen erreicht werden:

- bei 2 CPUs um den Faktor 1,7 bis 1,9
- bei 4 CPUs um den Faktor 3,3 bis 3,6
- bei 6 CPUs um den Faktor 4,6 bis 5,2
- bei 8 CPUs um den Faktor 5,8 bis 6,6
- bei 10 CPUs um den Faktor 6,8 bis 7,8
- bei 12 CPUs um den Faktor 7,8 bis 8,8
- bei 16 CPUs um den Faktor 8,8 bis 9,9

Um bei erhöhter Task-Anzahl die gestiegenen Betriebsmittelanforderungen zu befriedigen, ist der Ausbau des Hauptspeichers und die Erhöhung der Anzahl der Plattenlaufwerke unbedingt erforderlich.

Empfohlene maximale CPU-Auslastung

Bei antwortzeitkritischen Anwendungen sollte bei Monoprozessoren eine CPU-Auslastung von 70% für die Hauptanwendung nicht überschritten werden, weil mit steigender Auslastung die Wartezeit in der Warteschlange vor der CPU überproportional ansteigt (siehe auch [Abschnitt „Untersuchen eines systemorientierten Leistungsproblems“ auf Seite 291](#)).

Bei Multiprozessor-Systemen ist die Wahrscheinlichkeit, eine freie CPU zu finden, umso größer, je mehr CPUs zur Verfügung stehen. Deshalb können auch höhere Auslastungen toleriert werden. Bei antwortzeitkritischen Anwendungen sollten folgende CPU-Auslastungen für die Hauptanwendung nicht überschritten werden:

- 75% bei 2 CPUs
- 80% bei 4 CPUs
- 85% bei 6 CPUs
- 90% ab 8 CPUs

3.1.3 CPU-Leistung von SQ-Servern

Grundlegende Informationen über SQ-Server finden Sie im Handbuch „Bedienen und Verwalten“ [4]. Die folgenden Abschnitte beschäftigen sich mit den Komponenten der SQ-Server, die Auswirkungen auf die CPU-Leistung der SQ-Server haben.

CPUs

Die verfügbaren CPUs werden logisch aufgeteilt in CPUs für Anwendungen unter BS2000/OSD (BS2000-CPU) und CPUs, auf denen BS2000-I/Os und Administrationsaufgaben für den SQ-Server durch X2000 abgewickelt werden (I/O-Prozessoren).

Abhängig vom Modell können bis zu 16 CPUs (SQ210-160F) als BS2000-CPU genutzt werden. 25% der verfügbaren CPUs werden als I/O-Prozessoren genutzt.

Eine spezielle X2000-Firmware-Schicht unterstützt den Ablauf des Betriebssystems BS2000/OSD und den kompatiblen Ablauf von /390-Anwendungen auf den BS2000-CPU, sowie die Durchführung von Ein-/Ausgaben auf den I/O-Prozessoren.

CISC-Firmware, JIT390

Die CISC-Firmware (CISCFW) ist die Firmware-Komponente zur Abbildung von nicht privilegiertem /390-Code auf x86-64-Code. Sie ergänzt den x86-64-Modus und ermöglicht es, existierenden /390-Code objektcompatibel (im /390-Modus, synonym: Kompatibilitätsmodus) auf x86-64-Hardware ablaufen zu lassen.

Die CISC-Firmware enthält ab SQ200 die Komponente JIT390, einen Just-In-Time-/390-Code-Übersetzer, der /390-Code zur Ablaufzeit in X86-64-Code umsetzt. Ein Code-Block wird erst bei seiner Ausführung übersetzt und in einem task-lokalen JIT-Puffer vorgehalten. Kommt der Code-Block erneut zur Ausführung, so wird der bereits übersetzte und optimierte Code aus dem JIT-Puffer direkt ausgeführt. Die Wiederverwendung von vorgefertigten und optimierten Code-Teilen ist im Vergleich zur erneuten Interpretation deutlich schneller bei gleichzeitig geringerer Prozessorbelastung. Zur weiteren Optimierung wird der JIT-Puffer als residenter Speicher angelegt.

Bestehende (unprivilegierte) Kundenanwendungen laufen unverändert im /390-Code objektcompatibel mittels der CISC-Firmware ab. Dies gilt sowohl für Anwendungen, die mit ASSEMBH erzeugt wurden, als auch für Programme, die mit den BS2000-Compilern für höhere Programmiersprachen erzeugt wurden. SQ-Server unterstützen auch ESA-Datenräume in Kundenanwendungen.

Die effektive Leistung eines SQ-Servers ist davon abhängig, in welchem Ausmaß innerhalb einer /390-Anwendung Systemaufrufe vorkommen und wie stark die /390-Anwendung von der Effizienz der CISC-Firmware profitiert.

Systemaufrufe werden im Funktionszustand TPR bzw. SIH des Servers (X86-64-Modus) abgearbeitet. Die Anwendungen laufen im Funktionszustand TU (/390-Modus) ab.

Der Wirkungsgrad der CISC-Firmware hängt also von der Nutzung des JIT-Puffers durch die jeweilige Last ab: einerseits von der Wiederholungsrate bestimmter /390-Code-Teile (geringerer CPU-Bedarf) und andererseits von der Größe der in X86-64-Code übersetzten Programmteile einer /390-Anwendung (höhere Hauptspeicher- und Hauptspeicher-Cache-Belastung).

CPU-Leistung

Die nominalen RPF-Werte (siehe [„Richtwerte für SQ-Server“ auf Seite 334](#)) lassen nicht immer zuverlässige Rückschlüsse auf Laufzeiten, Transaktionsraten oder CPU-Zeitverbrauch zu. Diese Werte können anwendungsbedingt nach oben oder nach unten abweichen.

Die für einen SQ-Server gemessenen RPF-Werte gelten für Anwendungen mit einem TU-Anteil von 40 - 50%. Bei Anwendungen mit einem TU-Anteil außerhalb dieses Bereichs kann die Leistung der Anwendung von den nominalen RPF-Werten abweichen.

Siehe auch den [Abschnitt „Migration von S-Servern auf SQ-Server“ auf Seite 41](#).

3.2 CPU-naher Speicher

3.2.1 Hauptspeicher

Insbesondere für TP-Betrieb ist eine zu hohe Paging-Rate kritisch (siehe die Server-spezifische maximal empfohlene Anzahl von Paging-Ein-/Ausgaben im [Abschnitt „Richtwerte für die BS2000/OSD-Server“ auf Seite 331](#)). Eine zu hohe Paging-Rate führt immer zu unbefriedigenden Antwortzeiten. Wenn die kritische Anzahl von Paging-Ein-/Ausgaben immer wieder erreicht oder sogar überschritten wird, dann ist eine Hochrüstung des Hauptspeicherausbaues nötig.

Im VM2000-Betrieb sollte vor einer allgemeinen Hochrüstung des Hauptspeicherausbaues der Hauptspeicher optimal auf die Gastsysteme aufgeteilt werden.

Überwachen der Hauptspeichernutzung

Die Auslastung des Hauptspeichers sollte von der Systembetreuung kontinuierlich mit SM2 beobachtet werden, um einen drohenden Hauptspeicherengpass frühzeitig zu erkennen.

Für die Berechnung und Bewertung der Speicherauslastung sind folgende Kennzahlen von Bedeutung (siehe auch [„Seitenverwaltungs-Algorithmus \(System Working Set-Verfahren, SYS-WS\)“ auf Seite 256](#)):

- NPP (Number of Pageable Pages): Anzahl der seitenwechselbaren Seiten (entspricht der Größe des zur Verfügung stehenden Hauptspeichers minus der Anzahl residenter Seiten)
- SWS (System Working Set): Anzahl der global genutzten Seiten (entspricht NPP minus der Anzahl der frei nutzbaren Seiten im so genannten Freepool)



Das Ergebnis der hier empfohlenen Bewertungsmethode ist umso genauer, je mehr man sich einem Speicherengpass bzw. der kritischen Paging-Rate nähert.

Der Nutzungsgrad des Hauptspeichers (in Prozent) ergibt sich aus der Formel:

$$N = \text{SWS} * 100 / \text{NPP} \%$$

Messwerte für N bis 75% gelten als unkritisch.

Bei Werten für N größer als 90% sollte an eine Hochrüstung des Hauptspeicherausbaues gedacht werden. Dies gilt insbesondere, wenn die neue Version eines Softwareprodukts eingeführt werden soll und diese Software von mehreren Tasks verwendet wird.

3.2.2 Hauptspeicher auf SQ-Servern

Der Hauptspeicher wird zwischen BS2000/OSD und der Domäne 0 (Dom0) bzw. X2000 (I/O-Prozessoren) aufgeteilt.

Eine Empfehlung für den Hauptspeicherausbau der SQ-Server, die für die meisten Anwendungsfälle geeignet ist, finden Sie im Abschnitt „[Richtwerte für SQ-Server](#)“ auf Seite 334. Die angebotenen Standardmodelle der SQ-Server werden mit diesem empfohlenen Speicherausbau ausgeliefert.

In den meisten Fällen müssen die Standardeinstellungen für die Hauptspeicherverwaltung nicht geändert werden. Eine Vergrößerung des Hauptspeichers ist in folgenden Fällen in Betracht zu ziehen:

- Vergrößerung des gesamten Hauptspeichers

Bei Einsatz mehrerer BS2000- oder Linux-/Windows-Gastsysteme muss die Größe des gesamten Hauptspeichers so gewählt werden, dass die Summe des Hauptspeicherbedarfs aller parallel ablaufenden Gastsysteme den insgesamt zur Verfügung stehenden Hauptspeicher nicht überschreitet. Dabei ist auch der Hauptspeicheranteil für Dom0/X2000 (Standard: 30% des gesamten Hauptspeichers) zu berücksichtigen.

- Hauptspeichergroße von BS2000/OSD

Für jede VM mit BS2000-Gastsystem werden für das Gastsystem und den Mikro-Kernel zusammen mindestens 512 MB Hauptspeicher benötigt; der von den Anwendungen benötigte Hauptspeicher ist darin nicht enthalten. Der insgesamt benötigte Hauptspeicher für ein Gastsystem hängt stark von den Anwendungen und der geforderten Leistung ab. Grundsätzlich erfordert der Einsatz extrem vieler oder extrem großer Tasks bzw. Prozesse einen größeren Hauptspeicherausbau.

- JIT-Puffer

Der Bedarf an JIT-Puffer für den JIT-Compiler ist zu berücksichtigen. Dafür werden standardmäßig 40% des Hauptspeichers von BS2000/OSD reserviert. Dieser Speicher steht dem Betriebssystem und den Anwendungen nicht direkt zur Verfügung. Weitere Informationen finden Sie im [Abschnitt „Puffergröße der CISC-Firmware“](#) auf Seite 348.

3.3 Ein-/Ausgabe

3.3.1 Ein-/Ausgabe auf S-Servern

An den S-Servern können Geräte über Kanalanschlusstechnik (Kanalperipherie) betrieben werden. Fibre-Channel-Peripherie (FC-Peripherie) kann über den Kanal Typ FC angeschlossen werden.

3.3.1.1 Kanal Typ FC

Die Datenübertragung erfolgt seriell über einen Lichtwellenleiter mit einer maximalen Datenrate von 100 Mbyte/s.

Wesentliche Leistungsmerkmale des Kanals Typ FC sind:

- Vollduplex Betrieb
- Maximal **acht** Ein-/Ausgabe-Operationen (von verschiedenen Steuerungen) können zu einer Zeit durchgeführt werden
- Effektive Datenrate bei großen Ein-/Ausgabe-Blöcken: mehr als 90 Mbyte/s möglich
- Anschluss von FC-Geräten über FC-Switch möglich (HNC und SKP können auch direkt angeschlossen werden)
- Maximale Entfernung eines FC-Switches: 550 m
- Distanzen bis 100 km werden ohne große Performanceverluste überbrückt
- Es werden nur Platten mit FBA-Formatierung unterstützt (siehe [Abschnitt „Eigenschaften externer Plattenspeichersysteme“ auf Seite 45](#)).

Folgende Plattenspeichersysteme und Geräte werden über Kanal Typ FC unterstützt:

- ETERNUS DX
- Symmetrix DMX, VMAX
- ETERNUS CS HE:
VTA (Virtual Tape Appliance), SBU (Smart Backup Unit), VTC (Virtual Tape Controller)
- MBK-Geräte LTO-x in den MBK-Archivsystemen Scalar Scalar i500, i2000, i6000 und 10K
- High-Speed Net Connect HNC ab HNC-IV 91853
- Service-/Konsolprozessor SKP ab SKP 3970-51

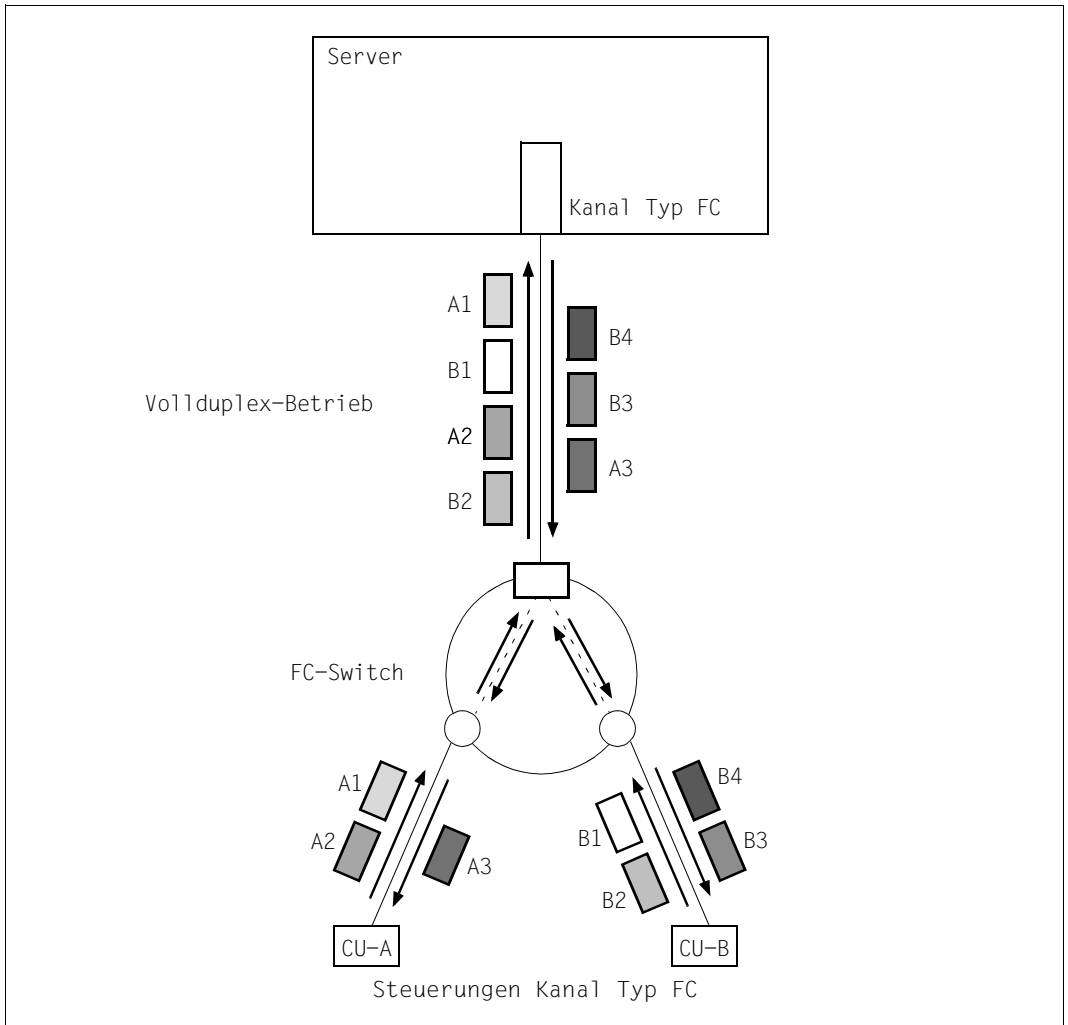


Bild 3: Kanal Typ FC

Der Durchsatz auf einem S-Server ist gekennzeichnet durch folgende Eigenschaften:

- Bei Durchsatz-orientierten Lasten (also mit großen Blöcken ab 32 KB) ist der Durchsatz begrenzt durch die Leistungsfähigkeit der FC-Verbindung zum Plattenspeichersystem (1 GBit/s).

Bei reinem Lesen oder Schreiben ist ein Durchsatz von mehr als 90 MB/s möglich.

Bei Zugriffen mit 25% Schreibanteil kann die Duplex-Eigenschaft des Kanals Typ FC ausgenutzt werden; damit kann ein Durchsatz von 110 MB/s erreicht werden.

- Im OLTP-Betrieb (also mit kleinen Blöcken von 2 - 4 kB) wird der Durchsatz entweder durch die Leistungsfähigkeit der CPUs, des Kanals oder des Plattenspeichersystems begrenzt.
- Kanäle Typ FC skalieren sehr gut. Das heißt, mit n Kanälen Typ FC kann annähernd der n-fache Durchsatz erreicht werden. Voraussetzung dafür ist eine ausreichende Leistung der BS2000-CPU's und der Peripherie.



Durch die parallele Abwicklung von bis zu acht Kanalprogrammen kann die Auslastung eines Kanals Typ FC nicht direkt mit openSM2 gemessen werden. Die Auslastung wird daher indirekt bestimmt, indem die aktuelle Anzahl Ein-/Ausgabe-Operationen mit einem Durchsatzwert verglichen wird, der bei Labormessungen maximal erreicht werden konnte.

3.3.1.2 Kanal Typ S



Dieser Kanaltyp wird in BS2000/OSD nur noch aus Kompatibilitätsgründen unterstützt. Moderne Platten- und Bandspeichersysteme können nicht mehr über Kanal Typ S angeschlossen werden. Kanäle Typ S werden durch die wesentlich performanteren Kanäle Typ FC abgelöst.

Bei Transaktions-orientierten Lastfällen können etwa 4 Kanäle Typ S durch einen Kanal Typ FC ersetzt werden. Bei Durchsatz-orientierten Lastfällen können bis zu 7 Kanäle Typ S durch einen Kanal Typ FC ersetzt werden.

3.3.2 Ein-/Ausgabe auf SQ-Servern

An SQ-Servern können Geräte unterschiedlich angeschlossen werden:

- über lokale SAS-Verbindungen (z.B. ETERNUS JX40)
- über Fibre Channel (FC-Peripherie (z.B. ETERNUS DX)
- über Ethernet (z.B. OnetStor)

Die Ein-/Ausgaben erfolgen über I/O-Prozessoren und die PCI-Controller des SQ-Servers.

3.3.2.1 I/O-Prozessoren (X2000)

Alle Ein-/Ausgaben zur Peripherie werden über die I/O-Prozessoren (X2000) geleitet. Die entsprechenden Schnittstellen werden in X2000 bedient.

Der Anschluss der Peripherie erfolgt über PCI-Controller:

- SAS:
Anschluss der internen Systemplatten (nicht zur Nutzung unter BS2000/OSD freigegeben), des internen Plattenspeichersystems ETERNUS JX40, des internen LTO-Laufwerks (ab SQ200) und des internen MBK-Systems ETERNUS LT40
- Fibre Channel:
 - ETERNUS DX
 - Symmetrix DMX, VMAX
 - ETERNUS CS HE: VTA (Virtual Tape Appliance), SBU (Smart Backup Unit), VTC (Virtual Tape Controller)
 - MBK-Geräte LTO-x in den MBK-Archivsystemen Scalar Scalar i500, i2000, i6000 und 10K
 - MBK-System ETERNUS LT40 mit LTO-x-Laufwerken.
- LAN-Controller (Ethernet, Fast Ethernet, Gigabit Ethernet)

Die Leistung der I/O-Prozessoren ist sehr gut und für alle SQ-Server ausreichend:

- Platten:

Die folgenden Daten wurden auf einer SQ210-80F gemessen, die über 4 FC-Anschlüsse mit je 8 GBit an ein Plattenspeichersystem angeschlossen war. Die Platten innerhalb des Plattenspeichersystems waren über 2 Pfade erreichbar. Mit dieser Konfiguration wäre rein rechnerisch ein maximaler Durchsatz von etwa 3.000 MB/s erreichbar.

[In eckigen Klammern werden zusätzlich die gemessenen Ergebnisse auf einer SQ200-80E ausgegeben, die über 4 FC-Anschlüsse mit je 2 GBit an ein Plattenspeichersystem angeschlossen war. Mit dieser Konfiguration wäre rein rechnerisch ein maximaler Durchsatz von etwa 800 MB/s erreichbar.]

- Bei Einsatz von nur einer Task und Verwendung kleiner Blöcke (2 KB) liegt die I/O-Rate beim Lesen bei über 2.700 [2.500] IO/s und beim Schreiben bei über 2.300 [1.700] IO/s.
- Bei paralleler Verarbeitung mit 32 Tasks und Verwendung kleiner Blöcke (2 KB und 4KB) liegt die maximale I/O-Rate bei über 48.000 [30.000] IO/s.
- Bei Einsatz von nur einer Task und Verwendung großer Blöcke (160 KB) liegt der Durchsatz beim Lesen bei 100 [85] MB/s und beim Schreiben bei 125 [80] MB/s.
- Bei paralleler Verarbeitung mit 32 Tasks liegt der maximale Durchsatz - in Abhängigkeit von den Aktionen (Lesen, Schreiben, Mischlast) - bei Blockgrößen ab 32 KB bei ca. 1.900 [740] MB/sec. Dies ist angesichts des theoretisch erreichbaren maximalen Durchsatzes ein gutes Ergebnis.

Generell gilt: mit zunehmender Blockgröße sinkt die I/O-Rate und steigt der Durchsatz.

- Bänder:

Die Ein-/Ausgabe auf Band wird im I/O-Prozessor optimiert. Der Durchsatz auf ein leistungsfähiges Band (z.B. auf ETERNUS CS HE) liegt bei 200 Mbyte/s.

- Networking:

Die folgenden Daten wurden auf einer SQ210-160F gemessen

[In eckigen Klammern werden zusätzlich die gemessenen Daten auf einer SQ200-80E ausgegeben.]

- Es wird eine maximale Transaktionsrate von annähernd 88.500 [33.000] TA/s erreicht.
- Mit parallelen Verbindungen kann das 1 GBit-LAN auf beiden Servern, sowohl beim Senden als auch beim Empfangen, ausgelastet werden (max. 120 MB/s in jeder Richtung).
- Der Durchsatz und der CPU-Bedarf beim Senden hängt von der Nachrichtenlänge ab (8, 16, 32, 64 KB). Je länger die Nachricht, desto höher der Durchsatz und geringer der CPU-Bedarf. Beim Empfangen von Nachrichten besteht diese Abhängigkeit nicht; das Netz ist immer voll ausgelastet.

Um Hochleistungsnetze ausnutzen zu können, ist eine ausreichende CPU-Kapazität erforderlich. Die Transaktionsrate ist nur durch die BS2000-CPU's begrenzt.

3.3.2.2 SAS-Anschluss

Als Direct Attached Storage (DAS) erweitert das Plattensystem ETERNUS JX40 die Speicherkapazität der SQ-Server über eine extrem schnelle SAS-Verbindung (6 Gbit/s) um max. 72 Terabyte.

Auf einer SQ200-80E mit ETERNUS JX40 wurde bei Random-Zugriffen mit 25% Schreibanteil und 100% Read-Hit-Rate bei 32 parallelen Tasks eine maximale Transaktionsrate von 30.000 TA/s (bei 2-KB-Blöcken) und ein maximaler Durchsatz von über 570 MByte/s (bei 32-KB-Blöcken) bzw. von über 900 MB/s (bei 160-KB-Blöcken) erreicht.

Die Leistung eines I/O-Prozessors ist auch für hohe Ein-/Ausgabe-Lasten selbst auf dem größten SQ-Server ausreichend.

Beim Einsatz des optionalen, über SAS-RAID angeschlossenen Plattensystems ETERNUS JX40 mit Backup Battery Unit (BBU) für den Write-Cache wird standardmäßig alle 30 Tage ein BBU Relearn Cycle durchgeführt. Der BBU Relearn Cycle dauert bis zu 15 Stunden. Er entlädt und lädt die BBU mehrfach vollständig. Während dieser Zeit ist der Write-Cache ausgeschaltet; der damit verbundene zusätzliche Beschleunigungseffekt entfällt. Das kann bei bestimmten Anwendungen zu höheren Lauf- bzw. Antwortzeiten führen. Der BBU Relearn Cycle kann durch den Service kundenspezifisch konfiguriert werden, so dass z.B. der BBU Relearn Cycle stets nur an einem Wochenende ausgeführt wird.

3.3.2.3 FC-Anschluss

Auf SQ-Servern können PCI-Controller mit einer Leistung von bis zu 8 Gbit/s eingesetzt werden.

Bei Platten-Ein-/Ausgaben über FC ist die Leistung eines I/O-Prozessors auf einem SQ-Server vergleichbar mit den Messwerten des SAS-Anschlusses.

Die Leistung eines I/O-Prozessors ist auch für hohe Ein-/Ausgabe-Lasten selbst auf dem größten SQ-Server ausreichend.

3.4 Migration

Unabhängig vom Quell- bzw. Ziel-HSI eines Servers gibt es einige allgemeine Gesichtspunkte, die bei der Migration eines Servers zu beachten sind:

Anzahl der BS2000-CPU's eines Servers

Wenn sich der Multiprozessorgrad eines Servers bei einer Migration ändert, dann kann dies auch ein geändertes Leistungsverhalten der auf dem Server ablaufenden Software zur Folge haben. Prüfen Sie in einem solchen Fall die relevanten Parameter und Einstellungen, die das Leistungsverhalten der Software steuern.

Prüfen Sie insbesondere die Einstellungen für den OLTP-Betrieb, z.B. die Anzahl der verwendeten UTM-Tasks, die Anzahl der TAC-Klassen, die Anzahl der Tasks pro TAC-Klasse, die Anzahl der DBHs, die Puffergrößen des Datenbanksystems.

Passen Sie die Parameter und Einstellungen ggf. an.

Nähere Informationen dazu finden Sie im [Kapitel „Leistungsverhalten der Software“ auf Seite 141](#) sowie in den Handbüchern zu openUTM, im Handbuch „SESAM/SQL-Server Performance“ [27] und in den Handbüchern zu ORACLE.

Hauptspeicher-Größe

Prüfen Sie im Zuge einer Migration, ob der Hauptspeicher zur Steigerung der Performance vergrößert werden muss:

- zur Vermeidung unnötiger Paging-I/Os
- zur Vergrößerung der Puffer für den OLTP-Betrieb
- zur Vergrößerung der Caches für das Softwareprodukt DAB

Betriebsart: native oder unter VM2000

Beachten Sie folgende Punkte, wenn im Zuge einer Migration der Native-Betrieb auf einen virtuellen Betrieb unter VM2000 umgestellt wird:

- Ein virtueller Betrieb verursacht Overhead, der bei der Leistungsberechnung des neuen Servers entsprechend zu berücksichtigen ist. Die Größe des Overheads ist abhängig von der Konfiguration, siehe [Seite 117](#).
- Beachten Sie bei der Konfiguration von VM2000 einige Randbedingungen, um unnötigen Overhead zu vermeiden. Siehe den [Abschnitt „Optimale Konfiguration von VM2000“ auf Seite 132](#).

Ausgangszustand

Der Ausgangszustand im momentanen Betrieb ist die Basis für die Betrachtungen. Dazu müssen die wichtigsten Anwendungsfälle wie OLTP-Betrieb oder kritische Batch-Anwendungen auf dem Ausgangs-Server mit openSM2 vermessen werden. Anschließend erfolgt die Auswertung der Ergebnisse und eine Prognose welcher Ziel-Server geeignet ist.

Generell ist bei einem Server-Wechsel zu beachten:

- Derzeitige CPU-Auslastung des Ausgangs-Servers (hoch / niedrig)
Im OLTP-Betrieb sollten die CPUs des Servers weniger als 70% ausgelastet sein.
Im Batch-Betrieb sind meist bis 100% CPU-Auslastung vertretbar.
- Einplanung des CPU-Mehrbedarfs bei BS2000/OSD-Versionswechsel
Der Mehrbedarf ist versionsabhängig. Im Mittel sind 1 - 2% pro Version anzusetzen.
- Berücksichtigung des eigenen Wachstumsbedarfs
Beispiel: jeweils 5% für 3 Jahre, ergibt etwa 15% Wachstumsbedarf.
- Einfluss von Neuaufnahmen oder Verlagerungen von Anwendungen
Wenn auf dem neuen Server Anwendungen wegfallen bzw. neu hinzukommen, dann muss das berücksichtigt werden.



Bei Bedarf wird empfohlen, das Angebot des BS2000 Optimization Service und des BS2000-Democenters zu nutzen:

- Performanceberatung bei der Migration
- Testmessungen von Kundenanwendungen auf der Ziel-Hardware

Zur Kontaktaufnahme wenden Sie sich bitte an Ihren Vertriebsbeauftragten.

3.4.1 Migration von S-Servern auf SQ-Server

Folgendes ist zu beachten:

- Mono-Leistung
Wenn von einem S-Server (Monoprozessor) auf einen SQ-Server (Multiprozessor) migriert werden soll, dann kann der Gesamt-RPF-Wert zwar gleich bleiben, die Monoprozessor-Leistung wird aber in den meisten Fällen kleiner sein. Dies ist zu berücksichtigen, z.B. bei Batch-Anwendungen, die nicht parallel betrieben werden können.
- CPU-Leistung: TU-Anteil
Wie im [Abschnitt „CPU-Leistung von SQ-Servern“ auf Seite 29](#) erklärt, sorgt die CISC-Firmware dafür, dass Kunden-Anwendungen auf SQ-Servern unverändert ablaufen. Der RPF-Wert des SQ-Servers gilt für einen TU-Anteil von 40 - 50% am Gesamt-CPU-Bedarf. Bei einem TU-Anteil ab 50% sollte ein zusätzlicher Leistungsbedarf von ca. 10% für den neuen SQ-Server vorgesehen werden.
- Peripherie: keine Kanäle
Auf SQ-Servern stehen keine Kanäle zur Verfügung. Aus Performance-Sicht ist eine Umstellung auf SAS oder Fibre Channel vorteilhaft.
Eine Datenmigration kann erforderlich sein, siehe Handbuch „Migration Guide“ [17].
- Peripherie: Networking
Es stehen die integrierten LAN-Controller des SQ-Servers zur Verfügung.
Ein HNC ist nicht erforderlich und kann auch nicht verwendet werden.
- Speicherbedarf
In aller Regel ist der Standard-Hauptspeicherausbau der SQ-Server ausreichend.
Zu Ausnahmen siehe den [Abschnitt „Hauptspeicher auf SQ-Servern“ auf Seite 32](#).
- Objektformat
System-Exit-Routinen müssen bereinigt und neu übersetzt werden, siehe Handbuch „Migration Guide“ [17].

Beispiel: Berechnung des RPF-Werts für einen Ziel-SQ-Server

Für den bisher genutzten S-Server sollen folgende Annahmen gelten:

- S155-10A (204 RPF)
- Last: OLTP-Betrieb (UTM/SESAM)
- CPU-Auslastung 80% (davon 50% TU)
- BS2000/OSD V8.0
- Wachstumsbedarf 15%

Damit ergibt sich folgende Rechnung für den Ziel-SQ-Server:

$$204 \text{ RPF} * 1.1 \text{ (erhöhter Leistungsbedarf wegen TU-Anteil 50\%)} \\ * 1.02 \text{ (Übergang von BS2000/OSD V8.0 nach V9.0)} = 229 \text{ RPF}$$

Um die Auslastung trotz Wachstum bei den für TP-Betrieb empfohlenen 65% zu halten:

$$229 \text{ RPF} * 1.2 \text{ (Ziel-Auslastung 65\%)} * 1.15 \text{ (Wachstumsbedarf)} = 316 \text{ RPF}$$

D.h. ein Server SQ210-20F mit 320 RPF wäre geeignet.

Zu beachten ist dabei der Wechsel von einem Monoprozessor auf einen Multiprozessor mit zwei CPUs.

In seltenen Fällen können anwendungsabhängige Sondereinflüsse die CPU-Leistung mindern, z.B.:

- intensive Nutzung von Dezimal-, Floatingpoint- oder EX-Befehlen
- selbstmodifizierender Code
- Code und Daten gemischt auf derselben Speicherseite
- Ausrichtungsverstöße
- sehr häufige SVC-Aufrufe

3.4.2 Migration von SX-Servern auf SQ-Server

Folgendes ist zu beachten:

- Mono-Leistung
Wenn von einem SX-Server (Monoprozessor) auf einen SQ-Server migriert werden soll, dann kann die Monoprozessor-Leistung auf jeden Fall wieder bereitgestellt werden.
- CPU-Leistung: TU-Anteil
Wie im [Abschnitt „CPU-Leistung von SQ-Servern“ auf Seite 29](#) erklärt, sorgt die CISC-Firmware dafür, dass Kunden-Anwendungen auf SX und SQ-Servern unverändert ablaufen. Der RPF-Wert des SQ-Servers gilt für einen TU-Anteil von 40 - 50% am Gesamt-CPU-Bedarf.
- Peripherie: keine Kanäle
Auf SQ-Servern stehen keine Kanäle zur Verfügung. Aus Performance-Sicht ist eine Umstellung auf SAS oder Fibre Channel vorteilhaft.
Eine Datenmigration ist normalerweise nicht erforderlich, siehe Handbuch „Migration Guide“ [\[17\]](#).
- Peripherie: Networking
Es stehen die integrierten LAN-Controller des SQ-Servers zur Verfügung.
Ein HNC ist nicht erforderlich und kann auch nicht verwendet werden.
- Peripherie: LTO-Laufwerk
Mit dem im SQ-Server eingebauten LTO-n-Laufwerk können LTO-(n-1)-Datenträger vom Ausgangssystem weiter verwendet werden. LTO-(n-2)-Datenträger können nur noch gelesen werden und sollten daher baldmöglichst auf LTO-n-Datenträger migriert werden. Sicherungsbestände auf LTO-(n-3)-Datenträgern können mit LTO-n-Laufwerken nicht mehr gelesen werden. Hier ist eine Migration im Ausgangssystem vorzusehen.
- Speicherbedarf
In aller Regel ist der Standard-Hauptspeicherausbau der SQ-Server ausreichend.
Zu Ausnahmen siehe den [Abschnitt „Hauptspeicher auf SQ-Servern“ auf Seite 32](#).
- Objektformat
System-Exit-Routinen müssen bereinigt und neu übersetzt werden, siehe Handbuch „Migration Guide“ [\[17\]](#).

Beispiel: Berechnung des RPF-Werts für einen Ziel-SQ-Server

Für den bisher genutzten SX-Server sollen folgende Annahmen gelten:

- SX160-10C (105 RPF)
- Last: OLTP-Betrieb (UTM/SESAM)
- CPU-Auslastung 80% (davon 50% TU)
- BS2000/OSD V7.0
- Wachstumsbedarf 15%

Damit ergibt sich folgende Rechnung für den Ziel-SQ-Server:

$$105 \text{ RPF} * 1.04 \text{ (Übergang von BS2000/OSD V7.0 nach V9.0)} * 1.15 \text{ (Wachstumsbedarf)} \\ = 125 \text{ RPF}$$

D.h. ein Server SQ210-10E mit 125 RPF wäre geeignet.

4 Leistungsverhalten der Peripherie

4.1 Eigenschaften externer Plattenspeichersysteme

Dieser Abschnitt beschreibt die performance-relevanten Eigenschaften folgender externer Plattenspeichersysteme, soweit sie zum Verständnis ihres Leistungsverhaltens aus Sicht von BS2000/OSD notwendig sind:

- ETERNUS DX
- Symmetrix DMX und VMAX

4.1.1 Komponenten von Plattenspeichersystemen

Die wichtigsten, performance-relevanten Komponenten der Plattenspeichersysteme sind:

- Server-Anschluss
- Cache
- Platten
- Controller

Server-Anschluss

Plattenspeichersysteme ETERNUS DX oder Symmetrix werden an einen BS2000/OSD-Server angeschlossen über Kanal Typ FC (S-Server, Leistung 1 Gbit/s) bzw. FC-Anschluss (SQ-Server, Leistung bis 8 Gbit/s)

Die Anzahl benötigter Anschlüsse ist vom Ausbau des Plattenspeichersystems abhängig. Der Anschluss über mehrere Pfade muss berücksichtigt werden. Aus Performancesicht sind Kanäle vom Typ FC bzw. FC-Anschlüsse uneingeschränkt zu empfehlen.

Die Leistung der Kanäle bzw. der I/O-Prozessoren bei SQ-Servern ist im [Kapitel „Leistungsverhalten der Server“ auf Seite 25](#) beschrieben.

Cache

Die Komponente eines Plattenspeichersystems mit dem größten Einfluss auf die Performance ist der Cache. Jede Ein-/Ausgabe-Anforderung unterliegt dem Caching. Die Hardware-Bedienzeit (I/O-Zeit) hängt wesentlich davon ab, ob sich die Daten im Cache befinden oder ob sie erst von der Platte gelesen bzw. dorthin transferiert werden müssen.

Falls sich beim Lesen der gesuchte Datenblock nicht im Cache befindet, muss er von den Platten gelesen werden (Read-Miss). Ein solcher Read-Miss kostet erheblich mehr Zeit als ein Read-Hit, bei dem sich die Daten schon im Cache befinden. Beim Erkennen sequenzieller Verarbeitung erfolgt ein Lesen von Daten im Voraus (Read Ahead), um einen Read-Miss zu vermeiden.

Da der Cache der Plattenspeichersysteme durch Batterien gegen Stromausfall gesichert ist, gelten Schreib-Aufträge als beendet, wenn sie in den Cache geschrieben und verifiziert sind (Write-Hit-Rate = 100 %). Eine Schreib-Ein-/Ausgabe dauert also annähernd so lange wie ein Read-Hit. Das Schreiben auf Platte erfolgt asynchron zur Ein-/Ausgabe. Falls im Cache nicht genügend Platz ist, treten so genannte „Delayed Fast Writes“ auf, d.h. Cache-Inhalte müssen zuerst gesichert werden, bevor die Einlagerung neuer Daten in den Cache möglich ist (Write-Hit-Rate < 100 %). Dies tritt typischerweise bei der Rekonstruktion großer Datenmengen auf.

Als Hit-Rate wird der Anteil der Cache-Hits an der Gesamtzahl der Ein-/Ausgaben auf das Plattenspeichersystem bezeichnet. Teilweise wird noch unterschieden zwischen der Read-Hit-Rate und der Total-Hit-Rate. In der Total-Hit-Rate sind die Cache Write-Hits enthalten. In diesem Handbuch werden nur Read-Hit-Raten betrachtet.

Hersteller externer Plattenspeichersysteme geben oft die „Total-Hit-Rate“ an, die auch die Cache Write-Hits beinhaltet und damit höher ist als die hier betrachtete Read-Hit-Rate.

Durch den Einsatz großer Caches kann auch für Lese-Anforderungen eine hohe Hit-Rate (> 90 %) erreicht werden. Eine gute Leistung eines Plattenspeichersystems wird normalerweise ab einer Read-Hit-Rate von 80 % erreicht.

Für Performance-kritische Anwendungen sind große Cache-Ausbauten in den Plattenspeichersystemen ETERNUS DX und Symmetrix unbedingt zu empfehlen.

Platten

Von physikalischen Platten sind meist bekannt:

- Speichertyp (HD/SSD)
- Speicherkapazität (in Gbyte, z.B. 300 Gbyte)
- Umdrehungsgeschwindigkeit (in „rotations per minute“, z.B. 15.000 rpm).

Weiterhin sind für die Leistung einer Platte bestimmend:

- Interner Anschluss mit Anschlussart (z.B. Fibre Channel, SAS, Serial ATA (SATA), Nearline-SAS (NL-SAS)) und Leistung (z.B. 8 Gbit/s)
- Cache auf dem Laufwerk (Laufwerks-Cache)

Ein Anwender sieht die Leistung der physikalischen Platten in der Regel nur bei einem Read-Miss.

Aus Sicht von BS2000/OSD bestehen bei den heute angebotenen Plattengrößen keine Einschränkungen für deren Verwendung.



Die Daten performance-kritischer Anwendungen sollten **nicht** auf SATA- oder NL-SAS-Platten gelegt werden.

Controller

Die Controller der Plattenspeichersysteme bearbeiten die Ein-/Ausgabe-Anforderungen des Servers, steuern den Cache-Zugriff und wickeln den Zugriff zu den physikalischen Plattenlaufwerken ab.

Die Anzahl der Controller sowie die Anzahl der Wege zu den bedienten Servern und zu den internen Platten sind modellabhängig.

4.1.2 Konfigurierung von Plattenspeichersystemen

Dieser Abschnitt beschreibt die performance-relevanten Aspekte der Konfigurierung eines Plattenspeichersystems soweit sie aus Sicht von BS2000/OSD wichtig sind.

4.1.2.1 RAID-Gruppen und LUNs

Mehrere Platten werden bei der Konfigurierung zu RAID-Gruppen zusammengefasst. Diese RAID-Gruppen haben einen einheitlichen RAID-Level (RAID1, RAID1/0, RAID5, RAID6).

Auf diesen RAID-Gruppen werden so genannte Logical Unit Numbers (LUNs) definiert. Es ist üblich, dass mehrere LUNs (logische Volumes) auf derselben physikalischen Platte eingerichtet werden. Zu den Auswirkungen logischer Volumes auf die Performance siehe [„Logische Volumes und physikalische Platten“ auf Seite 52](#).

Die Konfigurierung im BS2000/OSD verwendet diese LUNs mit LUN-Nummern und Plattentyp (z.B. D3435).

4.1.2.2 RAID-Level

Alle Plattenspeichersysteme ETERNUS DX und Symmetrix sind hochverfügbare Plattenspeichersysteme, die neben den ursprünglich von der Berkeley University definierten RAID-Standards auch alle vom „RAID Advisory Board“ festgelegten Verfügbarkeitsstufen erfüllen. Sie unterscheiden sich lediglich in den Ausbaumöglichkeiten. Der RAID-Level bezeichnet die Methode zur Bildung fehlertoleranter und performanter Plattenspeichersysteme. Bei den Plattenspeichersystemen für BS2000/OSD sind die folgenden RAID-Level üblich:

- RAID1 (full mirror)
- RAID1/0 (striping + full mirror)
- RAID5 (striping + parity mirror)
- RAID6 (striping + dual parity mirror)

RAID1

Jeweils zwei identische Plattenlaufwerke, Original- und Spiegelplatte, bilden eine RAID1-Gruppe (Kurzbezeichnung im Folgenden „RAID1(1+1)“).

Sie enthalten identische Daten. Damit stehen für den Anwender nur 50% der Plattenkapazität zur Verfügung. Die Schreibleistung ändert sich durch die Spiegelung nicht. Bei Leseaufträgen versucht das Plattenspeichersystem zu optimieren und von der günstigeren Platte zu lesen. Bei Ausfall einer Platte kann bis zum Ersatz der Platte weitgehend ohne Performanceverlust weitergearbeitet werden.

RAID1/0

Gegenüber RAID1 erfolgt bei RAID1/0 eine zusätzliche Verteilung der Daten über mehrere Platten in Form des „data striping“. Eine RAID1/0-Gruppe besteht also aus mehreren, oft 4 Platten, auf denen die Originaldaten mit „data striping“ verteilt sind. Dazu kommen ebenso viele Platten mit den gespiegelten Daten (Kurzbezeichnung im Folgenden z.B. „RAID10(4+4)“).

Auf Symmetrix-Systemen ist für RAID1/0 auch der Begriff „Meta Volumes“ üblich. Wie bei RAID1 steht dem Anwender nur 50% der installierten Plattenkapazität zur Verfügung. Allerdings bietet die Datenspiegelung mit RAID1/0 durch Erhöhung der Parallelität beim Datenzugriff und durch gleichmäßigere Auslastung der Platten die beste Performance. Bei RAID1/0 können bei geringem Performanceverlust sogar mehrere Platten ausfallen, solange nicht Originalplatte und Spiegelplatte gleichzeitig ausfallen.

RAID1/0 wird für besonders hohe Anforderungen an die Schreib-Performance empfohlen.

RAID5

RAID5 realisiert eine gemeinsame Parity-Sicherung für mehrere Plattenlaufwerke. Die Daten werden wie bei RAID1/0 mit „data striping“ in Blöcken über die Plattenlaufwerke der RAID5-Gruppe verteilt und mit der Parity-Information, ebenfalls verteilt über alle Laufwerke, gesichert („rotating parity“). Da beim Schreiben zusätzlich die Parity-Informationen berechnet und geschrieben werden müssen, ist RAID5 etwas aufwändiger als RAID1/0. Durch das „data striping“ bietet es aber wie RAID1/0 die Vorteile bei parallelen Zugriffen.

Eine RAID5-Gruppe besteht häufig aus 4 Platten, dabei stehen drei Viertel der Gesamtkapazität für Benutzerdaten zur Verfügung, Ein Viertel der Kapazität wird für die Parity-Informationen genutzt (Kurzbezeichnung im Folgenden „RAID5(3+1)“). Beim Ausfall einer Platte ist mit verschlechterter Performance zu rechnen als bei RAID1/0.

RAID6

Wie bei RAID5 wird der Datenstrom in Stripes unterteilt, jedoch werden nicht ein, sondern zwei Fehlerkorrekturwerte berechnet. Daten und Parity-Informationen werden so über die Platten verteilt, dass beide Parity-Informationen auf unterschiedlichen Platten liegen („dual rotating parity“). Bei RAID6 können bis zu zwei Platten ausfallen. Der Aufwand für die Resynchronisation, insbesondere bei zwei ausgefallenen Platten, ist jedoch erheblich höher als bei RAID5.

4.1.2.3 Replikationsfunktionen externer Plattenspeichersysteme

Externe Plattenspeichersysteme bieten Funktionen zur lokalen Replikation innerhalb eines Plattenspeichersystems sowie Funktionen zur entfernten (remote) Replikation zwischen räumlich getrennten Plattenspeichersystemen.

Für die lokale Replikation stehen folgende Funktionen zur Verfügung:

- für ETERNUS DX die Funktion EC (Equivalent Copy)
- ab ETERNUS DX S2 zusätzlich die Funktion SnapOPC+
- für Symmetrix die TimeFinder-Produktfamilie

Für remote Replikationen stehen folgende Funktionen zur Verfügung:

- für ETERNUS DX die Funktion REC (Remote Equivalent Copy)
- für Symmetrix die SRDF-Produktfamilie

Das Softwareprodukt SHC-OSD ist die BS2000-Host-Komponente für externe Plattenspeichersysteme. Es stellt Informationsdienste und Kommandos zur Steuerung der Replikationsfunktionen der Plattenspeichersysteme zur Verfügung. Eine ausführliche Beschreibung der Replikationsfunktionen ist im Handbuch „SHC-OSD“ [28] enthalten.

Leistungsverhalten mit REC / SRDF

Wird eine Kopie der Daten in einem entfernten zweiten Plattenspeichersystem geführt, dann wird jeder Schreibzugriff zuerst in den lokalen Cache und anschließend in den „Remote-Cache“ eingetragen.

Das bedeutet, dass sich durch den Einsatz von SRDF die Hardware-Bedienzeit für Schreibzugriffe wenigstens verdoppelt. Zusätzlich ist eine erhöhte Protokollzeit zur Bearbeitung der Funktion zu berücksichtigen.

Ein weiterer wesentlicher Faktor ist die Entfernung zwischen den Standorten. Es ist jeweils die 2-fache Laufzeit des Lichts in Glas zu addieren (Hin- und Rückweg).

Üblicherweise beträgt die Write-Hit-Rate 100% (sofern genügend zeitliche „Erholungsphasen“ zum Sichern der Cache-Inhalte vorliegen), d.h. jedes Schreiben wird schnell durchgeführt. Bei kleinen Blöcken fällt die Verdoppelung der Hardware-Bedienzeit i.d.R. nicht ins Gewicht, bei großen Blöcken (z.B. Kopiervorgänge) ist die Verlängerung der Hardware-Bedienzeit deutlich.

Eine zusätzliche Verlängerung der Hardware-Bedienzeit kann eintreten, wenn die Leistungsfähigkeit der „Remote Data Links“ nicht ausreichend ist.



Bei asynchroner Replikation (Symmetrix: SRDF/A) fällt nur die „Protokollzeit“ an. Die Ein-/Ausgabe wird also wesentlich weniger verzögert.

Details siehe bei „[Auslastung der Ein-/Ausgabe-Peripherie](#)“ auf Seite 292.

4.1.2.4 Thin Provisioning

Die Funktion „Thin Provisioning“ erlaubt die effiziente Nutzung der Kapazität von Plattenspeichersystemen. Der Anwendung werden Geräte (LUNs) mit vorkonfigurierter virtueller Kapazität angeboten, während das Plattenspeichersystem intern physikalisch die jeweils benötigte Kapazität bereitstellt. Thin Provisioning erfordert eine entsprechende Konfiguration im Plattenspeichersystem.

SHC-OSD unterstützt Thin Provisioning für die Plattenspeichersysteme ab ETERNUS DX S2 und für Symmetrix (dort als „Virtual Provisioning“ bezeichnet). SHC-OSD verfügt über Überwachungs- und Informationsfunktionen für Thin Provisioning.

Eine ausführliche Beschreibung der Funktion „Thin Provisioning“ ist im Handbuch „SHC-OSD“ [28] enthalten.

4.1.3 Konfigurierung in BS2000/OSD

Logische Volumes

Einzelne Platten sind auf den aktuellen Plattenspeichersystemen nur mehr bedingt sichtbar. BS2000/OSD sieht „logische Volumes“, die den LUNs (Logical Unit Numbers) zugeordnet sind. Die Zuordnung von LUNs zu physikalischen Platten ist für BS2000/OSD weitgehend verborgen. Die Plattenspeichersysteme sorgen, gesteuert über die RAID-Level, für eine performante Bedienung.

Es können auch „große“ logische Volumes verwendet werden. Das sind Volumes mit einer Größe von mehr als 32 Gbyte, siehe Handbuch „Dateien und Volumes größer 32 GB“ [3]. Für einen performanten Betrieb sind dann die Empfehlungen im [Abschnitt „Performerter Betrieb“ auf Seite 55](#) zu beachten.

Logische Volumes und physikalische Platten

Aus Sicht von BS2000/OSD sind logische Volumes voneinander unabhängige Einheiten, auf die parallel Ein-/Ausgaben eingeleitet werden können. Bei Einsatz der aktuellen, hochkapazitiven Platten befinden sich auf einem Plattenlaufwerk normalerweise mehrere logische Volumes.

Wenn Ein-/Ausgaben auf logische Volumes, die sich auf demselben Plattenlaufwerk befinden, parallel eingeleitet werden, dann tritt zwangsläufig eine Verlängerung der Hardware-Bedienzeit auf. Ob die Verlängerung der Hardware-Bedienzeit zu einer spürbaren Performance-Einbuße führt, ist abhängig von der Cache-Hit-Rate und der Auslastung des Plattenlaufwerks – möglicherweise auch durch Nutzer außerhalb von BS2000/OSD.

Bei physikalischen Platten, die gemeinsam von mehreren Programmen benutzt werden, liegt normalerweise keine zeitliche Gleichverteilung der Ein-/Ausgabe-Anforderungen vor. Bei höheren Gesamt-Auslastungen führt dies entsprechend den Gesetzen über Warteschlangenbildung zu untragbar hohen Wartezeiten vor den Platten, d.h. die Software-Bedienzeit für jede Ein-/Ausgabe ist wesentlich größer als die Hardware-Bedienzeit (siehe auch den [Abschnitt „Ablauf einer Ein-/Ausgabe“ auf Seite 265](#)). Zur Minimierung dieser Wartezeiten sollte die Auslastung gemeinsam benutzter Platten 30 % nicht überschreiten. Dieser Forderung liegt die Annahme zugrunde, dass die Wartezeit vor der Platte generell nicht höher als ein Drittel der Hardware-Bedienzeit sein sollte.

Anforderungen an die Konfigurierung des Plattenspeichersystems

Aus Sicht von BS2000/OSD kann die Auslegung der Plattenperipherie unter verschiedenen Gesichtspunkten erfolgen:

- hohe Kapazität (statischer Speicherplatz im Plattenspeichersystem, in GByte)
- hoher Durchsatz (in Mbyte/s)
- kurze Ein-/Ausgabezeiten (in ms pro Ein-/Ausgabe)

Da BS2000/OSD keine Kenntnis über die Zuordnung von logischen Volumes zu physikalischen Platten hat, müssen bei der Generierung des Plattenspeichersystems gewisse Performance-Anforderungen berücksichtigt werden. Insbesondere sind das

- Größe der Volumes
- Anzahl der Ein-/Ausgabe-Pfade
- RAID-Level
- Art der Server-Last
(OLTP-Betrieb, durchsatz-orientierter Batch-Betrieb - die Hauptanwendung muss berücksichtigt werden)
- Ein-/Ausgabelast des jeweiligen Volumes
(Ein-/Ausgaben pro Sekunde oder allgemeine Aussage: niedrig / hoch - die Spitzenlast muss berücksichtigt werden)

Die konkrete Konfigurierung des Plattenspeichersystems kann bzw. muss in vielen Fällen dem Service überlassen werden. Insbesondere muss vermieden werden, dass mehrere, hochbelastete logische Volumes auf dieselbe physikalische Platte gelegt werden. Bei Mischkonfigurationen mit anderen Systemen sollte auf die speziellen Eigenschaften des BS2000-Betriebs hingewiesen werden (z.B. sehr hohe und sehr gleichmäßige Last, die aber auch eine zuverlässige Performance der Peripherie voraussetzt).

Datenformat

Es wird empfohlen, das NK-Datenformat zu verwenden. Gegenüber dem K-Datenformat, bei dem auch die Schlüsselinformationen mitgeführt werden müssen, nutzt das NK-Datenformat die Plattenkapazität und den Cache-Speicher besser aus und erreicht damit kürzere Hardware-Bedienzeiten.

Messungen

Mit SHC-OSD können Sie Folgendes ermitteln:

- welche logischen Volumes sind auf einer Platte eingerichtet (nicht bei Thin Provisioning)
- auf welcher oder welchen physikalischen Platte(n) liegt ein BS2000-Volume (nicht bei Thin Provisioning)
- Für ETERNUS DX: in welcher RAID-Gruppe befindet sich ein BS2000-Volume
- Für ETERNUS DX: welche weiteren Volumes befinden sich in derselben RAID-Gruppe

Wenn das Softwareprodukt SHC-OSD installiert ist, dann werden in der Reportgruppe `STORAGE-SYSTEM-SYMMETRIX` von openSM2 Messwerte über Symmetrix-Systeme erfasst, z.B. die Anzahl der Lese- und Schreibvorgänge sowie die Anzahl der Read-Hits und damit die Read-Hit-Rate. Auch für physikalische Geräte können Informationen, z.B. der Durchsatz, ermittelt werden. Weitere Informationen dazu finden Sie im [Kapitel „Einsatz des Mess-Systems openSM2“ auf Seite 271](#).

Zur Überwachung der Plattenspeichersysteme ETERNUS DX benötigen Sie das Softwareprodukt openSM2 (Open Systems).

Die Laufwerksauslastung auf ETERNUS DX ist mit BS2000-Mitteln nicht zugänglich und kann daher mit openSM2 nicht gemessen werden. Sie setzt sich aus der Summe der Ein-/Ausgabe-Lasten der jeweiligen logischen Volumes zusammen.

4.1.4 Performanter Betrieb

Voraussetzung für den performanten Betrieb von Plattenspeichersystemen ist der Server-Anschluss über Kanal Typ FC bzw. einen FC-Anschluss.

Ein erheblicher Leistungszuwachs kann zudem durch die Nutzung der BS2000-Funktion Parallel Access Volume (PAV) auf S-Servern bzw. der Funktion Remote System Call (RSC) auf SQ-Servern (standardmäßig verwendet) erreicht werden. Um das Potenzial von PAV nutzen zu können, müssen geeignete RAID-Levels gewählt werden.

Anschlussart Fibre Channel

Voraussetzung für eine performante Bedienung jeder Peripherie ist ein Server-Anschluss über Kanal Typ FC (S-Server) bzw. einen FC-Anschluss.

Mit Kanälen Typ S kann die Leistungsfähigkeit der aktuellen Peripheriegeräte nicht ausgenutzt werden.

Parallel Access Volume (PAV, DPAV)

Beim einfachen Plattenzugriff (Standard) ist auf S-Servern zur gleichen Zeit nur ein Ein-/Ausgabeauftrag auf ein logisches Volume in Bearbeitung. Weitere Aufträge werden in BS2000/OSD serialisiert.

Alternativ dazu wird mit der BS2000-Funktion PAV ein paralleler Plattenzugriff an S-Servern unterstützt. Mit PAV werden mehrere Geräteadressen (Alias-Geräte) einem logischen Volume (Basis-Gerät) zugeordnet. Damit können mehrere Ein-/Ausgabeaufträge parallel auf ein Volume eingeleitet werden. Im Plattenspeichersystem können dann mehrere Cache-Hits gleichzeitig bedient werden, während parallel dazu bei einem Cache-Miss ein physikalischer Plattenzugriff durchgeführt wird. Nähere Informationen zu PAV finden Sie im Handbuch „Systembetreuung“ [10].

Alias-Geräte am Kanal Typ FC müssen in BS2000/OSD generiert werden. In der Steuerung ist kein Eingriff erforderlich.

„Statisches PAV“ verlangt eine vorausschauende Planung auf die zukünftige Geräteauslastung. Den stark ausgelasteten Geräten muss im Voraus die richtige Anzahl von Alias-Geräten zugewiesen werden.

Beim „Dynamischen PAV“ für Kanal Typ FC weist das Dienstprogramm IORM automatisch den stark belasteten Volumes Alias-Geräte (DPAV-Geräte) zu, die am meisten davon profitieren. Die Generierung von Alias-Geräten entfällt. Ein-/Ausgabeengpässe werden dadurch entschärft.

Remote System Call (RSC)

Für SQ-Server gibt es mit der Funktion Remote System Call (RSC) für Platten eine dem PAV ähnliche Funktion. Bis zu sechs Ein-/Ausgaben sind gleichzeitig auf ein Volume bzw. Gerät möglich. Die Serialisierung erfolgt im Plattenspeichersystem. RSC wird standardmäßig genutzt, der Anwender muss **keine** besondere Konfigurationseinstellung für die Nutzung von RSC und für die Parallelisierung von Ein-/Ausgaben treffen.

PAV und RAID-Level

Durch den Einsatz von PAV bzw. RSC können bessere Ein-/Ausgabezeiten und höherer Durchsatz erwartet werden. Über die tatsächliche Performance-Verbesserung entscheidet der RAID-Level (siehe [Abschnitt „PAV und RAID-Level“ auf Seite 63](#)):

- Bessere Ein-/Ausgabezeiten werden bei allen RAID-Levels erreicht.
- Ein besserer Durchsatz wird nur erreicht, wenn der verwendete RAID-Level die Parallelität von PAV bzw. RSC auch nutzen kann. Dies ist vor allem bei RAID-Levels mit „data striping“ möglich, also RAID1/0, RAID5 und RAID6. Für RAID1 gilt dies nur eingeschränkt.

Für einen performanten Einsatz wird RAID1/0 oder, falls die Plattenkapazität sehr wichtig ist und bei nicht zu hoher Random-Schreiblast, RAID5 oder RAID6 empfohlen, auf S-Servern zusammen mit DPAV.

Empfehlungen für einen performanten Betrieb von Plattenspeichersystemen

- Anschluss nur über Kanal Typ FC bzw. einen FC-Anschluss
- Möglichst neue Plattenspeichersysteme verwenden:
 - mit neuestem Firmware-Stand
 - mit ausreichend großem Cache-Speicher (muss so groß gewählt werden, dass bei OLTP-Betrieb eine Read-Hit-Rate von mindestens 80% erreicht wird)
 - mit schnellen Platten (mindestens 10.000 rpm)
 - große Platten können verwendet werden
- RAID-Level RAID1/0 oder, bei nicht zu hoher Random-Schreiblast, RAID5 oder RAID6
 - bei S-Servern Einsatz von PAV, besser DPAV
 - dann können auch „große“ Volumes (mit mehr als 32 Gbyte) verwendet werden
- Blockgröße 160 KB pro Ein-/Ausgabe bei durchsatz-orientierten Anwendungen, falls möglich
- allgemeine Erhöhung der Parallelität (im Betriebssystem, in Dienstprogrammen und Anwendungen)

4.1.5 Lasttypen

Anwendungen

Für die Auslegung der Plattenperipherie sind sowohl das Lastprofil als auch die zeitlichen Anforderungen der Anwendung maßgebend. In der Praxis treten im Wesentlichen zwei Lasttypen auf, die sich folgendermaßen charakterisieren lassen:

	OLTP-Betrieb (Datenbank-Anwendungen)	Anwendungen mit großer Blockgröße (Batch-Verarbeitung, Datensicherung)
typische Blockgröße pro Ein-/Ausgabe	„klein“: 2 KB oder 4 KB	„groß“: 16 KB oder größer
Lasttyp	Ein-/Ausgaberate (Ein-/Ausgaben/s)	Durchsatz (Mbyte/s)
typische Ein-/Ausgabe-Operationen	Random-Zugriff	Random-Zugriff sequenzieller Zugriff

Anwendungen in BS2000/OSD können Blockgrößen bis zu 160 KB pro Ein-/Ausgabe verwenden. Bestimmte Plattentypen erlauben sogar Blockgrößen bis zu 480 KB. Der Durchsatz kann damit erheblich gesteigert werden.

Anwendungen können die Performanceverbesserungen mit PAV bzw. RSC und entsprechenden RAID-Levels nur dann realisieren, wenn sie Anforderungen parallel und ohne wesentliche gegenseitige Beeinträchtigung bearbeiten können. Dies ist beim OLTP-Betrieb in der Regel der Fall. Auch die aktuellen Versionen der Datensicherungsprodukte HSMS/ARCHIVE und FDDRL sind dafür vorbereitet. Im Batch-Betrieb können die Performanceverbesserungen nur bei parallelen Ein-/Ausgaben realisiert werden.

Je nach Anwendungsart stehen unterschiedliche Performance-Ziele im Vordergrund. Bei der Datensicherung und im Batch-Betrieb wird ein hoher Durchsatz angestrebt, beim OLTP-Betrieb werden gute Antwortzeiten erwartet. Einen wichtigen Anteil an Durchsatz bzw. Verweilzeit und Antwortzeit bzw. Transaktionszeit haben die Ein-/Ausgaben (siehe [Abschnitt „Kenngrößen für die Beschreibung der Leistung eines IT-Systems“ auf Seite 15](#)). Im Folgenden werden die Ein-/Ausgabezeiten aus Sicht des Anwenders („Software-Bedienzeit“) und die reine Zeit für die Hardware-Bedienung („Hardware-Bedienzeit“) betrachtet (siehe [Abschnitt „Ablauf einer Ein-/Ausgabe“ auf Seite 265](#)). Die „Software-Bedienzeit“ setzt sich im Wesentlichen zusammen aus der Hardware-Bedienzeit und den Wartezeiten bis zur Einleitung der Ein-/Ausgabe.

Standardisierte Lasttypen

Für die Messungen in den folgenden Abschnitten wurden standardisierte Lastfälle verwendet:

- „Random25“
Random-Zugriffe mit einem Anteil von 25% Schreiben und 75% Lesen.
Dieser Fall ist bei kurzen Blöcken einem OLTP-Betrieb nachempfunden.
- „Sequential write“, „Sequential read“
Sequenzielles Schreiben bzw. Lesen.
Dies ist bei großen Blöcken einer Batch-Verarbeitung nachempfunden.

4.2 Plattenspeichersysteme an S-Servern

Dieser Abschnitt enthält Messergebnisse und Folgerungen für die Performance der Plattenspeichersysteme ETERNUS DX und Symmetrix bei Anschluss an S-Server über Kanäle Typ FC.

4.2.1 ETERNUS DX

Dieser Abschnitt enthält Messergebnisse und Folgerungen für die Performance des Plattenspeichersystems ETERNUS DX8400 bei Anschluss an S-Server über Kanäle Typ FC.

Inzwischen steht die neue Generation ETERNUS DX8700 S2 zur Verfügung. Diese Generation zeichnet sich aus durch deutlich größere Caches und spezielle Performance-Optimierungen für einen besseren I/O-Durchsatz bei sequenzieller Verarbeitung von Daten (Durchsatzsteigerung bis zu 300%). Die Performance bei Random-Verarbeitung ist vergleichbar mit derjenigen von ETERNUS DX8400.

ETERNUS DX8700 S2 ist in den dargestellten Messungen noch nicht berücksichtigt.

Bei den folgenden Messungen wurde folgende Konfiguration verwendet:

- Server S200-20 unter VM2000 (VMs mit Privileg IO-PRIORITY)
- Anschluss des S-Servers an das Plattenspeichersystem über 4 Kanäle Typ FC mit einer Leistung von je 1 GBit.
- Plattenspeichersystem ETERNUS DX8400 mit 32 GB Cache und physikalischen Platten mit je 450 GB und 15.000 rpm
- 16 Volumes auf Platten des Typs D3435 mit Datenformat NK2 mit geringer gegenseitiger Beeinflussung (wenige logische Volumes auf einem physikalischen Gerät)

Es wird empfohlen, die Plattenspeichersysteme ETERNUS DX mit RAID1/0 zu betreiben. Die Messungen unterschiedlicher RAID-Varianten zeigen die besten Ergebnisse für RAID 1/0.

4.2.1.1 Durchsatz mit einer Task

Die folgenden beiden Tabellen zeigen den erzielten Durchsatz (die Ein-/Ausgaberate in Ein-/Ausgaben pro Sekunde (IO/s) und den Durchsatz in Mbyte/s), den eine Anwendertask bei Plattenzugriffen zum Plattenspeichersystem ETERNUS DX8400 erreichen kann.

Der Durchsatz wurde für die standardisierten Lasttypen bei verschiedenen Blockgrößen und einer Read-Hit-Rate von 100% gemessen:

Eine Task: Ein-/Ausgaben/s	Blockgröße (KB)			
	2	16	32	160
„Random25“	3.650	2.230	1.565	450
„Sequential read“	3.870	2.325	1.620	465
„Sequential write“	3.140	2.010	1.385	420

Eine Task: Mbyte/s	Blockgröße (KB)			
	2	16	32	160
„Random25“	7,3	36	50	72
„Sequential read“	7,7	37	42	74
„Sequential write“	6,3	32	44	67

4.2.1.2 Durchsatz mit bis zu vier Kanälen Typ FC

Die folgenden beiden Tabellen zeigen den erzielten Durchsatz (die Ein-/Ausgaberate in Ein-/Ausgaben pro Sekunde (IO/s) und den Durchsatz in Mbyte/s) bei Plattenzugriffen zum Plattenspeichersystem ETERNUS DX8400 bei bis zu vier Kanälen Typ FC.

Der Durchsatz wurde für den Lasttyp „Random25“ bei verschiedenen Blockgrößen und einer Read-Hit-Rate von 100% gemessen:

„Random25“: Ein-/Ausgaben/s	Blockgröße (KB)			
	2	16	32	160
1 Kanal Typ FC	7.175	5.410	3.140	710
4 Kanäle Typ FC	28.725	20.885	12.305	2.800

„Random25“: Mbyte/s	Blockgröße (KB)			
	2	16	32	160
1 Kanal Typ FC	13	84	98	111
4 Kanäle Typ FC	56	326	384	437

Kanäle Typ FC skalieren sehr gut. Das heißt, mit n Kanälen Typ FC kann annähernd der n-fache Durchsatz erreicht werden. Voraussetzung dafür ist eine ausreichende Leistung der BS2000-CPU's und der Peripherie.

Verglichen mit dem Durchsatz des Plattenspeichersystems Symmetrix VMAX (siehe [Seite 72](#)) liegen diese Werte darunter. Dies erklärt sich dadurch, dass bei den Messungen auf Symmetrix VMAX ein Cache verwendet wurde, der im Vergleich zu den Messungen auf ETERNUS DX8400 3-mal so groß war.

In der Last „Random25“ finden 25% Schreib-I/Os und 75% Lese-I/Os statt. Angesichts eines deutlich größeren Caches können damit im eingeschwungenen Zustand (nahezu) alle Lese-I/Os aus dem Cache befriedigt werden und bewirken damit kürzere I/O-Zeiten und einen größeren Durchsatz.

Dieser Effekt ist auch mit den größeren Caches der ETERNUS DX8700 S2 zu erwarten.

4.2.1.3 Hardware-Bedienzeiten

Hardware-Bedienzeiten beim Lesen

Folgende Tabelle zeigt die Ein-/Ausgabezeiten (Hardware-Bedienzeiten) für Read-Hit und Read-Miss, wie sie in üblichen Konfigurationen und bei üblichem Lastaufkommen erreicht werden. Die Messungen beziehen sich auf den RAID-Level RAID1/0 (siehe die Empfehlung auf [Seite 59](#)).

Die Hardware-Bedienzeiten sind neben der Ein-/Ausgaberate auch abhängig vom Ausbau des Plattenspeichersystems. Bei kleinen Konfigurationen können die Zeiten etwas günstiger sein. Bei sehr großen Konfigurationen können die Zeiten schlechter sein.

Ein-/Ausgabezeit Lesen (ms)	Blockgröße (KB)			
	2	16	32	160
Read-Hit	0,3	0,5	0,7	2,3
Read-Miss RAID1/0	4,8	5,3	5,8	10

Diese Zeiten gelten für eine durchschnittliche Last, bei der weder Platten noch Kanäle zu hoch ausgelastet sind.

Für andere RAID-Level wurden folgende Unterschiede gemessen:

- RAID1: ca. 5% besser
- RAID5: ca. 10% schlechter
- RAID6: ca. 10% (lange Blöcke) bis 15% (kurze Blöcke) schlechter

Hardware-Bedienzeiten bei OLTP-Betrieb

Die Ein-/Ausgabezeiten bei einer typischen OLTP-Last („Random25“) sind bei hohen Read-Miss-Raten meist um ca. 10% schlechter als die Ein-/Ausgabezeiten beim Lesen (siehe vorigen Abschnitt).

4.2.1.4 PAV und RAID-Level

Für Plattenspeichersysteme wird neben einem Anschluss über Kanal Typ FC für S-Server der Einsatz von PAV/DPAV (Dynamic Parallel Access Volume) zusammen mit einem geeigneten RAID-Level empfohlen. Dieser Abschnitt zeigt, wie sich die Ein-/Ausgabezeiten und der Durchsatz (Ein-/Ausgaberate) durch den Einsatz von PAV verbessern lassen.

Die folgenden Diagramme zeigen die Ein-/Ausgabezeiten (Software-Zeiten = Summe aus Hardware-Bedienzeiten und Software-Wartezeiten) bei verschiedenen RAID-Levels und bei einer für einen OLTP-Betrieb typischen Last:

- kurze Blöcke (2 KBytes)
- standardisierter Lasttyp „Random25“
- Read-Hit-Rate 80%

Bei den Messungen wurde folgende Konfiguration verwendet:

- Wie auf [Seite 59](#) beschrieben
- Gemessen wurde pro RAID-Gruppe auf einem Volume.
Pro Volume waren 4 Tasks aktiv.

Bei steigender Last (Ein-/Ausgaberate) werden die Ein-/Ausgabezeiten (Hardware + Software) bei verschiedenen RAID-Levels verglichen. Für jeden RAID-Level wurden Messungen mit unterschiedlich vielen Alias-Geräten durchgeführt.

Auf der DX8400 generierte RAID-Gruppen mit jeweils 16 Volumes:

- RAID1/0 (4+4 Platten)
- RAID6 (6+2 Platten)
- RAID5 (3+1 Platten)
- RAID1 (1+1 Platten)

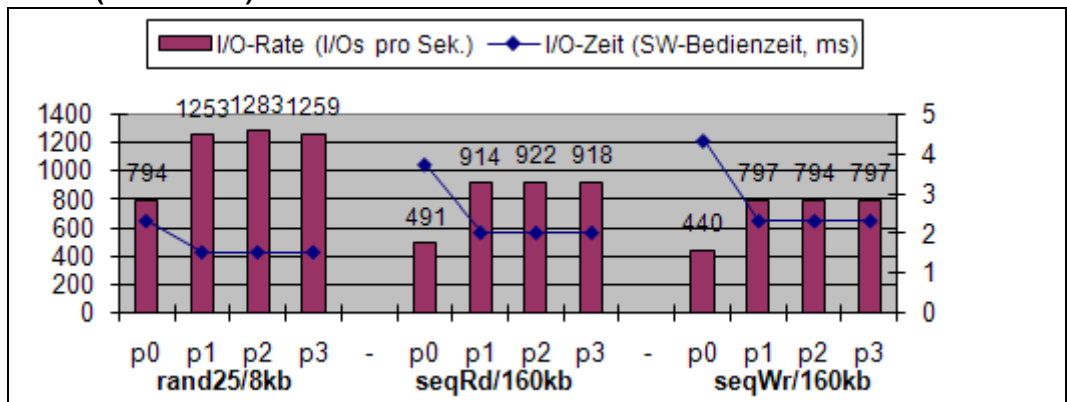
PAV-Varianten:

- p0: kein PAV
- p1-3: PAV mit 1-3 Alias-Geräten

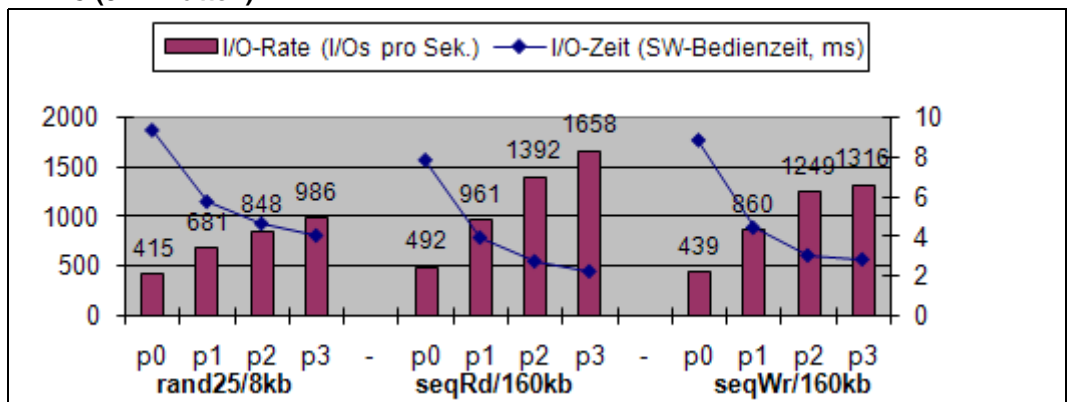
Untersuchte Lastfälle:

- rand25/8kb: Random25 mit 8 KB-Blöcken (Nachbildung eines TP-Betriebs)
- seqRd/160kb: Sequential read mit 160 KB-Blöcken (Durchsatz-orientierte Last)
- seqWr/160kb: Sequential write mit 160 KB-Blöcken (Durchsatz-orientierte Last)

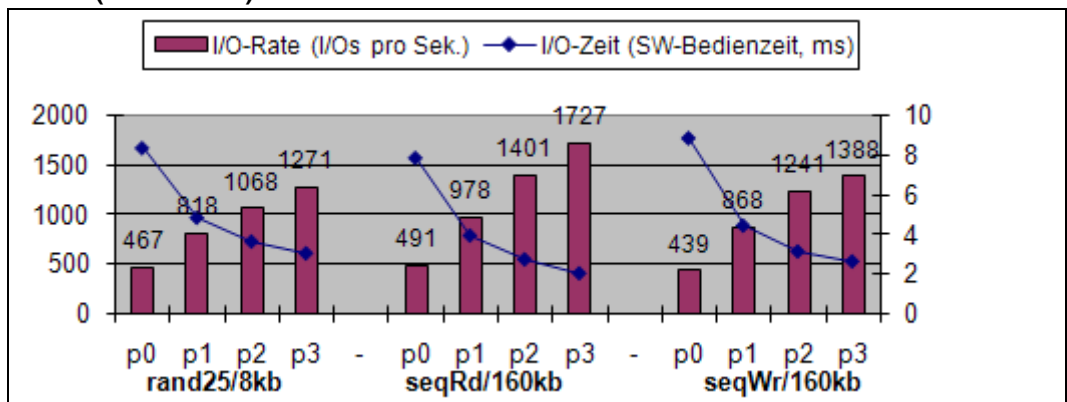
RAID1 (1+1 Platten)

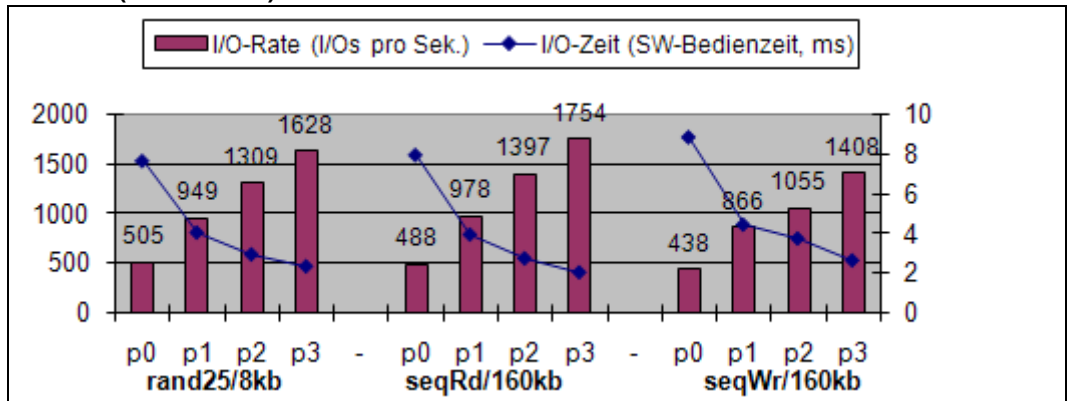


RAID5 (3+1 Platten)



RAID6 (6+2 Platten)



RAID1/0 (4+4 Platten)

RAID1/0, RAID6, RAID5:

Bei diesen RAID-Level mit Data Striping zeigen sich erhebliche Verbesserungen sowohl bei den I/O-Raten, als auch bei den I/O-Zeiten. Die Verbesserungen steigen bei entsprechend hoher Last mit der Anzahl der Alias-Geräte und gelten für Transaktions- und Durchsatz-orientierte Lastfälle.

Bei RAID1 (kein Data Striping) genügt ein Alias-Gerät. Weitere Alias-Geräte bringen keine weitere Verbesserung.

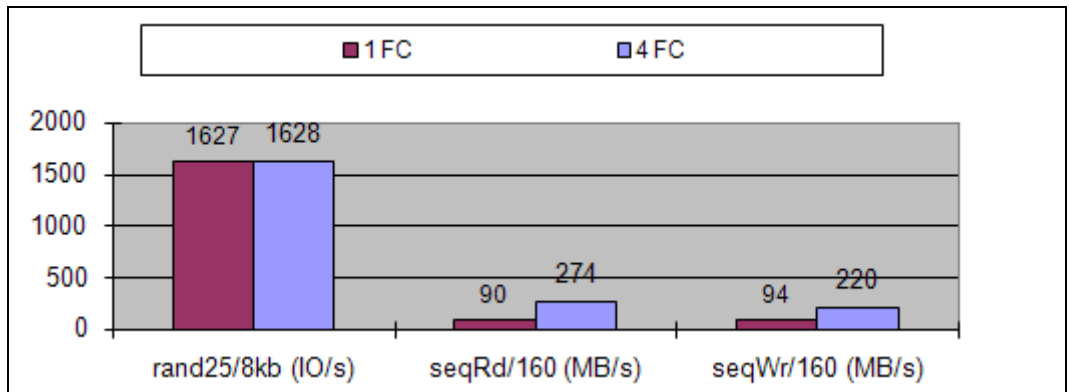
Ohne PAV liegen die Software-Bedienzeiten erheblich über den Hardware-Bedienzeiten. Die Ein-/Ausgaben warten im Betriebssystem. Mit steigender Anzahl der Alias-Geräte sinkt die Software-Bedienzeit und erreicht mit 4 Alias-Geräten (und 4 Tasks) die Hardware-Bedienzeiten. Die Hardware-Bedienzeiten erhöhen sich durch die erhöhte Parallelität auf dem Plattenspeichersystem nur sehr wenig.

4.2.1.5 Grenzen von PAV

Um die erwünschten Verbesserungen mit PAV zu erreichen, sind die in diesem Abschnitt beschriebenen Randbedingungen zu beachten.

Anschluss über mehrere Kanäle Typ FC

Der Einsatz von PAV erfordert eine ausreichend performante Anbindung des Plattenspeichersystems ETERNUS DX an BS2000/OSD. Das folgende Bild zeigt am Beispiel RAID1/0 mit PAV-Variante p3 die Leistungsminderung bei einem Anschluss mit nur einem Kanal gegenüber einem Anschluss mit vier Kanälen. Es werden die Lastfälle rand25/8kb (I/Os pro Sekunde), seqRd/160kb (MB/s) und seqWr/160kb betrachtet. Die Dateigröße ist mit 10 GB pro Task so groß, dass nahezu 100% Cache-Miss erreicht wird.



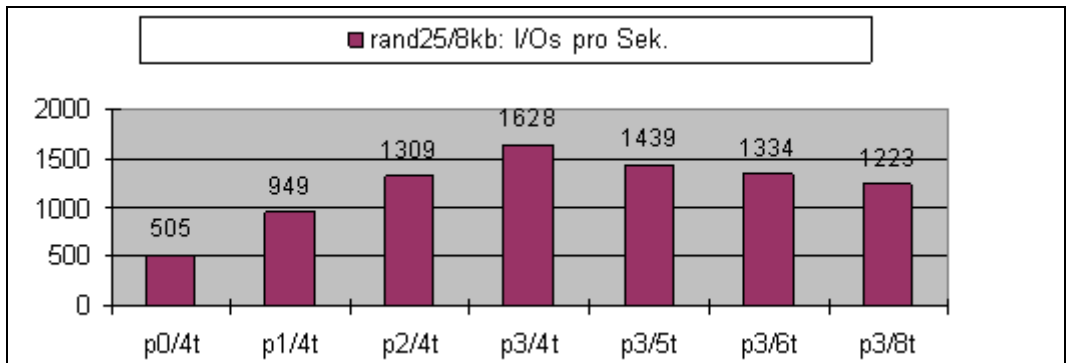
Bei den relativ kurzen Blöcken des Lastfalls rand25/8kb stellt ein Kanal Typ FC (1 Gbit/s) keinen Engpass dar.

Anders ist es bei den großen Blöcken. Hier kann die volle Leistung der PAV-Variante p3 nur mit mehreren FC erreicht werden. Im Beispiel sind mindestens 3 Anschluss-Pfade notwendig.

Viele parallele Ein-/Ausgaben

Die Messungen zu PAV erfolgten bisher mit 4 Tasks, die parallele Ein-/Ausgaben auf ein Volume ausführten. Dabei zeigten sich mit steigender Anzahl der Alias-Geräte deutliche Verbesserungen.

In diesem Abschnitt wird am Beispiel der Last rand25/8kb gezeigt, dass bei zu vielen parallelen Tasks (und damit Ein-/Ausgaben) die insgesamt erreichbare I/O-Rate wieder sinkt. Variiert wird das Verhältnis „Anzahl der Alias-Geräte / Anzahl der Tasks“. Gemessen wurde mit großen Dateien (10 GB pro Task, bei 5 GB und 6 bis 8 Tasks ergaben sich keine Unterschiede) und damit nahezu 100% Cache-Miss.



Bei zu vielen parallelen Tasks, die auf zu große Dateien zugreifen, sinkt die insgesamt erreichbare I/O-Rate. Ursache sind die zunehmend aufwändigeren Zugriffe auf die physikalischen Platten.

Ohne PAV würde dieser Abfall schon nach „p0/4t“ (kein PAV, 4 Tasks) mit 505 Ein-/Ausgaben beginnen.

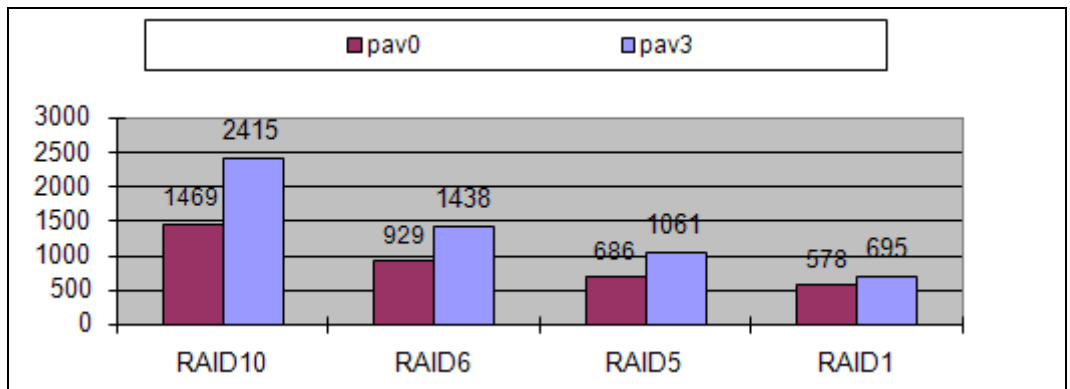
Leistung der RAID-Gruppe

Die Leistung eines Volumens kann mit PAV nur dann gesteigert werden, wenn die entsprechende RAID-Gruppe noch Reserven hat. Wenn die RAID-Gruppe, etwa durch Belastung durch andere Volumens, schon entsprechend ausgelastet ist, dann ist keine Verbesserung mehr möglich.

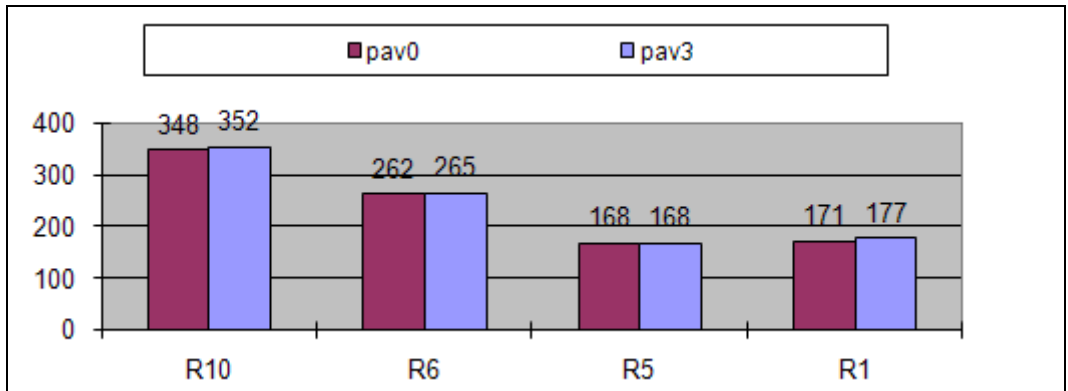
Die folgenden Grafiken zeigen, ob trotz sehr hoher Last auf alle Volumens mit PAV noch Verbesserungen möglich sind.

Es werden für alle 16 Volumens einer RAID-Gruppe jeweils 4 parallele Tasks gestartet. Jeder Task führt eine der Lasttypen rand25/8kb, seqRd/160kb bzw. seqWr/160kb aus. Verglichen werden die Ergebnisse ohne PAV (pav0) und mit PAV mit 3 Alias-Geräten (pav3). Zusätzlich zu den RAID-Gruppen wird auch das Verhalten bei „Cache-Hit“ angegeben.

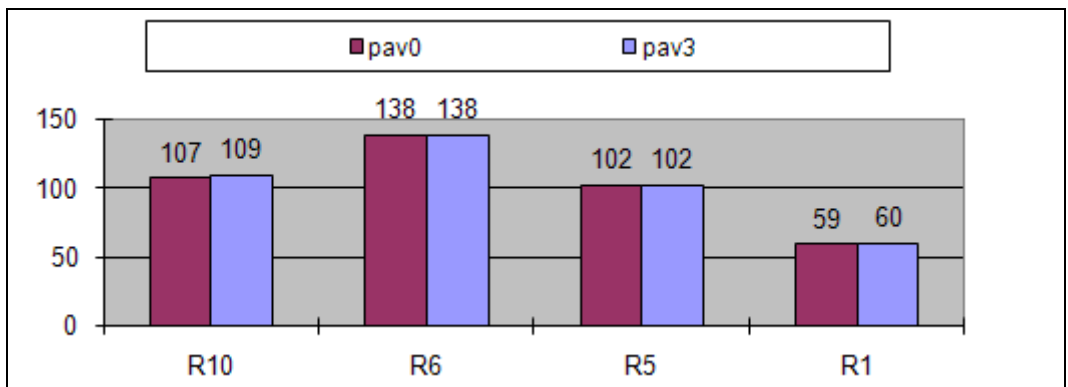
Last rand25/8kb (Ein-/Ausgaben)



Cache-Hit: pav0 = 23071, pav3 = 23030 IO/s.

Last seqRd/160kb (Ein-/Ausgaben)

Cache-Hit: pav0 = 349, pav3 = 354 IO/s.

Last seqWr/160kb (Ein-/Ausgaben)

Cache-Hit: pav0 = 365, pav3 = 351 IO/s.

Beim Transaktions-orientierten Lastfall rand25/8kb können noch deutliche Verbesserungen erreicht werden, obwohl die RAID-Gruppen auch ohne PAV sehr hoch ausgelastet sind.

Bei den Durchsatz-orientierten Lastfällen ist keine Steigerung mehr möglich, die Platten sind ausgelastet.

Auch bei Cache-Hit sind durch PAV keine Verbesserungen mehr möglich.

4.2.2 Symmetrix DMX und VMAX

Dieser Abschnitt enthält Messergebnisse und Folgerungen für die Performance der Plattenspeichersysteme Symmetrix DMX und VMAX bei Anschluss an S-Server über Kanäle Typ FC.



Der Durchsatz eines Plattenspeichersystems Symmetrix DMX-4 und VMAX ist vergleichbar mit dem Durchsatz von DMX-3. Auch die Hardware-Bedienszeiten bei Read-Hit sind vergleichbar. Die Hardware-Bedienszeiten bei Read-Miss sind bei DMX-4 und VMAX kürzer als bei DMX-3.

Bei den folgenden Messungen wurde folgende Konfiguration verwendet:

- S-Server mit 4 CPUs unter VM2000 mit einem Mono-Gastsystem (Privileg IO-PRIORITY) mit einer dedizierten CPU (dies entspricht den Verhältnissen eines native Monoprozessor-Betriebs mit ca. 320 RPF)
- Anschluss des S-Servers an das Plattenspeichersystem über 4 Kanäle Typ FC mit einer Leistung von je 1 GBit.
- Plattenspeichersysteme:
 - Symmetrix DMX-3 mit 32 GB Cache und physikalischen Platten mit je 146 GB und 15.000 rpm
 - Symmetrix VMAX mit 96 GB Cache und physikalischen Platten mit je 146 GB und 15.000 rpm
- 16 Volumes auf Platten des Typs D3435 mit Datenformat NK2 mit geringer gegenseitiger Beeinflussung (wenige logische Volumes auf einem physikalischen Gerät)

4.2.2.1 Durchsatz mit einer Task

Die folgenden beiden Tabellen zeigen den erzielten Durchsatz (die Ein-/Ausgaberate in Ein-/Ausgaben pro Sekunde (IO/s) und den Durchsatz in Mbyte/s), den eine Anwendertask bei Plattenzugriffen zum Plattenspeichersystem Symmetrix DMX-3 erreichen kann.

Der Durchsatz wurde für die standardisierten Lasttypen bei verschiedenen Blockgrößen und einer Read-Hit-Rate von 100% gemessen:

Eine Task: Ein-/Ausgaben/s	Blockgröße (KB)					
	2	4	8	16	32	160
„Random25“	3.400	3.200	2.500	2.200	1.550	430
„Sequential read“	3.700	3.500	3.000	2.300	1.600	450
„Sequential write“	3.000	2.800	2.500	2.100	1.500	400

Eine Task: Mbyte/s	Blockgröße (KB)					
	2	4	8	16	32	160
„Random25“	7,1	13	21	37	51	70
„Sequential read“	7,7	14	25	38	53	74
„Sequential write“	6,2	12	21	34	50	67

4.2.2.2 Durchsatz mit bis zu vier Kanälen Typ FC

Die folgenden beiden Tabellen zeigen den erzielten Durchsatz (die Ein-/Ausgaberate in Ein-/Ausgaben pro Sekunde (IO/s) und den Durchsatz in Mbyte/s) bei Plattenzugriffen zum Plattenspeichersystem Symmetrix VMAX bei bis zu vier Kanälen Typ FC.

Der Durchsatz wurde für den Lasttyp „Random25“ bei verschiedenen Blockgrößen und einer Read-Hit-Rate von 100% gemessen:

„Random25“: Ein-/Ausgaben/s	Blockgröße (KB)					
	2	4	8	16	32	160
1 Kanal Typ FC	8.800	8.800	8.200	5.700	3.400	700
2 Kanäle Typ FC	17.000	17.000	16.000	11.000	6.800	1.400
4 Kanäle Typ FC	34.000	33.000	29.000	21.000	12.000	2.600

„Random25“: Mbyte/s	Blockgröße (KB)					
	2	4	8	16	32	160
1 Kanal Typ FC	18	36	67	93	110	115
2 Kanäle Typ FC	34	69	130	180	220	225
4 Kanäle Typ FC	69	135	235	345	400	425

Kanäle Typ FC skalieren sehr gut. Das heißt, mit n Kanälen Typ FC kann annähernd der n-fache Durchsatz erreicht werden. Voraussetzung dafür ist eine ausreichende Leistung der BS2000-CPU's und der Peripherie.

4.2.2.3 Hardware-Bedienzeiten

Hardware-Bedienzeiten beim Lesen

Folgende Tabelle zeigt die Ein-/Ausgabezeiten (Hardware-Bedienzeiten) für Read-Hit und Read-Miss, wie sie in üblichen Konfigurationen und bei üblichem Lastaufkommen erreicht werden. Die Zeiten für den Read-Miss werden abhängig vom RAID-Level dargestellt.

Die Hardware-Bedienzeiten sind neben der Ein-/Ausgaberate auch abhängig vom Ausbau des Plattenspeichersystems. Bei kleinen Konfigurationen können die Zeiten etwas günstiger sein. Bei sehr großen Konfigurationen können die Zeiten schlechter sein.

Ein-/Ausgabezeit Lesen (ms)	Blockgröße (KB)					
	2	4	8	16	32	160
Read-Hit	0,24	0,28	0,36	0,52	0,84	3,4
Read-Miss RAID1	4,8	4,9	5,0	5,3	6,0	11
Read-Miss RAID1/0	4,8	4,9	5,0	5,3	6,0	11
Read-Miss RAID5	5,3	5,4	5,6	6,0	6,8	13

Hardware-Bedienzeiten bei OLTP-Betrieb

Folgende Tabellen zeigen die Ein-/Ausgabezeiten (Hardware-Bedienzeiten) bei unterschiedlichen RAID-Leveln und Read-Hit-Raten, wie sie für einen OLTP-Betrieb (Lasttyp „Random25“) bei üblichen Konfigurationen und üblichem Lastaufkommen erreicht werden. Bei den Schreibaufträgen wird von einem ausreichend großen Cache des Plattenspeichersystems ausgegangen, d.h. der Schreibauftrag ist nach der Übertragung in den Cache abgeschlossen. Der Transfer auf die physikalischen Platten erfolgt asynchron.

RAID1(1+1), RAID1/0(4+4)

Ein-/Ausgabezeit „Random25“ (ms)	Blockgröße (KB)					
	2	4	8	16	32	160
Read-Hit 100%	0,27	0,31	0,39	0,55	0,87	3,4
Read-Hit 80%	0,95	1,0	1,1	1,3	1,6	4,6
Read-Hit 60%	1,6	1,7	1,8	2,0	2,4	5,7

RAID5(3+1)

Ein-/Ausgabezeit „Random25“ (ms)	Blockgröße (KB)					
	2	4	8	16	32	160
Read-Hit 100%	0,27	0,31	0,39	0,55	0,87	3,4
Read-Hit 80%	1,0	1,1	1,2	1,4	1,8	4,9
Read-Hit 60%	1,8	1,8	2,0	2,2	2,6	6,4

Die Ein-/Ausgabezeiten gelten für „normale“ Last, d.h. einer Ein-/Ausgaberate, bei der das logische Volume nicht überlastet ist, wobei RAID1/0(4+4) die höchste und RAID1(1+1) die niedrigste Last verträgt.

4.2.2.4 PAV und RAID-Level

Für Hochleistungssysteme wird neben einem Anschluss über Kanal Typ FC für S-Server der Einsatz von PAV/DPAV (Dynamic Parallel Access Volume) zusammen mit einem geeigneten RAID-Level empfohlen. Dieser Abschnitt zeigt, dass sich die Ein-/Ausgabezeiten und der Durchsatz (Ein-/Ausgaberate) durch den Einsatz von PAV verbessern lassen.

Die folgenden Diagramme zeigen die Ein-/Ausgabezeiten (Software-Zeiten = Summe aus Hardware-Bedienzeiten und Software-Wartezeiten) bei verschiedenen RAID-Levels und bei einer für einen OLTP-Betrieb typischen Last:

- kurze Blöcke (2 KBytes)
- standardisierter Lasttyp „Random25“
- Read-Hit-Rate 80%

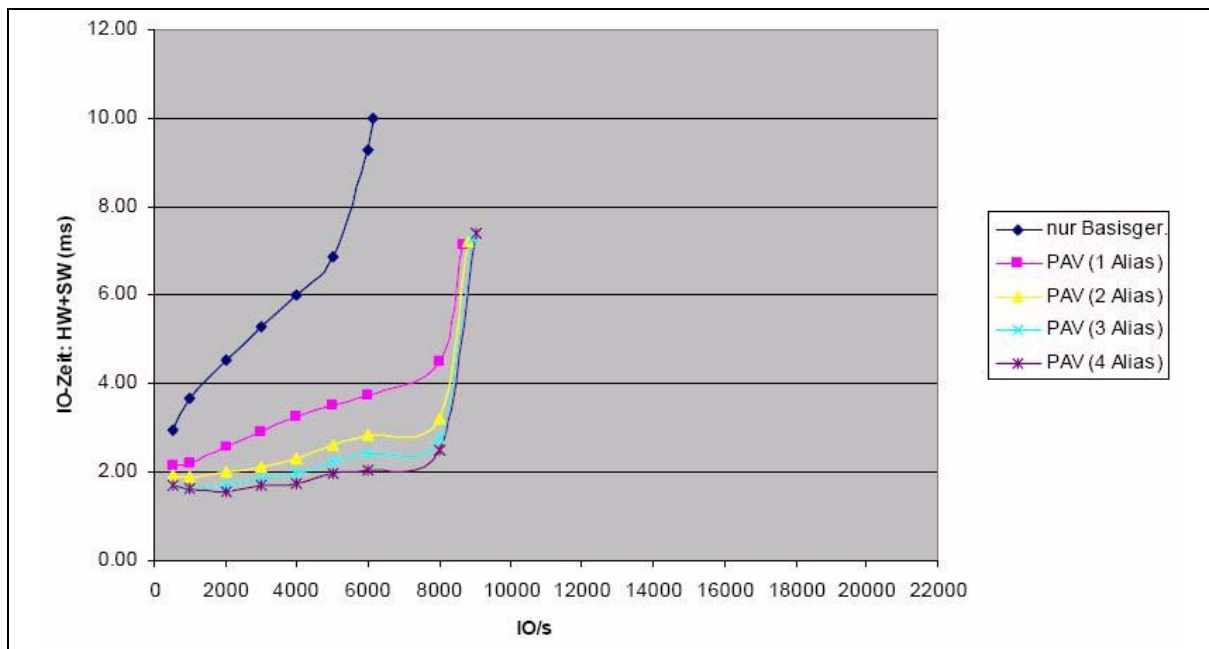
Bei den Messungen wurde folgende Konfiguration verwendet:

- Wie auf [Seite 70](#) beschrieben
- Jeweils 8 Volumes auf unabhängigen physikalischen Platten.
Pro Volume waren 8 Tasks aktiv.

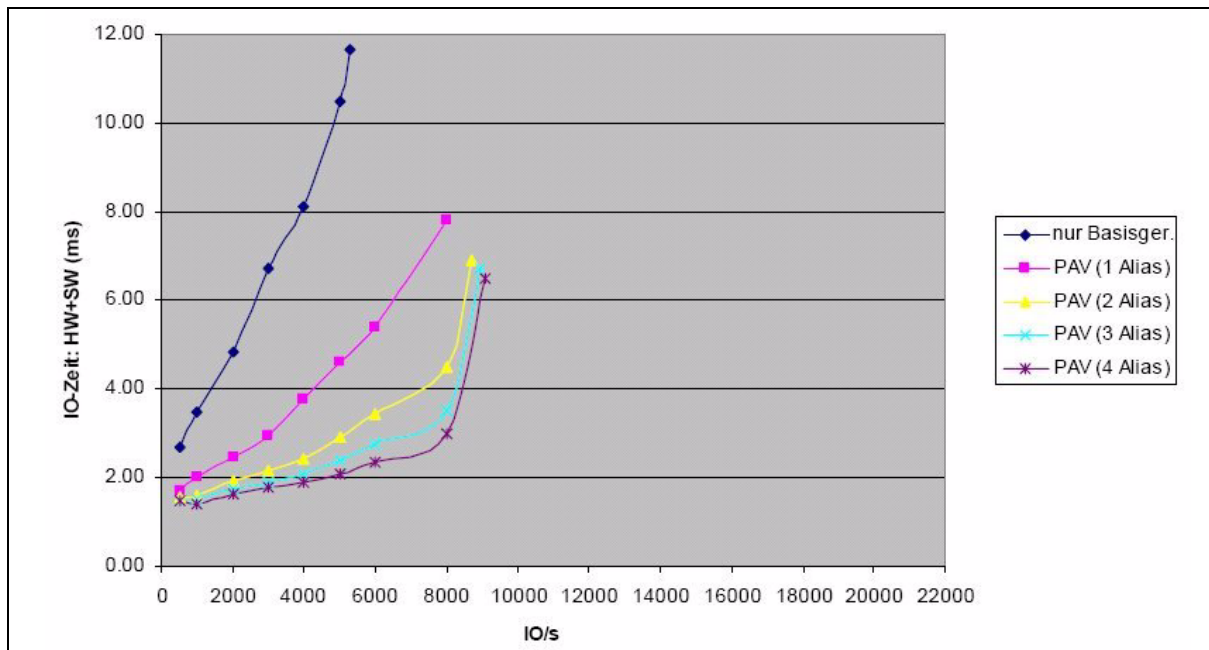
Bei steigender Last (Ein-/Ausgaberate) werden die Ein-/Ausgabezeiten (Hardware + Software) bei verschiedenen RAID-Levels verglichen. Für jeden RAID-Level wurden Messungen mit unterschiedlich vielen Alias-Geräten durchgeführt.

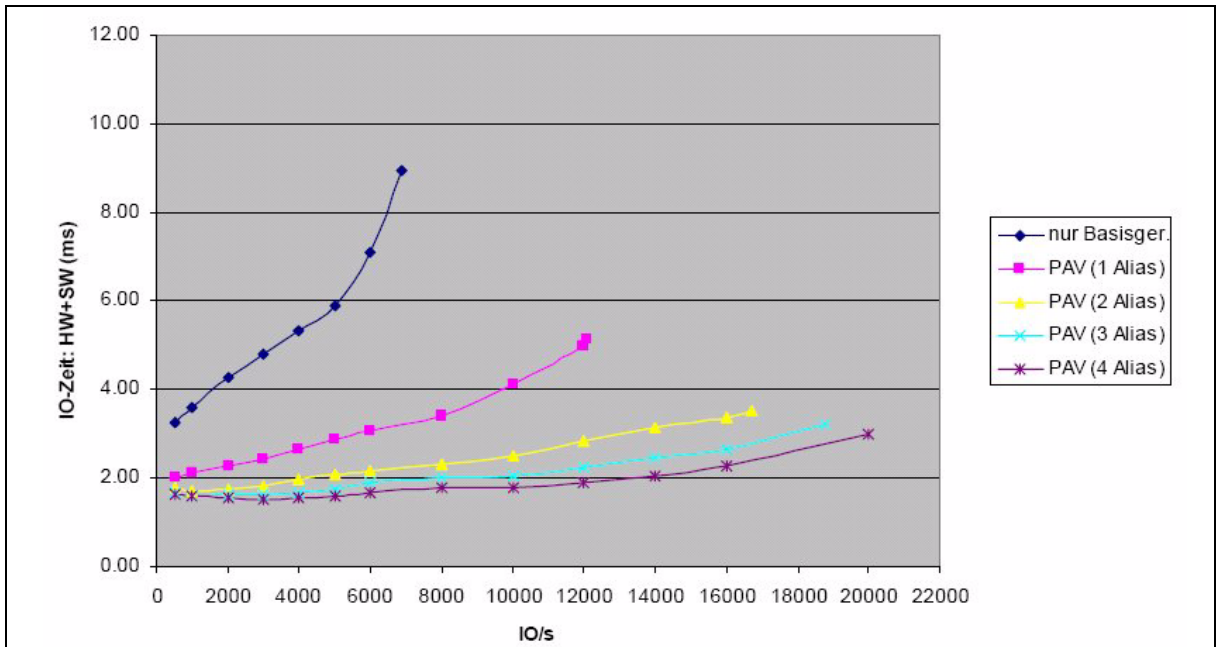
Die folgenden Messungen wurden mit dem Plattenspeichersystem Symmetrix DMX-3 durchgeführt. Die Ergebnisse zu PAV gelten analog auch für Symmetrix DMX-4 und VMAX.

RAID1 (1+1 Platten)



RAID5 (3+1 Platten)



RAID1/0 (4+4 Platten)

Mit PAV und bei RAID1/0 kann die Parallelität der Ein-/Ausgabelast effektiv genutzt werden.

Mit PAV ergeben sich erhebliche Verbesserungen. Schon mit einem Aliasgerät sinken die Ein-/Ausgabezeiten und steigen die Ein-/Ausgaberaten deutlich. Eine weitere Verbesserung ergibt sich mit 2 Aliasgeräten. Weitere Aliasgeräte bringen nur noch geringe Verbesserungen.

Beim Einsatz von DPAV sorgt das Dienstprogramm IORM für die Zuordnung einer adäquaten Anzahl von Aliasgeräten.

Bei gleicher Ein-/Ausgaberate ergeben sich mit PAV bei allen RAID-Levels deutlich bessere Ein-/Ausgabezeiten (Hardware- und Softwarezeiten).

Bei den RAID-Levels RAID1 und RAID5 werden mit PAV deutlich bessere Ein-/Ausgaberaten erreicht. Wegen des höheren Aufwandes beim Schreiben (Berechnen und Schreiben der Parity-Informationen) liegt die Ein-/Ausgaberate von RAID5 trotz des „data striping“ über mehrere Platten in der Größenordnung von RAID1. Die Ein-/Ausgabezeiten und Ein-/Ausgaberaten von RAID6 liegen in der Größenordnung von RAID5.

Eine wesentlich höhere Ein-/Ausgaberate erreicht RAID1/0. Hier spielt neben dem „data striping“ auch die größere Anzahl Platten eine Rolle.

Die für RAID1/0 in dieser Konfiguration mit akzeptablen Ein-/Ausgabezeiten erreichte Ein-/Ausgaberate liegt bei über 20.000 Ein-/Ausgaben pro Sekunde.

RAID6 (6+2 Platten)

Die Leistung von RAID6 (6+2 Platten) für S-Server mit Plattenspeichersystem DMX-3 wird im Vergleich zu RAID5 (3+1) und RAID1/0 (4+4) bewertet. Nicht betrachtet werden Kapazitäts- und Sicherheitsaspekte. Betrachtet wird das Verhalten mit PAV mit bis zu 4 Alias-Geräten.

Lesen

- Ohne PAV werden bei allen drei RAID-Leveln beim „Sequential read“ gute Durchsätze von etwa 70 Mbyte/s erreicht.
- Mit PAV wird bei mehreren Alias-Geräten ein ausgezeichneter Durchsatz auf einem logischen Volume erreicht. Er beträgt bei mehreren Tasks für alle drei RAID-Level 250 Mbyte/s.

Schreiben

- Ohne PAV ist beim „Sequential write“ der Durchsatz in allen RAID-Leveln etwa 60 Mbyte/s. Beim „Random write“ sinkt der Durchsatz bei RAID5 und RAID6 aber auf 20-30 Mbyte/s, also deutlich unter den RAID1/0-Wert (60 Mbyte/s).
- Auch beim Schreiben mit PAV und mehreren Alias-Geräten ist RAID1/0 performanter als RAID5 oder RAID6, bei denen die Parity-Informationen zusätzlich berechnet und geschrieben werden müssen. Mit RAID1/0 wird beim „Sequential write“ auf ein Volume ein hervorragender Durchsatz von mehr als 200 Mbyte/s erreicht. Bei RAID5 und RAID6 wird ein Durchsatz von etwa 100 Mbyte/s erreicht. Beim „Random write“ sind die Vorteile von RAID1/0 noch ausgeprägter (Faktor 3-5).
- Der Durchsatz von RAID5 und RAID6 liegt in der gleichen Größenordnung.

Bewertung:

- Mit PAV kann der Durchsatz auf S-Servern erheblich gesteigert werden, auch mit 2 oder 3 Alias-Geräten. (D)PAV wird uneingeschränkt empfohlen.
- TP-Betrieb:
Beim Schreiben in einem normalen TP-Betrieb wird davon ausgegangen, dass im Cache des Plattenspeichersystems genügend Platz ist, dass also kein „Delayed Fast Write“ nötig ist. Alle drei RAID-Level werden empfohlen.
- Datensicherung:
Hier wird von den Platten gelesen. Alle drei RAID-Level werden empfohlen.
- Datenrekonstruktion:
Hier wird auf die Platten geschrieben. Dabei ist RAID1/0 vorteilhaft. Die Begrenzungen auf der Band-Seite sind zu beachten (siehe [Seite 112](#)).

4.3 Plattenspeichersysteme an SQ-Servern

Die Ein-/Ausgabe auf FC-Peripherie erfolgt über I/O-Prozessoren und die PCI-Controller des SQ-Servers.

4.3.1 ETERNUS DX

Dieser Abschnitt enthält Messergebnisse und Folgerungen für die Performance des Plattenspeichersystems ETERNUS DX440 S2 bei FC-Anschluss an einen SQ-Server.

Bei den dargestellten Messungen wurde folgende Konfiguration verwendet:

- Server SQ210-80F im native Betrieb
- Anschluss des SQ-Servers an das Plattenspeichersystem über 4 FC-Anschlüsse mit einer Leistung von je 8 GBit. Die Platten innerhalb des Plattenspeichersystems waren über 2 Pfade erreichbar.
- Plattenspeichersystem ETERNUS DX440 S2 mit 96 GB Cache und physikalischen Platten mit je 450 GB und 15.000 rpm
- 16 Volumes auf Platten des Typs D3435 mit Datenformat NK2 mit geringer gegenseitiger Beeinflussung (wenige logische Volumes auf einem physikalischen Gerät)

Es empfohlen, die Plattenspeichersystem ETERNUS DX mit RAID1/0 zu betreiben. Die Messungen unterschiedlicher RAID-Varianten zeigen die besten Ergebnisse für RAID1/0.

4.3.1.1 Durchsatz mit einer Task

Folgende Tabellen zeigen den erzielten Durchsatz (die Ein-/Ausgaberate in Ein-/Ausgaben pro Sekunde (IO/s) und den Durchsatz in Mbyte/s), den ein Anwendertask bei Plattenzugriffen zum Plattenspeichersystem ETERNUS DX 440 S2 erreichen kann.

Der Durchsatz wurde für die standardisierten Lasttypen bei verschiedenen Blockgrößen und einer Read-Hit-Rate von 100% gemessen:

Eine Task: Ein-/Ausgaben/s	Blockgröße (KB)			
	2	16	32	160
„Random25“	2.670	2.015	1.730	685
„Sequential read“	2.790	2.070	1.735	650
„Sequential write“	2.365	1.880	1.765	820

Eine Task: Durchsatz [Mbyte/s]	Blockgröße (KB)			
	2	16	32	160
„Random25“	5	31	55	107
„Sequential read“	5	32	54	102
„Sequential write“	4	29	55	128

Der Durchsatz steigt mit zunehmender Blockgröße deutlich an. Es wird empfohlen, mit großen Blöcken zu arbeiten.

4.3.1.2 Durchsatz mit 32 Tasks

Folgende Tabellen zeigen den erzielten Durchsatz (die Ein-/Ausgaberate in Ein-/Ausgaben pro Sekunde (IO/s) und den Durchsatz in Mbyte/s), der bei Plattenzugriffen zum Plattenspeichersystem ETERNUS DX 440 S2 erreicht werden kann.

Der Durchsatz wurde für die standardisierten Lasttypen bei verschiedenen Blockgrößen und einer Read-Hit-Rate von 100% bei 32 parallel arbeitenden Tasks (je 2 Tasks bearbeiten ein Volume) gemessen:

32 Tasks: Ein-/Ausgaben/s	Blockgröße (KB)			
	2	16	32	160
„Random25“	48.250	39.640	32.490	12.210
„Sequential read“	49.160	40.130	32.500	11.935
„Sequential write“	47.425	38.855	31.630	11.430

32 Tasks: Durchsatz [Mbyte/s]	Blockgröße (KB)			
	2	16	32	160
„Random25“	94	620	1.015	1.905
„Sequential read“	95	625	1.015	1.865
„Sequential write“	90	605	990	1.785

Die CPUs der SQ210-80F waren bei den Messungen mit den kleinen Blöcken voll ausgelastet. Mit den kleinen Blöcken konnte der maximale Durchsatz des Plattenspeichersystems nicht erreicht werden.

Bedingt durch den großen Cache konnten im eingeschwungenen Zustand (nahezu) alle Lese-Ein-/Ausgaben aus dem Cache befriedigt werden. Auch die meisten Schreib-Ein-/Ausgaben wurden bereits nach der Ablage der Daten im Cache quittiert. Dies bewirkt sowohl beim Schreiben, als auch beim Lesen kürzere Ein-/Ausgabe-Zeiten und einen größeren Durchsatz.

4.3.1.3 Hardware-Bedienzeiten

Hardware-Bedienzeiten beim Lesen

Die Hardware-Bedienzeiten sind neben der Ein-/Ausgaberate auch abhängig vom Ausbau des Plattenspeichersystems (vor allem von der Cache-Größe).

Bei Plattenspeichersystemen mit großem Cache wird der größte Teil der Ein-/Ausgaben über den Cache bedient. Die Plattenzugriffe erfolgen asynchron im Plattenspeichersystem und haben kaum Einfluss auf die Hardware-Bedienzeiten.

Beim Arbeiten mit kleinen Blöcken (2 KB bis 16 KB) liegen die Hardware-Bedienzeiten bei bis zu 1 ms. Bei Blöcken mit 32 KB bis 160 KB liegen die Hardware-Bedienzeiten bei 2 - 8 ms.

Hardware-Bedienzeiten bei OLTP-Betrieb

Die Ein-/Ausgabezeiten bei einer typischen OLTP-Last („Random25“) sind bei hohen Read-Miss-Raten meist um ca. 10% schlechter als die Ein-/Ausgabezeiten beim Lesen (siehe vorigen Abschnitt).

4.3.2 Symmetrix DMX, VMAX

Dieser Abschnitt enthält Messergebnisse über und Folgerungen für die Performance des Plattenspeichersystems Symmetrix DMX-3 (Platten der Größe 146 GB mit 15.000 rpm) bei FC-Anschluss an SQ-Server.



Der Durchsatz eines Plattenspeichersystems Symmetrix DMX-4 oder VMAX ist vergleichbar mit dem Durchsatz von DMX-3. Auch die Hardware-Bedenzeiten bei Read-Hit sind vergleichbar. Die Hardware-Bedenzeiten bei Read-Miss sind bei DMX-4 und VMAX kürzer als bei DMX-3.

4.3.2.1 Durchsatz mit einer Task

Die folgenden beiden Tabellen zeigen den erzielten Durchsatz (die Ein-/Ausgaberate in Ein-/Ausgaben pro Sekunde (IO/s) und der Durchsatz in Mbyte/s), den eine Anwendertask bei Plattenzugriffen zum Plattenspeichersystem Symmetrix DMX-3 über FC erreichen kann.

Bei den Messungen wurde ein Plattenspeichersystem Symmetrix DMX-3 mit 32 GB Cache, angeschlossen an einen Server SQ200-80E über 4 FC-Anschlüsse mit einer Leitung von je 2 GBit, mit 16 Volumes auf Platten vom Typ D3435 mit Datenformat NK2 und mit geringer gegenseitiger Beeinflussung verwendet.

Der Durchsatz wurde für die standardisierten Lasttypen bei verschiedenen Blockgrößen und einer Read-Hit-Rate von 100% gemessen:

Eine Task: Ein-/Ausgaben/s	Blockgröße (KB)			
	2	16	32	160
„Random25“	2.240	1.730	1.520	540
„Sequential read“	2.520	1.640	1.540	550
„Sequential write“	1.750	1.810	1.480	520

Eine Task: Mbyte/s	Blockgröße (KB)			
	2	16	32	160
„Random25“	4	27	47	83
„Sequential read“	4	25	48	85
„Sequential write“	3	28	46	80

Der Durchsatz steigt mit zunehmender Blockgröße deutlich an. Es wird empfohlen, mit großen Blöcken zu arbeiten.

Das NK2-Datenformat liefert die besten Ergebnisse. Mit K-Datenformat liegen die Durchsätze niedriger: um etwa 5-10 % bei kurzen, bis etwa 20-30 % bei großen Blöcken.

4.3.2.2 Hardware-Bedienzeiten

Hardware-Bedienzeiten beim Lesen

Folgende Tabelle zeigt die Ein-/Ausgabezeiten (Hardware-Bedienzeiten) für Read-Hit und Read-Miss, wie sie in üblichen Konfigurationen und bei üblichem Lastaufkommen erreicht werden. Die Zeiten für den Read-Miss werden abhängig vom RAID-Level dargestellt.

Die Hardware-Bedienzeiten sind neben der Ein-/Ausgaberate auch abhängig vom Ausbau des Plattenspeichersystems. Bei den Messungen wurde ein Plattenspeichersystem Symmetrix DMX-3 mit 32 GB Cache, angeschlossen an einen Server SQ200-80E über 4 FC-Anschlüsse mit einer Leistung von je 2 GBit, mit 16 Volumes auf Platten vom Typ D3435 mit Datenformat NK2 und mit geringer gegenseitiger Beeinflussung verwendet.

Bei kleinen Konfigurationen können die Zeiten etwas günstiger sein. Bei sehr großen Konfigurationen können die Zeiten schlechter sein.

Ein-/Ausgabezeit Lesen (ms)	Blockgröße (KB)			
	2	16	32	160
Read-Hit	0,3	0,6	1,1	6,3
Read-Miss RAID1	4,1	4,3	4,5	9,2
Read-Miss RAID1/0	4,1	4,5	4,8	10,0
Read-Miss RAID5	4,0	4,4	4,6	10,5

Die Hardware-Bedienzeiten hängen stark von der Cache-Größe des Plattenspeichersystems ab. Mit zunehmender Cache-Größe verringert sich die Anzahl der Cache-Miss und die Hardware-Bedienzeiten werden zunehmend kürzer.

Hardware-Bedienzeiten bei OLTP-Betrieb

Folgende Tabelle zeigt die Ein-/Ausgabezeiten (Hardware-Bedienzeiten) für Read-Hit bei unterschiedlichen RAID-Leveln, wie sie für einen OLTP-Betrieb (Lasttyp „Random25“) bei üblichen Konfigurationen und üblichem Lastaufkommen erreicht werden.

Ein-/Ausgabezeit „Random25“ [ms]	Blockgröße (KB)			
	2	16	32	160
Read-Hit RAID1	0,3	0,6	1,2	7,8
Read-Hit RAID1/0	0,3	0,5	0,9	5,0
Read-Hit RAID5	0,3	0,5	0,9	5,0

Das NK2-Datenformat liefert die besten Ergebnisse.

Mit K-Datenformat liegen die Hardware-Bedienzeiten höher: um etwa 5-10 % bei kurzen, bis etwa 20-30 % bei langen Blöcken.

4.4 Net-Storage

BS2000/OSD ab V9.0 ermöglicht den Zugang zu UNIX-Dateisystemen über NFS. BS2000-Dateien können damit in freigegebenen Verzeichnissen von File-Servern, dem Net-Storage, abgelegt und bearbeitet werden.

Aus BS2000-Sicht werden auch auf dem Speichermedium Net-Storage Daten auf Volumes abgelegt, die allerdings einem Pubset fest zugeordnet sind. Sie dienen als Speichererweiterung des Pubsets. Damit ist es möglich, den maximal möglichen Speicherplatz eines SF-Pubsets (4 TByte) zu überschreiten.

Grundlegende Informationen zu Net-Storage finden Sie im Handbuch „Systembetreuung“ [10].

Net-Storage dient in erster Linie zur Ablage von (großen) Dateien, auf die nicht mit höchster Performance zugegriffen werden muss. Performance-kritische Daten (z.B. Datenbank-Dateien) sollten nach wie vor auf Pubsets abgelegt werden.

Dies hat mehrere Gründe:

- Die Leistungsfähigkeit moderner Netzwerke steigt zwar stetig, aber die Bandbreite muss mit anderen Teilnehmern im Netz geteilt werden. Deshalb können, im Gegensatz zu FC-Peripherie, keine exakten Voraussagen zu Durchsatz und Latenzzeiten getroffen werden.
- Lesender Zugriff auf Net-Storage ist performanter als schreibender Zugriff. Mit Lesen und Schreiben größerer I/O-Blöcke lässt sich ein wesentlich höherer Durchsatz erzielen als mit kleinen I/O-Blöcken.
- Der Durchsatz wächst mit der Anzahl der parallelen Aufträge und mit der Blockgröße, bis ein Engpass erreicht wird. So können bei einer sequenziellen Verarbeitung von großen Blöcken (HSMS/ARCHIVE, COPY, File Transfer) sehr gute Durchsatzwerte erzielt werden, die bis an die Grenzen der vorhandenen Bandbreite im Netzwerk reichen.
- Wenn die Konfiguration bzw. die Hardware des Netzwerks es erlaubt, sollte mit Jumbo-Frames gearbeitet werden (MTU 9.000 Bytes). Die MTU muss auch am Net-Client sowie an allen beteiligten Netz-Komponenten (z.B. Switches) konfiguriert bzw. unterstützt werden.

Details zur Konfiguration von Net-Storage sind im „Net-Storage Installation Guide“ zusammengefasst. Den „Net-Storage Installation Guide“ finden Sie im Internet unter <http://www.fujitsu.com> bei der Produktbeschreibung von BS2000/OSD-BC V9.0.

4.5 LAN-Anschluss

Dieser Abschnitt analysiert die Kommunikationsleistung der BS2000/OSD-Server.

Folgende typische Anwendungsarten werden betrachtet:

- Transaktions-orientierter Betrieb:
Senden und Empfangen von kurzen Nachrichten (10 Bytes).
Messwert: Transaktionen pro Sekunde (TA/s).
- Durchsatz-orientierter Betrieb (Massendatentransfer ähnlich File-Transfer):
Senden bzw. Empfangen von Nachrichten mit einer Länge von 8, 16, 32, 64 Kbyte.
Messwert: Megabyte pro Sekunde (Mbyte/s).

Die Ergebnisse gelten bei optimalen Bedingungen für eine Socket-Anwendung (TCP/IP), wobei die Nachrichten nicht weiter bearbeitet werden. Die Übertragung der Nachrichten erfolgt von Hauptspeicher zu Hauptspeicher, es finden also keine Plattenzugriffe statt. Bei realen Anwendungen sind deren Anforderungen an CPU und Platten sowie eventuell nicht optimale Netzwerke zu berücksichtigen.

4.5.1 LAN-Anschluss für S-Server

S-Server werden über einen Kanaladapter HNC (High-speed Net Connect) an ein LAN (Local Area Network) angeschlossen.

HNC

Der High Speed Net Connect (kurz: HNC) ist ein hoch performanter Netzanschluss für BS2000-Systeme auf S-Servern, siehe Handbuch „HNC“ [13].

BS2000/OSD V9.0 bedient folgende HNC-Modelle. Die Auslieferung erfolgt immer mit einem Linux-Betriebssystem.

- HNC-IV 91853 (kurz: HNC-IV)
Basiert auf einem Primergy-Server TX300-S3.
Die Anbindung über Kanal Typ S wird mit diesem HNC letztmalig unterstützt.
- HNC-V 91854 (kurz: HNC-V)
Basiert auf einem Primergy-Server TX300-S6.
Ab HNC-V wird nur noch die Anbindung über Kanal Typ FC unterstützt.
Darüber hinaus werden ab HNC-V folgende neue Funktionen angeboten:
 - Checksum-Offload (siehe Seite 90)
 - Link-Aggregation (siehe Seite 90)
 - Unterstützung von Net-Storage (siehe Handbuch „Systembetreuung“ [10])
- HNC-VI 91855 (kurz: HNC-VI)
Basiert auf einem Primergy-Server TX300-S7.
Hardware- und Software-Update ohne neue Funktionen.

Der HNC unterstützt den Anschluss an einen S-Server über Kanal Typ S (HNC-IV) sowie über Kanal Typ FC. Er bedient Fast Ethernet und Gigabit Ethernet LANs. Die hohe Durchsatzleistung des Gigabit LAN erfordert die Anbindung des HNC an den Server über Kanal Typ FC.

Die folgenden Ausführungen beziehen sich ausschließlich auf den Anschluss über Kanal Typ FC. Der im Folgenden nicht mehr behandelte Anschluss des HNC-IV über Kanal Typ S begrenzt den Durchsatz auf etwa 12 Mbyte/s, während beim FC-Anschluss (1 Gbit/s) und Gigabit Ethernet ein bis zu 7-facher Durchsatz erreicht werden kann.

Die Transaktionsraten und Durchsatzwerte in den folgenden Abschnitten wurden mit dem HNC-V gemessen. Verglichen werden sie mit den Messwerten des Vorgängermodells HNC-IV.

Es werden nur Gigabit Ethernet LANs (1.000 Mbit/s) betrachtet. Dabei wird unterschieden zwischen den Standardeinstellungen mit einer Maximum Transmission Unit (MTU) von 1.500 Bytes und Jumbo-Frames mit einer MTU von 9.000 Bytes.



Bei Jumbo-Frames müssen Netz, Switches und Partner-Systeme ebenfalls für Jumbo-Frames konfiguriert sein.

HNC an Kanal Typ FC

Folgende Tabelle zeigt die Messergebnisse für einen S-Server mit 4 CPUs und einer Leistung von 1.060 RPF, der mit einem HNC an ein Gigabit-LAN mit einer MTU von 1.500 bzw. 9.000 Bytes angeschlossen ist.

HNC-IV und HNC-V Gigabit Ethernet (1.000 Mbit/s)				
Kanal-Anschluss	1 Kanal Typ FC			
MTU [Bytes]	1.500		9.000	
Transaktionen [TA/s]	88.000		84.000	
CPU [%]	96		96	
	Senden	Empfangen	Senden	Empfangen
Durchsatz [Mbyte/s]	60	75	65	78
CPU [%]	21	26	14	14

Die maximale Transaktionsrate wird mit dem abwechselnden Senden und Empfangen kurzer Nachrichten (10 Bytes) erreicht. Der Server, auf dem die Messungen stattfanden, war damit allein durch die Kommunikation fast ausgelastet.

Der maximale Durchsatz wird mit dem einseitigen Senden bzw. Empfangen langer Nachrichten (ab 8 KB) erzielt.

Bei einer MTU von 9.000 Bytes ist die Transaktionsrate ca. 5% geringer, der Durchsatz ca. 8% höher als bei einer MTU von 1.500 Bytes.

Vergleich HNC-V mit HNC-IV

Die Werte für den maximal erreichbaren Durchsatz sowie die maximal erreichbaren I/O-Raten sind auf HNC-V und HNC-IV identisch.

Neue Funktionen mit HNC-V und openNet Server ab V3.4 (siehe Handbuch „BCAM“ [2])

● Checksum-Offload

Mit dieser Funktion wird die Checksum-Berechnung von BS2000/OSD auf den (wenig ausgelasteten) HNC verlagert. Damit wird in BS2000/OSD beim Arbeiten mit langen Nachrichten der CPU-Bedarf für die Netzwerklast reduziert. Der maximale Durchsatz bleibt unverändert.

Diese Funktion wird ab HNC-V und openNet Server ab V3.4 standardmäßig benutzt.

Bei langen Nachrichten kann damit der CPU-Bedarf von BS2000/OSD deutlich verringert werden:

- mit Standard-Frames (MTU 1.500): um 10-15 %
- mit Jumbo-Frames (MTU 9.000): ca. 15 % beim Senden und ca. 35 % beim Empfangen

● Link-Aggregation

Mit dieser Funktion können mehrere virtuelle Kanaladapter (VCAs) über eine oder mehrere physikalische FC-Verbindungen gebündelt und als ein einziges Gerät betrieben werden. Ziel ist es, die Netzleistung zu erhöhen (siehe Handbuch „BCAM“ [2]).

Es ist zu beachten, dass beim Einsatz dieser Funktion vor allem beim Senden von Nachrichten zusätzliche CPU-Leistung des BS2000-Servers verbraucht wird. Je höher der Bündelungsgrad, desto stärker steigt der CPU-Verbrauch für die Netzlast.

In einer Konfiguration, mit der die VCAs von zwei Kanälen Typ FC gebündelt wurden, konnte das Gigabit Ethernet voll ausgelastet werden. Ohne diese Funktion lag die Auslastung des Ethernets bei 50 - 60%.

Bei Übergang von einer Bündelung von zwei Kanälen Typ FC mit je zwei VCAs zu einer Bündelung von zwei Kanälen Typ FC mit je vier VCAs steigt der CPU-Bedarf in BS2000/OSD um ca. 30%. Daher wird empfohlen, nur die nötigen VCAs zu bündeln.

Durchsatz mit einer Verbindung

Mit einer (TCP-)Verbindung sind zu einem leistungsstarken Partnersystem folgende Werte möglich:

LAN	Durchsatz (Mbyte/s)
Kanal-Anschluss	Typ FC
Gigabit Ethernet (Ethernet Frames)	60 - 75
Gigabit Ethernet (Jumbo Frames)	65 - 80

Diese Werte werden bei weniger leistungsstarken Partnersystemen nicht erreicht. Sie können ebenfalls nicht erreicht werden, wenn andere Komponenten, wie Platten oder ein nicht optimales Netz, den Datentransfer bremsen.

Die besten Antwortzeiten zwischen einem BS2000-System und einem über Gigabit Ethernet verbundenen Partnersystem liegen für kurze Nachrichten bei 0,5 Millisekunden.

Formel für Serverleistung

Als Faustformel für die Serverleistung, die alleine durch die Kommunikation ab der Sockets-Schnittstelle (ohne Verarbeitung der Nachrichten) benötigt wird, gilt bei Fast Ethernet und Gigabit Ethernet LAN:

- MTU 1.500 Bytes
 - 60 - 90 Transaktionen pro RPF
 - 0,10 - 0,20 Mbyte/s pro RPF
- MTU 9.000 Bytes
 - 60 - 80 Transaktionen pro RPF
 - 0,20 - 0,30 Mbyte/s pro RPF

4.5.2 LAN-Anschluss für SQ-Server

SQ-Server werden über einen integrierten Controller (PCI-Controller) an ein LAN angeschlossen. Die Ein-/Ausgaben zum integrierten Controller werden, wie alle Ein-/Ausgaben, über den I/O-Prozessor geleitet (siehe [Abschnitt „I/O-Prozessoren \(X2000\)“ auf Seite 36](#)).

Folgende Tabelle zeigt die Messergebnisse für ein Gigabit Ethernet mit einer MTU von 1.500 bzw. 9.000 Bytes. Die angegebenen Maximalwerte werden mit mehreren parallelen (TCP-)Verbindungen erreicht. Dargestellt werden die Werte für die beiden SQ-Server SQ200-80E und SQ210-160F, jeweils mit openNet Server V3.5.

SQ200-80E	Gigabit Ethernet (1.000 Mbit/s)			
	1.500		9.000	
MTU [Bytes]	1.500		9.000	
Transaktionen [TA/s]	33.000		33.000	
CPU [%]	100		100	
IOP [%]	5		5	
Durchsatz [Mbyte/s]	Senden 16 KB	Empfangen	Senden 16 KB	Empfangen
	120	120	115	115
	CPU [%]	16	30	21
	IOP [%]	3	4	5

SQ210-160F	Gigabit Ethernet (1.000 Mbit/s)			
	1.500		9.000	
MTU [Bytes]	1.500		9.000	
Transaktionen [TA/s]	88.500		87.500	
CPU [%]	98		98	
IOP [%]	17		17	
Durchsatz [Mbyte/s]	Senden	Empfangen	Senden	Empfangen
	116	116	120	121
	CPU [%]	10	13	7
	IOP [%]	14	13	13

Die Randbedingungen zur Bestimmung der maximale Transaktionsrate sowie des maximalen Durchsatzes sind im Abschnitt [„HNC an Kanal Typ FC“ auf Seite 89](#) beschrieben.

Die maximalen Transaktionsraten liegen deutlich über allen heute üblichen Anforderungen, z.B. von OLTP-Anwendungen. Die Transaktionsrate wird in der Regel durch die Vollaustastung der BS2000-CPU's begrenzt.

Um Hochleistungsnetze ausnutzen zu können, ist eine ausreichende CPU-Kapazität erforderlich. Soweit nicht die Grenzen des LANs erreicht werden, erhöht sich mit der Anzahl CPU's auch die erreichbare Leistung.

Der Datendurchsatz beim Senden von Nachrichten ist abhängig von der verwendeten Nachrichtenlänge. Grundsätzlich gilt: je länger eine Nachricht, desto höher der Durchsatz. Mit nur einer Verbindung ist folgender Durchsatz möglich (gemessen wurden Nachrichtenlängen von 8 KB bis 64 KB):

LAN	Durchsatz (Mbyte/s)			
	SQ200-80E		SQ210-160F	
	Senden	Empfangen	Senden	Empfangen
Gigabit Ethernet / Ethernet Frames	62 - 100	115	49 - 100	115
Gigabit Ethernet / Jumbo Frames	62 - 100	119	71 - 100	120

Diese Werte können nur bei leistungsstarken Partnersystemen und einer optimalen Netzkonfiguration erreicht werden.

Schon mit einer Task kann beim Empfangen von Nachrichten der maximale Netz-Durchsatz von 120 Mbyte/s erzielt werden.

Homogene Kopplung

Bei der Kopplung zweier virtueller Maschinen (mit je 4 CPUs) auf einem Server SQ200-80E über LOCLAN mit einer Standard-MTU von 65.512 Bytes werden folgende Werte erreicht:

- Die maximale Transaktionsrate beträgt ca. 12.000 TA/s.
Die BS2000-CPU's der Gastsysteme sind dann zu 91 - 98% ausgelastet.
Die minimale Antwortzeit liegt bei ca. 0,7 ms.
- Der maximale Durchsatz liegt sowohl beim Senden als auch beim Empfangen von 64 KB langen Nachrichten bei über 500 Mbyte/s. Dabei sind die BS2000-CPU's zu ca. 80% ausgelastet.

4.5.3 Hilfsmittel zur Leistungskontrolle

BCMON

Das BCOMON-Kommando startet eine zyklische BCAM-Überwachung und gibt die gewünschten Werte in regelmäßigen Abständen aus.

Die Ausgabe der Werte erfolgt über den Bedienplatz und zusätzlich in die Protokolldatei (`$(SYSAUDIT.SYS.CONSL0G.<date>.<counter>)`), so dass die Werte später analysiert werden können. Siehe auch Handbuch „BCAM“ [2].

Einfaches Beispiel für BCOMON

Dieses Beispiel zeigt die Messung des Durchsatzes einer spezifischen Verbindung. Ohne Angabe von `LINE=` ist auch die Messung des Gesamt-Durchsatzes möglich.

`/BCMON MODE=ON,RECORD=L2,LINE=LINEFCGB,SEC=30` (Kommando am Bedienplatz)

Alle 30 Sekunden werden die in dieser Zeit aufgelaufenen Messwerte ausgegeben, z.B.:

```
<C %BCAM-000... % BCA0B15 Line LINEFCGB:
BYTES=(I=929.088.154/O=930.140.116); I/O'S=(I=50.259/O=33.216);
PACKETING=(I=4,60/O=6,98) _____ (1)
<C %BCAM-000... % BCA0B16 Line LINEFCGB: UNICASTs=(I=231.372/O=231.981);
MULTICASTs=(I=0/O=0) _____ (2)
<C %BCAM-000... % BCA0B17 Line LINEFCGB: I/O ERRORs=(I=0/O=0);
WAIT+RETRIES=0; DISCARDS=(I=0/O=0); PROTOCOL ERRORs=(I=0/O=0);
L2 PROTOCOLs=(I=0/O=0) _____ (3)
```

- (1) Line LINEFCGB : Name der Leitung
 BYTES=(I= : Anzahl der empfangenen Bytes
 BYTES=(O= : Anzahl der gesendeten Bytes
 I/O'S=(I= : Anzahl der Eingaben
 I/O'S=(O= : Anzahl der Ausgaben
 PACKETING=(I= : Anzahl der Nachrichten pro Eingabe
 PACKETING=(O= : Anzahl der Nachrichten pro Ausgabe
- (2) Line LINEFCGB : Name der Leitung
 UNICASTs=(I= : Anzahl der empfangenen UNICASTs
 UNICASTs=(O= : Anzahl der gesendeten UNICASTs
 MULTICASTs=(I= : Anzahl der empfangenen MULTICASTs
 MULTICASTs=(O= : Anzahl der gesendeten MULTICASTs
- (3) Die Meldung BCA0B17 ist lediglich ein Fehlerzähler

NETSTAT

Das Programm NETSTAT ermöglicht die Abfrage von Informationen über Anwendungen, Verbindungen, Routing-Daten und Netzanschlüsse. Die Funktionalität des Programms NETSTAT steht auch via BCAM-Kommando SHOW-NET-STATISTICS/NETSTAT zur Verfügung. Das Programm NETSTAT kann mit dem SHOW-NET-STATISTICS/NETSTAT Kommando unter jeder beliebigen Benutzererkennung ohne Privilegierung genutzt werden. Siehe auch Handbuch „BCAM“ [2].

Einfaches Beispiel für NETSTAT

Dieses Beispiel zeigt die Messung des Gesamt-Durchsatzes mit NETSTAT.

`/netstat interface-sum-rate=*yes,waittime=30` (Benutzer-Kommando)

Nach den Lademeldungen und einer Anzeige der derzeit aufgelaufenen Werte werden alle 30 Sekunden die normierten Werte für die Eingaben und Ausgaben (jeweils in Paketen und Bytes) ausgegeben:

```
INTERFACE sum rate                               Date: Fri Nov 23 10:20:35 2012
PacketsIn      BytesIn      ErrorsIn  PacketsOut      BytesOut  ErrorsOut
58255627 1979306135058           0    70849955  516799149494           0
```

```
INTERFACE sum rate                               Date: Fri Nov 23 10:21:05 2012
dt(s)   PktsIn/s   KByteIn/s   ErrIn  PktsOut/s   KByteOut/s   ErrOut
30.007   363.35     24.35       0      265.17     552.28       0
30.005   345.44     22.70       0      229.20     476.90       0
30.005   319.58     20.66       0      194.33     404.37       0
```

openSM2

Folgende SM2-Messprogramme liefern wichtige Informationen zum Networking (siehe Handbuch „openSM2“ [18]):

- **BCAM-CONNECTION**
Messdaten für Verbindungen
- **CHANNEL-IO**
Messdaten zur Kanalauslastung und Kanalübertragungsraten.
Die Auslastung des Kanals Typ FC bei Anschluss eines HNC-IV 91853 kann mit SM2-Mitteln nicht gemessen werden.
- **PERIODIC TASK**
Messdaten über Tasks: Auslastung der BCAM-Tasks (z.B. BCAM und BCA0)
- **RESPONSETIME**
Messdaten über Antwortzeiten, Denkzeiten, Transaktionszeiten und Nachrichtenwartezeiten im BCAM-Pool
- **SAMPLING-DEVICE**
Messdaten über Ein-/Ausgaben, Datenmenge und Auslastung von Geräten
- **TCP-IP**
Messdaten zu TCP/IP-Verbindungen

5 Datensicherung

Die Leistungsfähigkeit der Online- und Nearline-Peripherie ist in den letzten Jahren kontinuierlich gestiegen. In diesem Zusammenhang haben sich Verbindungen vom Server zur Online- und Nearline-Peripherie in Fibre-Channel-Technologie zum Standard für offene Server-Systeme entwickelt.

An S-Servern (Kanal Typ FC) und an SQ-Servern ist Fibre Channel die Standard-Anschlusstechnik für Online- und Nearline-Peripherie.

Für eine optimale Performance bei der Datensicherung ist ein ausgewogener und kombinierter Einsatz von Online- und Nearline-Peripherie erforderlich. Die Konfiguration solcher Systeme muss am Sicherungsumfang und an den verfügbaren Zeitfenstern für die Datensicherung ausgerichtet werden. Dieser Abschnitt stellt dafür grundlegende Informationen bereit.

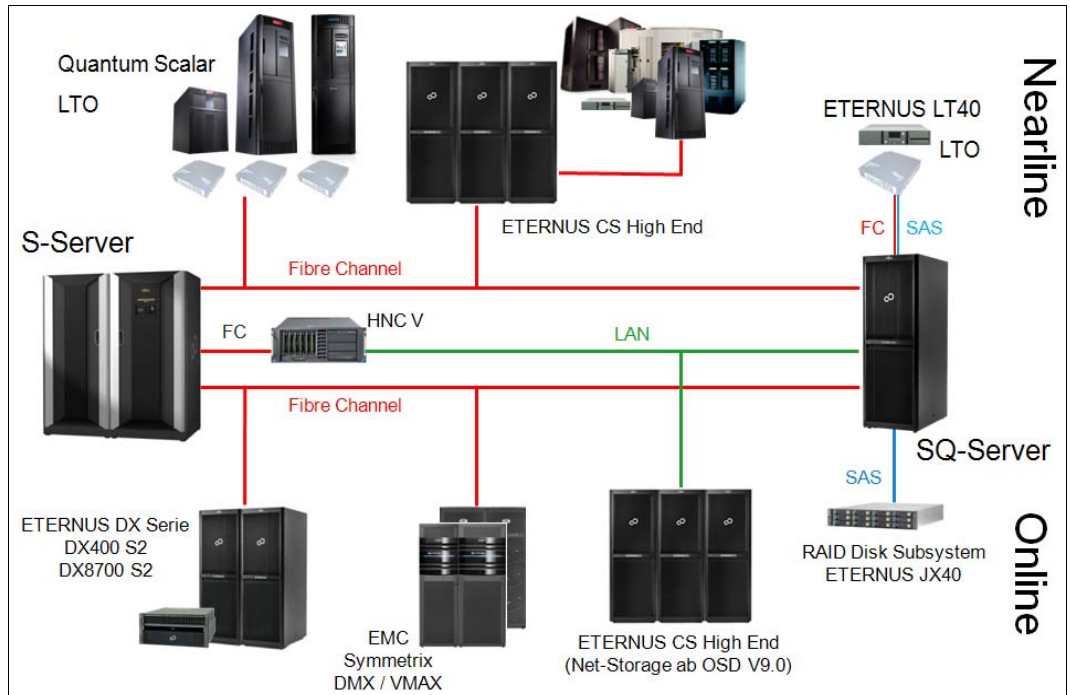


Bild 4: Aktuelle Online- und Nearline-Peripherie der BS2000/OSD-Server

Formen der Datensicherung in BS2000/OSD

Es gibt zwei Formen der Datensicherung:

- Bei der **logischen** Datensicherung werden Dateien und Jobvariablen von einem oder mehreren Datenträgern gelesen und zusammenhängend in eine Sicherungsdatei auf einem oder mehreren Datenträgern (z.B. Magnetbandkassetten) geschrieben. Dateien und Jobvariablen werden einzeln gesichert und können einzeln restauriert werden.
Für die logische Datensicherung stehen in BS2000/OSD die Softwareprodukte HSMS und ARCHIVE zur Verfügung (siehe die Handbücher „HSMS“ [14] und „ARCHIVE“ [1]).
- Bei der **physikalischen** Datensicherung werden nicht einzelne Dateien, sondern ganze Datenträger gesichert (einzelne Platten oder ganze Pubsets). Dabei werden sämtliche Daten eines Datenträgers, einschließlich der Datenkennsätze, blockweise in der physikalischen Reihenfolge auf einen zweiten Datenträger (z.B. Magnetbandkassette) geschrieben.
Für die physikalische Datensicherung steht in BS2000/OSD das Softwareprodukt FDDRL zur Verfügung (siehe Handbuch „FDDRL“ [11]).

Sicherungsvolumen

Das Sicherungsvolumen bei der **logischen** Datensicherung hängt vom Anwendungsprofil im Data Center ab. Beispielhaft für die unterschiedlichen Sicherungsarten und die gesicherten Datenmengen stehen folgende Werte:

- Vollsicherung: kompletter aktiver Datenbestand
- Differenzsicherung:
 - täglich z.B. 5% des kompletten Datenbestandes
 - wöchentlich z.B. 15% des kompletten DatenbestandesDabei wird in einer Differenzsicherung nicht die Differenz zur letzten Vollsicherung gebildet, sondern für jede Datei die Differenz zu allen vorherigen Sicherungen

Der Bedarf an Sicherungsmedien hängt von der Anzahl und vom Volumen der Voll- und Differenzsicherungen ab und muss für jeden Anwendungsfall separat ermittelt werden.

Bei der **physikalischen** Datensicherung eines Volumens überträgt FDDRL (Funktion DUMP) standardmäßig nur die Bereiche, die im F5-Kennsatz als belegt gekennzeichnet sind. Damit werden i.d.R. bei der physikalischen Sicherung mit FDDRL nicht mehr Daten übertragen als bei der logischen Sicherung mit HSMS. Bei physikalischen Sicherungen können aber weder Differenzsicherungen erstellt werden, noch können einzelne Dateien rekonstruiert werden.

5.1 Logische Datensicherung mit HSMS/ARCHIVE

Mit HSMS werden BS2000-Dateien, POSIX-Dateien und Jobvariablen des lokalen BS2000-Systems gesichert. Mit ARCHIVE können nur BS2000-Dateien und Jobvariablen gesichert werden.

Neben der Datensicherung ist auch die Datenrekonstruktion zu betrachten. Bei der (gegenüber der Sicherung seltenen) Wiederherstellung der gesicherten Daten gelten etwas andere Durchsatzwerte als bei der Sicherung. Diese sind jedoch i.d.R. erst dann kritisch, wenn eine vollständige Sicherung eingespielt werden soll. Bei der Rekonstruktion einzelner Dateien, die parallel zum normalen Betrieb läuft, spielt der Durchsatz eine untergeordnete Rolle.

Parallelität

Mit HSMS können mehrere Aufträge parallel ausgeführt werden. Der Grad dieser Parallelität kann durch die Anzahl der HSMS-Servertasks vorgegeben werden.

In einem Auftrag wiederum kann mit parallelen Datenströmen (auf mehrere Bandgeräte) gearbeitet werden. Dies wird mit dem Operanden `PARALLEL-RUNS` in der Aktionsanweisung oder als Archiv-Attribut gesteuert. Unter ARCHIVE wird dies mit dem Operanden `DRIVES` in der Anweisung `SAVE` gesteuert.

Mit HSMS/ARCHIVE werden außerdem beim Lesen von der Platte (Save) bzw. beim Schreiben auf die Platte (Restore) asynchrone Ein-/Ausgaben erzeugt. Diese Ein-/Ausgaben werden parallel ausgeführt, wenn die Funktion Parallel Access Volume (PAV) des Plattenspeichersystems zusammen mit einem geeignetem RAID-Level genutzt wird (siehe [Seite 55](#)).

Parallele Datenströme und parallele Ein-/Ausgaben führen beim Sichern zu kürzeren Sicherungszeiten.

HSMS/ARCHIVE-Directory

Das HSMS- bzw. ARCHIVE-Directory ist eine ISAM-Datei, in der alle Informationen über die gesicherten Datenbestände und die zugehörigen Datenträger hinterlegt sind.

Bei großer Anzahl zu sichernder Dateien und vielen Sicherungen hat das Directory einen entsprechend großen Umfang. Beim Sichern werden die gesicherten Dateistände in das Directory geschrieben, beim Restaurieren daraus gelesen. Zusätzlich muss bei Differenzsicherung noch gelesen werden, ob der aktuelle Dateistand schon im Sicherungsbestand enthalten ist. Die Zugriffe auf das Directory erfolgen in der ARCHIVE-Maintask parallel zum eigentlichen Sichern der Dateien in den Subtasks.

Sicherungsaufträge, die sich auf dasselbe Directory bzw. dasselbe HSMS-Archiv beziehen, können nicht parallel ablaufen.

Dateigrößen

Bei der logischen Datensicherung werden Dateien betrachtet. Für jede Datei entsteht ein gewisser Grundaufwand. Einen wesentlichen Einfluss auf den Durchsatz haben daher bei der logischen Datensicherung mit HSMS/ARCHIVE die Dateigrößen. Der Durchsatz mit vielen kleinen Dateien kann deutlich geringer ausfallen als im optimalen Fall mit wenigen sehr großen Dateien.

Große Bandblöcke

Die Verwendung von großen Bandblöcken steigert die Performance und die Bandausnutzung. Standardmäßig werden beim Sichern mit HSMS und ARCHIVE auf Magnetbandkassetten Bandblöcke von 256 KB geschrieben. Dazu ist in der ARCHIVE-Parameterdatei für den Parameter `BLOCK-SIZE-T-C` der Wert `BIG` voreingestellt. In den HSMS-Anweisungen und Archiv-Definitionen muss dann ggf. `BLOCKING-FACTOR=*STD / *MAX` angegeben werden.



Auf LTO-Laufwerke wird immer mit einer Blockgröße von 256 KB geschrieben, unabhängig von Parameter-Voreinstellungen und Archiv-Definitionen.

5.2 Performance-Maßnahmen im Sicherungsbetrieb (HSMS)

Dargestellt werden verschiedene Möglichkeiten, die vom Anwender eingesetzt werden können um eine bessere Performance bei der logischen Datensicherung mit HSMS/ARCHIVE zu erzielen. Hauptziel ist die Verkürzung der Sicherungszeit.

Die vorgestellten Maßnahmen können einzeln und unabhängig voneinander durchgeführt werden. Die Maßnahmen wirken sich in allen Situationen nur vorteilhaft aus. Eine Situationsanalyse ist nicht nötig.

Eine genaue Angabe des Performance-Gewinns jeder Maßnahme ist nicht möglich. Rahmensituation und Nutzungsart (z.B. Dateigröße) sind zu berücksichtigen. Alle Maßnahmen (mit Ausnahme von 4.) können mit einem regulären oder zeitkritischen Sicherungsauftrag einfach getestet werden. Der Performance-Gewinn ergibt sich aus dem Vergleich der Sicherungszeiten mit und ohne Maßnahme.

Die Maßnahmen werden vorrangig betrachtet in ihrer Auswirkung auf den Sicherungsbetrieb. Die Störung von parallel betriebenen Anwendungen kann i.d.R. vernachlässigt werden. Bei einigen Maßnahmen (z.B. 3. und 5.) wird die Sicherungszeit verkürzt bei unveränderter Gesamtzahl der Ein-/Ausgaben der Sicherung. Dadurch entsteht eine erhöhte Ein-/Ausgabe-Last pro Zeiteinheit.

1. Sicherungsumfang beschränken durch Dateiselektion und Differenzverfahren

Die Verwendung der Parameter `FROM-/EXCEPT-FILE` (Positiv- und Negativangabe mit Teilqualifizierung) wird empfohlen.

Eine Maximum-Backup-Class kann vorgegeben werden: nur Dateien mit einer Backup-Klasse kleiner oder gleich der angegebenen Backup-Klasse werden gesichert (als Dateiattribut mit Voreinstellung Backup-Class = A)

Temporäre Dateien und Paging-Dateien werden nie mitgesichert.

Management-Klassen in SM-Pubsets (frei vergebbares Dateiattribut für Gruppenbildung in SM-Pubsets) können zur Dateiselektion bei Backup und Migration benutzt werden

Es steht zur Wahl, ob auch migrierte Dateien mitgesichert werden sollen (Backup-Operand `SAVE-DATA`). Standardmäßig werden nur die Katalogeinträge von migrierten Dateien mitgesichert.

Bei Differenzsicherungen zwischen den periodischen Vollsicherungen kann zwischen zwei Verfahren gewählt werden:

- **MODIFIED:** Dateien nur sichern, wenn diese im Archiv noch nicht vorhanden sind
- **BY-CATALOG-MODIFICATION:** Dateien sichern, falls diese erst nach einem vorgegebenem Datum bzw. nach dem Datum des letzten Backup für das gleiche Archiv geändert wurden

2. Archive bzw. Archiv-Verzeichnisse sinnvoll aufteilen
 - Dezentrale Organisation (Archive pro Pubset) auch für große SF-Pubsets
3. Sicherungszeit verkürzen durch Parallelbetrieb (erhöhte Ein/Ausgabe-Last!)

Mehrere HSMS-Aufträge (für verschiedene Archive) gleichzeitig ausführen. Die Maximalzahl paralleler Aufträge kann durch die HSMS-Anweisung
`//MODIFY-HSMS-PARAMETERS NUMBER-OF-SUBTASKS=` eingestellt werden.

Es können pro Auftrag (Angabe von `PARALLEL-RUNS` entweder im Auftrag oder als Archiv-Attribut) mehrere Bandgeräte durch `ARCHIVE-Subtasking` genutzt werden.

Bei Kassetten mit hoher Kapazität sollte das `Several-SVID-Format` zum Fortschreiben (`CONTINUE`) benutzt werden. Dies ist praktisch eine Voraussetzung für einen Parallelbetrieb wegen des sonst hohen Bandverbrauchs.



Durch Erhöhung des Parallelbetriebes wird die Laufzeit verkürzt, die Anzahl der Ein-/Ausgaben pro Sekunde wird erhöht.

Bei Einsatz von vielen Bandgeräten (z.B. mit `ETERNUS CS HE`) können diese vermehrt parallel genutzt werden. `HSMS/ARCHIVE` sorgt für eine effiziente Nutzung dieser Geräte.

4. Durchsatz von Platte und Band erhöhen

Sowohl für Platten- als auch für Bandspeichersysteme ist wegen der höheren Datenraten der Anschluss über Kanal Typ `FC` dem Anschluss über Kanal Typ `S` vorzuziehen.

Platten am Kanal Typ `FC` sind immer im `FBA-Format`. Sie können mit einer Blockgröße von bis zu 480 KB betrieben werden (das `CKD-Format` erlaubt maximal 72 KB).

Um die Zahl der Band-Ein-/Ausgaben zu reduzieren, wird der Betrieb mit großen Bandblöcken (256 KB statt bisher 32 KB) empfohlen (Angabe `BLOCKING-FACTOR = *MAX` in Anweisungen, als Archivattribut oder als globaler `ARCHIVE-Parameter` `MBK-SIZE = *BIG`)

In `ARCHIVE` wird stets die Hardware-Komprimierung von/in `MBK-Geräten` genutzt. Wahlweise kann die Software-Komprimierung für `DMS-Sicherung` auf Platte genutzt werden. Die Komprimierung als Software-Funktion ist allerdings nachteilig für die `CPU-Performance`.



Zusätzlich sollte auf `K-Formatierung` verzichtet und das `NK2-Format` verwendet werden da die maximalen I/O-Längen bei `FBA-Platten` im `K-Format` 160 KB und im `NK2-Format` 480 KB betragen.

Der Parallelbetrieb wird in der Praxis durch die Kanalausstattung begrenzt, weniger durch die `CPU-Leistung`.

`HSMS/ARCHIVE` belastet die Plattenseite zusätzlich durch Katalogzugriffe und den Verschnitt von kleinen Dateien.

5. Multiplexing (Erzeugen paralleler Datenströme)

Im BS2000-Sicherungsbetrieb können mit HSMS/ARCHIVE und der Funktion PAV (Parallel Access Volume) mehrere Ein-/Ausgaben auf Platte parallel gestartet werden. Auf diese Weise kann ein Multiplexing der Datenströme zwischen Band und Platte erreicht werden.

6. Ausfallzeit für den Online-Betrieb verkürzen (Wiederanlauf der Anwendung vor dem Ende der eigentlichen Sicherung!)

Mit HSMS und seiner Funktion CCOPY (Concurrent Copy) können gleichzeitig konsistente Sicherungen erzeugt und die zu sichernden Daten in beliebiger Weise bearbeitet werden. Die Anwendung, die die zu sichernden Dateien bearbeitet, muss dazu vor Start der Sicherung kurzzeitig beendet oder zu einem Konsistenzpunkt geführt werden. Der Sicherungsauftrag kann dann gestartet werden. Nach einer Initialisierungsphase, die mit einer Jobvariablen überwacht werden kann, kann die Anwendung wieder aktiviert werden. Die Sicherung läuft dann parallel zur Dateibearbeitung.

Die Sicherung mit CCOPY verwendet entweder eine temporäre Arbeitsdatei oder sie nutzt eine Replikationsfunktion des Plattenspeichersystems. Nähere Informationen dazu finden Sie im Handbuch „HSMS“ [14].

Die Nutzung einer Replikationsfunktion des Plattenspeichersystems erlaubt die Lasttrennung zwischen der Anwendung auf der Originalplatte und der Sicherung auf der Spiegelplatte, so dass der Anwendungsbetrieb durch die Ein-/Ausgaben der Sicherung deutlich weniger gestört wird.



Eine Auftrags-Jobvariable zeigt in diesen Fällen den Status für den Anwendungs-Restart vor Abschluss des Auftrags an.

7. Performance der Katalogzugriffe bei DMS-Sicherungen



Bei Shared-Pubset-Betrieb werden normale Sicherungsaufträge HSMS-intern zur Master-Seite geschickt, um damit die schnelleren lokalen Katalogzugriffe ohne MSCF-Verkehr zu nutzen.

Sicherungsaufträge mit CCOPY von Spiegelplatten der Plattenspeichersysteme werden auch im Shared-Pubset-Betrieb stets lokal ausgeführt, weil die Katalogzugriffe lokal auf den abgetrennten Spiegelplatten der Plattenspeichersysteme erfolgen. Dies erlaubt eine Trennung der Ein-/Ausgabe-Last (z.B. mit zwei Systemen für Anwendung und für Sicherung)

Beim Sichern sehr vieler Dateien sollten, um einen kurzen Nachlauf zu erzielen, Summary-Reports (nur bei Fehlern erfolgen dateiweise Einträge) statt Full-Reports (Einträge für alle Dateien) verwendet werden.

8. Datenbank-Sicherungen im laufenden Datenbank-Betrieb

Die Sicherung von Dateien, die gesichert werden können, aber zum Schreiben eröffnet sind, ist mit der Backup-Option `SAVE-ONLINE-FILES = *YES` möglich, z.B. für UDS-Datenbanken mit Logfiles und für Oracle Datenbank-Dateien oder einzelne Tablespace.

Die Sicherung für SESAM-Datenbanken durch SESAM/SQL über die HSMS-Programmschnittstelle erfolgt in gleicher Weise, auch in Kombination mit CCOPY von Spiegelplatten der Plattenspeichersysteme. Dies führt zu minimalen Ruhephasen im Datenbank-Betrieb.

5.3 Physikalische Datensicherung mit FDDRL

Mit FDDRL werden Volumes auf Basis der physikalischen Blockstruktur und demnach ohne Katalogzugriff gesichert. FDDRL unterstützt Public Volumes (SF- und SM-Pubsets), Privatplatten und – unter besonderen Bedingungen – Volumes von Fremdsystemen.

Zur Sicherung von Pubsets gibt es mehrere Möglichkeiten:

- Sicherung eines exportierten Pubsets
- Sicherung eines abgetrennten Plattenspiegels (bei Einsatz von Replikationsfunktionen, siehe [Seite 50](#))
- Sicherung des Home-Pubsets (kann mit FDDRL ab V18.0 auch online gesichert werden)

Parallelität

Mit FDDRL können mehrere Volumes parallel und unabhängig voneinander unter eigenen Tasks (FDDRL-Subtasks) auf eine entsprechende Anzahl (ggf. virtueller) Bänder gesichert bzw. davon rekonstruiert werden. Der Grad der Parallelisierung wird durch den FDDRL-Parameter `TASK-LIMIT` vorgegeben.

Mit FDDRL werden außerdem beim Lesen von der Platte (DUMP) bzw. beim Schreiben auf die Platte (RELOAD) asynchrone Ein-/Ausgaben erzeugt. Diese Ein-/Ausgaben werden parallel ausgeführt, wenn die Funktion Parallel Access Volume (PAV, siehe [Seite 55](#)) des Plattenspeichersystems zusammen mit einem geeignetem RAID-Level genutzt wird.

RUN-PRIORITY

Die Task-Priorität für die FDDRL-Subtasks wird mit dem FDDRL-Parameter `RUN-PRIORITY` im Rahmen der Job-Klasse angegeben. Damit kann die Priorität der FDDRL-Verarbeitung gesteuert und der Durchsatz beeinflusst werden:

- Ziel: ein FDDRL-Lauf soll während eines gering belasteten BS2000-Betriebs möglichst rasch und ohne Rücksicht auf das Antwortzeitverhalten durchgeführt werden. In diesem Fall sollte der Parameter `TASK-LIMIT` den Wert der verfügbaren Bandgeräte haben (maximal 16), um durch den hohen Grad der Parallelisierung einen guten Durchsatz zu erreichen. Der Parameter `RUN-PRIORITY` sollte „hoch“ eingestellt werden.
- Ziel: ein FDDRL-Lauf soll während eines normalen BS2000-Betriebs ohne Beeinträchtigung der Performance der Hauptanwendung durchgeführt werden. In diesem Fall sollte der Parameter `TASK-LIMIT` mit Rücksicht auf die übrigen Anwendungen relativ niedrig eingestellt werden. Bei mehreren parallelen Läufen muss mit einer erkennbaren Verschlechterung des Antwortzeitverhaltens der übrigen Anwendungen rechnen. Der Parameter `RUN-PRIORITY` sollte „niedrig“ eingestellt werden.

Bandformat

Um der kontinuierlichen Entwicklung neuer Bandspeichersysteme Rechnung zu tragen, arbeitet FDDRL standardmäßig mit maximal großen Platten-Ein-/Ausgaben und „großen Bandblöcken“ in der gleichen Größe (für FBA z.B. 160 KB).

Bei sehr schnellen Bandspeichersystemen kann FDDRL Blöcke von bis zu 4 Platten parallel lesen und abwechselnd auf das Band schreiben. Dieses Verfahren, „Multiplexing“ genannt, ist besonders dann sinnvoll, wenn die Plattenverarbeitung langsamer ist als die Bandverarbeitung. Abhängig von den verfügbaren Bandspeichersystemen und der Größe der benutzten Band-Volumes können folgende Sicherungsstrategien eingesetzt werden:

- Viele Bandlaufwerke und kleine Band-Volumes

Mit ETERNUS CS HE sind in der Regel viele Bandlaufwerke verfügbar. Die Volume-Größe ist konfigurierbar und wird normalerweise wesentlich kleiner eingestellt als bei Einsatz von LTO-Kassetten. Hier empfiehlt sich zur Sicherung von Pubsets die FDDRL-Anweisung `//DUMP-PUBSET . . . , SAVE-ENTITY=*SINGLE-DISK`.

Pro Platten-Volume wird ein Band-Volume beschrieben

Damit wird eine maximale Performance erreicht. Um einen guten Band-Füllgrad zu erzielen sollten Band- und Plattengröße aufeinander abgestimmt sein.

- Wenige Bandlaufwerke und große Band-Volumes

Bei der Verwendung von z.B. LTO-Laufwerken wird folgende FDDRL-Anweisung empfohlen: `//DUMP-PUBSET . . . , SAVE-ENTITY=*DISK-SET(NUMBER-OF-DISK-SETS=n)`

Mehrere Platten-Volumes können im Multiplexing auf wenige Band-Volumes und Bandlaufwerke gesichert werden. Die Anzahl der parallel benutzten Laufwerke kann mit dem Parameter `NUMBER-OF-DISK-SETS` eingestellt werden. Das `TASK-LIMIT` muss mit `//MODIFY-FDDRL-PARAMETERS` darauf abgestimmt werden. Da mehrere Platten-Volumes auf ein Band-Volume gesichert werden, können große Band-Volumes gut ausgenutzt werden.

Die Erstellung von Disk-Sets mit `//DUMP-PUBSET` hat außerdem den Vorteil, dass die komplette und konsistente Restaurierung des Pubsets mit `//RELOAD-PUBSET` möglich ist. Ohne die Verwendung von Disk-Sets muss der Pubset mit `//RELOAD-DISK` restauriert werden. In diesem Fall ist es Aufgabe des Systemadministrators, die Vollständigkeit und Konsistenz der Restaurierung der einzelnen Volumes zu gewährleisten.



Neben den oben genannten Operandenwerten, sollte `TAPE-FORMAT=*STD` eingestellt sein (`//MODIFY-FDDRL-PARAMETERS`). So ermittelt FDDRL selbst das optimale Bandformat. Eine andere Formatangabe ist nur zum Datenaustausch mit Systemen mit früheren FDDRL-Versionen vorgesehen.

5.4 Komponenten bei der Datensicherung

Performance-relevante Komponenten bei der lokalen Datensicherung sind:

- Software: HSMS, ARCHIVE bzw. FDDRL
- BS2000-CPU
- Bandspeichersystem einschließlich Anschluss an den Server
- Plattenspeichersystem einschließlich Anschluss an den Server

In diesem Abschnitt wird die Leistung einiger Teilkomponenten betrachtet. Daraus lassen sich dann näherungsweise die zu erwartenden Durchsatzraten bei der Datensicherung bzw. -rekonstruktion in konkreten Konfigurationen abschätzen.

Die Sicherungsleistung wird vom kleinsten Durchsatzwert der beteiligten Teilkomponenten bestimmt.

Die angegebenen Durchsatzraten beziehen sich auf eine Task während der eigentlichen Sicherungsphase, also ohne Montieren des Bandes, ohne Erstellen der Report-Dateien oder dem Starten und Beenden der Programme.



Die Messergebnisse in diesem Abschnitt setzen günstige Bedingungen voraus, z.B. keine parallele Last auf der Messkonfiguration, ausreichend CPU-Leistung.

5.4.1 BS2000-CPU

Bei einer lokalen Datensicherung sind für den CPU-Bedarf in BS2000/OSD näherungsweise folgende Werte anzusetzen:

- Datensicherung mit HSMS, abhängig von der Dateigröße:
 - auf S-Servern: 1,3 (große/mittlere Dateien) - 2,3 (kleine Dateien) RPF pro Mbyte/s
 - auf SQ-Servern: 0,7 - 1,5 RPF pro Mbyte/s
- Datensicherung mit FDDRL: unter 0,2 RPF pro Mbyte/s



Insbesondere auf Multiprozessor-Systemen ist zu beachten, dass die für eine Sicherungstask notwendige CPU-Leistung von nur einer BS2000-CPU erbracht werden muss.

Die I/O-Prozessoren an S- und SQ-Servern haben i.A. keinen leistungsbeschränkenden Einfluss.

5.4.2 Anschluss an den Server

Für den Anschluss an den Server gelten die theoretischen Leistungen:

- Kanal Typ FC (S-Server): 1 Gbit/s
- FC-Anschluss (SQ-Server): 1 bis 8 Gbit/s

Plattenspeichersysteme sind normalerweise über mehrere Pfade angeschlossen. Wenn im Plattenspeichersystem die Möglichkeiten der Parallelisierung (geeignete RAID-Gruppen, PAV, Multiplexing) genutzt werden, dann können bei den Durchsatz-orientierten Lasten der Datensicherung die Anschlüsse sehr hoch ausgelastet werden.

Schnelle Bandspeichersysteme sollten an schnelle Kanäle Typ FC bzw. FC-Anschluss angeschlossen werden. Diese dürfen nicht gemeinsam mit den Plattenspeichersystemen genutzt werden.

Die Kanäle der S-Server zu den Bandspeichersystemen können bei der Datensicherung bereits durch wenige Geräte weitgehend ausgelastet werden.

5.4.3 Bandspeichersysteme

Virtuelles Archivsystem ETERNUS CS HE

Mit ETERNUS CS HE wird einem realen Archivsystem mit den physikalischen Laufwerken und Kassetten ein virtuelles Archivsystem vorgelagert. Während die Datensicherungsanwendung auf dem Server weiterhin davon ausgeht, mit MBK-Laufwerken zu arbeiten, nimmt das virtuelle Archivsystem eine Zwischenspeicherung auf Magnetplatte (Tape Volume Cache) vor. Dadurch kann die Parallelität bei der Datensicherung auch ohne Vervielfachung der Anzahl realer MBK-Laufwerke erhöht werden. Erst nach der vollständigen Übertragung der Daten auf das virtuelle MBK-Gerät und nach dessen Entladen werden die Daten physikalisch auf MBK gespeichert. Mit diesen Vorteilen wird ETERNUS CS HE für den Nearline-Betrieb im BS2000-Umfeld dringend empfohlen.

Die im Folgenden angegebenen Performance-Daten beziehen sich auf CentricStor V4.0 (CentricStor ist das Vorgängermodell von ETERNUS CS HE). Die daraus abgeleiteten Empfehlungen gelten aber auch für die aktuellen Versionen von ETERNUS CS HE.

MBK-Archivsysteme Scalar mit LTO-Technologie

Parallel oder alternativ zu ETERNUS CS HE können die LTO-Geräte der realen MBK-Archivsysteme Scalar der Fa. Quantum Corp. direkt an den BS2000/OSD-Server angeschlossen werden. Informationen zu diesen MBK-Archivsystemen finden Sie im Handbuch „ROBAR“ ([24]).

Im Unterschied zum ETERNUS CS HE ist die Parallelität durch die Anzahl der MBK-Laufwerke begrenzt.

MBK-Archivsysteme an SQ-Servern

Für eine Bandautomatisierung steht an den SQ-Servern optional ein MBK-Archivsystem ETERNUS LT40 (SQ200) bzw. ETERNUS LT40 S2 (SQ210) mit einem oder zwei LTO-Laufwerken zur Verfügung.

Durchsatz und Komprimierung bei LTO-Laufwerken



Die folgenden Ausführungen beziehen sich auf den Direktanschluss von LTO-Geräten an den BS2000/OSD-Server (ohne ETERNUS CS HE).

Die vom Server (z.B. einem BS2000-System) geschriebenen Daten werden zunächst in den Cache des LTO-Gerätes übertragen. Hier werden sie soweit möglich vom LTO-Gerät komprimiert. Die komprimierten Daten werden dann auf das physikalische Band im LTO-Laufwerk geschrieben. Beim Lesen wird die analoge Vorgehensweise angewandt.

Für LTO-Geräte gelten folgende Leistungswerte:

Gerätetyp	Maximaler Durchsatz	Kapazität
LTO-2	40 Mbyte/s	200 Gbyte
LTO-3	80 Mbyte/s	400 Gbyte
LTO-4	120 Mbyte/s	800 Gbyte
LTO-5	140 Mbyte/s	1.500 GByte

Diese Leistungswerte beziehen sich auf das Laufwerk bzw. die physikalischen Kassetten.

Davon zu unterscheiden sind folgende Daten:

Gerätetyp	Durchsatz (komprimiert)	Kapazität (komprimiert)
LTO-2	80 Mbyte/s	400 Gbyte
LTO-3	160 Mbyte/s	800 Gbyte
LTO-4	240 Mbyte/s	1.600 Gbyte
LTO-5	280 Mbyte/s	3.000 GByte

Diese Daten beziehen sich auf den Datentransfer zwischen Server und LTO-Gerät, also noch vor der Komprimierung (Schreiben), bzw. nach der Dekomprimierung (Lesen). Dabei wird ein Komprimierungsfaktor 2 angenommen. In BS2000/OSD liegen typische Komprimierungsfaktoren bei 2 bis 3. Damit ergibt sich für BS2000/OSD folgender Durchsatz:

- LTO-2: 70 bis über 100 Mbyte/s
- LTO-3: 160 bis über 240 Mbyte/s
- LTO-4: 240 bis über 360 Mbyte/s
- LTO-5: 280 bis über 420 Mbyte/s

Ein solcher Durchsatz ist in BS2000/OSD nur bedingt möglich und auch nur selten nötig. Bei S-Servern wird der Durchsatz ab LTO-3 durch den Kanal Typ FC (1 Gbit/s) begrenzt. Damit ergeben sich auf S-Servern weder beim Durchsatz noch beim Streamingverhalten große Änderungen. Bei SQ-Servern zeigen sich hier erheblich bessere Werte.

LTO-4- und LTO-5-Laufwerke unterstützen hardwareseitig die „Tape Encryption“. Die Datenrate verringert sich bei eingeschalteter Tape Encryption nach Herstellerangaben lediglich um ca. 1%.

Streaming-Grenze bei LTO-Laufwerken

LTO-Geräte arbeiten am besten, wenn die Daten möglichst gleichmäßig mit dem maximalen Durchsatz zum Band übertragen werden.

Wenn vom Server (unter Berücksichtigung der Komprimierungsrate) mit einer Datenrate geschrieben wird, die unterhalb des maximalen Durchsatzes liegt, dann passt sich das LTO-Gerät dieser niedrigeren Datenrate an („Rate Adaption“). Damit können Lastschwankungen ausgeglichen werden. Die untere Grenze für die „Rate Adaption“ ist die „Streaming-Grenze“ des LTO-Geräts. Die Streaming-Grenze ist Geräte- und Hersteller-abhängig und liegt bei etwa

- 15 Mbyte/s für LTO-2
- 30 Mbyte/s für LTO-3, LTO-4 und LTO-5

Wenn die Datenrate unter die Streaming-Grenze fällt, dann kommt es zum „Start-/Stopp-Betrieb“. Band und Laufwerk unterliegen dabei erhöhten mechanischen Belastungen, die die Lebensdauer verkürzen können. Ein lang andauernder Start-/Stopp-Betrieb sollte deshalb vermieden werden. Kurzzeitiger Start-/Stopp-Betrieb beim Montieren bzw. Demontieren ist nicht zu vermeiden.

Die richtige Kassettenwahl entscheidet über Qualität und Langlebigkeit: Kassetten und Laufwerk müssen zusammenpassen. Es wird dringend empfohlen, nur qualifizierte Kassetten mit der Kennzeichnung „Preferred Quality“ zu verwenden.

Bei einem Komprimierungsfaktor 2 müssen vom Server also bei LTO-2 30 Mbyte/s, bei LTO-3, LTO-4 und LTO-5 60 Mbyte/s transferiert werden um die Streaming-Grenze zu überschreiten. Wenn dies bei langsamen Servern nicht erreicht wird, dann kann, um mögliche Bandmaterialschäden zu vermeiden, in BS2000/OSD mit dem Dienstprogramm IORM die Komprimierung auf dem LTO-Laufwerk ausgeschaltet werden. Mit IORM kann die Komprimierung auch dynamisch, in Abhängigkeit von der aktuellen Transferrate, ein- und ausgeschaltet werden. Bei ausgeschalteter Komprimierung ist der Durchsatz von BS2000/OSD zum LTO-Gerät und weiter zum Laufwerk gleich.

Leistung der Bandspeichersysteme

Bei der Datensicherung werden die Daten auf ein oder mehrere Bandspeichersysteme geschrieben, bei der Datenrekonstruktion entsprechend gelesen.

Dieser Abschnitt betrachtet die Leistung von Bandspeichersystemen isoliert, also ohne die weiteren Komponenten für die Datensicherung.

Die Leistung neuerer Bandspeichersysteme kann erst mit Kanälen Typ FC (S-Server) bzw. FC-Anschluss voll genutzt werden. Wenn die Bandspeichersysteme über Kanal Typ S angeschlossen werden, dann kann ein Durchsatz von maximal 15 Mbyte/s erreicht werden.

Mit BS2000/OSD kann durch die „großen Bandblöcke“ (256 KB statt 32 KB pro Ein-/Ausgabe) die Leistung bis zum Faktor 2 gesteigert werden.

Die folgenden Tabellen zeigen den möglichen Durchsatz verschiedener Bandspeichersysteme. Die Last entspricht einer Datensicherung mit HSMS/ARCHIVE bzw. FDDRL mit großen Bandblöcken. Die Ein-/Ausgaben werden nur vom BS2000-Hauptspeicher (nicht von Platte) zum MBK-Gerät durchgeführt.

Der Durchsatz ist jeweils für schwer komprimierbare Daten (Komprimierungsfaktor 1) und für normal komprimierbare Daten (Faktor 2 - 3) angegeben.

S-Server, Anschluss über Kanal Typ FC (1 Gbit/s):

		Durchsatz (Mbyte/s)			
		1		2 - 3	
Komprimierungsfaktor					
System	Geräte	Schreiben	Lesen	Schreiben	Lesen
Scalar i2000	LTO-3	80	60	80	60
Scalar i500	LTO-3	80	80	95	95
Scalar i6000	ab LTO-3	80	80	95	95
ETERNUS CS HE	VTA	90	75	90	75

Mit fast 100 Mbyte/s wird meist die Grenze des Kanals Typ FC erreicht.

Der Einsatz von LTO-4 oder LTO-5 führt zu keiner weiteren Steigerung der Performance.

SQ-Server erreichen mit ihren schnelleren PCI-Controllern einen Durchsatz von 200 MByte/s und mehr.

5.4.4 Plattenspeichersysteme

Eigenschaften der Datensicherungslast

Bei der Datensicherung werden die Daten sequenziell gelesen bzw. geschrieben. Mit HSMS/ARCHIVE bzw. FDDRL erfolgt dies in mehreren parallelen, asynchronen Datenströmen von und zum Plattenspeichersystem. Normalerweise werden sehr große Datenmengen übertragen. Die Caches der Plattenspeichersysteme kommen dabei an ihre Grenzen. Deshalb muss mit Read-Ahead-Strategien von den physikalischen Platten gelesen bzw. mit Delayed Fast Writes auf die physikalischen Platten geschrieben werden. Für den erreichbaren Durchsatz ist deshalb die Leistung der physikalischen Platten und der RAID-Gruppen entscheidend. Die heutigen Plattenspeichersysteme erreichen bei einer Datensicherungslast auch mit Read-Ahead bzw. Delayed Fast Write einen sehr guten Durchsatz.

Leistung von Plattenspeichersystemen

Bei der Datensicherung werden die Daten von ein oder mehreren Plattenspeichersystemen gelesen, bei der Datenrekonstruktion entsprechend geschrieben. Dieser Abschnitt betrachtet die Leistung von Plattenspeichersystemen isoliert, also ohne die weiteren Komponenten für die Datensicherung.

Neuere Plattenspeichersysteme werden über Kanäle Typ FC bzw. den FC-Anschluss an den Server angeschlossen. Ihre Leistungsfähigkeit kann damit voll genutzt werden. Wenn Plattenspeichersysteme über Kanal Typ S angeschlossen werden, dann kann ein Durchsatz von maximal 8 - 12 Mbyte/s erreicht werden.

Um hohe Durchsätze zu erreichen, müssen neben einem mehrpfadigen Anschluss über Kanal Typ FC geeignete RAID-Level vorgesehen werden (siehe [Abschnitt „Eigenschaften externer Plattenspeichersysteme“ auf Seite 45](#)). Dabei bieten die RAID-Level RAID1/0, RAID5 oder RAID6 für die Datensicherung den besten Durchsatz. RAID1 ist insbesondere bei Lasten mit mittlerer bis hoher Parallelität wenig geeignet um einen hohen Durchsatz zu erzielen.

Die folgende Tabelle zeigt den möglichen Durchsatz mit verschiedenen Einstellungen. Die Messwerte gelten zum einen für den optimalen Fall mit wenigen großen Dateien (1000 MByte). Sie zeigen auch die Durchsatzverringering, wenn statt der optimalen sehr großen Dateien mittlere (10 MByte) oder kleine (1 MByte) Dateien gesichert werden.

Dabei wird das Lesen nur vom Plattenspeichersystem in den BS2000-Hauptspeicher ausgeführt, ohne das Schreiben auf Band. Das Plattenspeichersystem hat einen mehrpfadigen Anschluss über Kanal Typ FC bzw. FC-Anschluss.

**S-Server, Plattenspeichersystem ETERNUS DX oder Symmetrix DMX/VMAX,
Anschluss über Kanal Typ FC (1 Gbit/s, 4-pfadig)**

RAID-Level	Durchsatz Lesen (in Mbyte/s)					
	1000 Mbyte		10 Mbyte		1 Mbyte	
Dateigröße						
PAV (1 Alias)	ohne PAV	mit PAV	ohne PAV	mit PAV	ohne PAV	mit PAV
RAID1	70	75	50	70	35	40
RAID1/0	75	140	70	130	60	90
RAID5	75	130	70	125	50	90

Bei der Datensicherung ergibt sich für RAID-Gruppen mit „data striping“ (RAID1/0, RAID5, RAID6) gegenüber RAID1 ein erheblich höherer Durchsatz. Er liegt bei großen Dateien meist deutlich über dem Durchsatz, der auf der Bandseite erreicht wird.

Der Durchsatzverlust bei mittleren Dateigrößen (10 MByte) ist gering. Erst unterhalb der Grenze von etwa 1 MByte wird er signifikant.

5.5 Empfehlungen

Bei jeder Datensicherung sollte auf eine ausgewogene Leistung der verwendeten Band- und Plattenspeichersysteme geachtet werden. Dieser Abschnitt gibt Empfehlungen für einen möglichst hohen Durchsatz bei der Datensicherung.

Die allgemeinen Empfehlungen zur Geräteperipherie (siehe [Kapitel „Leistungsverhalten der Peripherie“ auf Seite 45](#)) gelten auch für die Datensicherung.

BS2000/OSD

- S-Server: „Dynamisches PAV“ (DPAV) verwenden
- Einsatz der Fibre Channel Anschlusstechnik (auf S-Servern Kanal Typ FC, nicht Kanal Typ S)

Bandspeichersysteme

- Einsatz von ETERNUS CS HE
- Wenn schnelle LTO-Bandgeräte (ab LTO-3) direkt an den Server angeschlossen werden sollen:
 - Bandmaterial: Nur qualifizierte Kassetten verwenden („preferred quality“)
 - Idealbetrieb oberhalb der Streaming-Grenze um Start-/Stopp-Betrieb zu vermeiden; ein zeitweises kurzzeitiges Unterschreiten der Streaming-Grenze kann toleriert werden

Plattenspeichersysteme

- Einsatz moderner Plattenspeichersysteme
- Einsatz schneller Platten
- Möglichst großer Cache
Dieser spielt zwar bei der Datensicherung nur eine geringe Rolle, ist aber bei normalem Betrieb sehr wichtig). Die Größe sollte so gewählt werden, dass bei einem OLTP-Betrieb die Read-Hit-Raten bei 60 - 80% liegen.
- Einsatz der RAID-Level RAID1/0, RAID5 oder RAID6
- Anforderungen zur Konfigurierung aus Sicht von BS2000/OSD berücksichtigen
- Mit PAV und RAID1/0, RAID5 oder RAID6 können auch „große“ Volumes (mit einer Größe von mehr als 32 Gbyte) verwendet werden
- Einsatz von FBA-Platten des Typs D3435 mit Datenformat NK2

6 Performance-Aspekte bei VM2000-Betrieb

VM2000 ermöglicht den gleichzeitigen Ablauf mehrerer, voneinander abgeschotteter BS2000-Gastsysteme auf einem realen S- oder SQ-Server. Die einzelnen Gastsysteme laufen auf virtuellen Maschinen (VMs) und verhalten sich funktional wie im Native-Betrieb.

Die Betriebsmittel CPU, Hauptspeicher und Geräte werden durch VM2000 den virtuellen Maschinen und damit den Gastsystemen zugeordnet.

Ein Hypervisor steuert den Ablauf der Gastsysteme auf den VMs. Insbesondere sorgt er für die Virtualisierung der globalen Betriebsmittel CPU und Hauptspeicher und bringt die ablaufbereiten CPUs der Gastsysteme auf den realen CPUs zum Ablauf (Scheduling).

- Auf S-Servern ist der VM2000-Hypervisor ein eigenes Lademodul von VM2000, das beim Einleiten des VM2000-Betriebs (automatisch) geladen wird.
- Auf SQ-Servern übernimmt der Xen-Hypervisor diese Rolle. Einige der Hypervisor-Aufgaben werden vom Trägersystem X2000 ausgeführt.

VM2000 unterscheidet auf S-Servern zwei Prozessorzustände:

- Hypervisor-Modus (in diesem Zustand läuft der VM2000-Hypervisor ab)
- VM-Modus (in diesem Zustand laufen die Gastsysteme ab)

Zu jedem Prozessorzustand gehört ein VM-Kontext (ein Satz von Registern über die Funktionszustände der CPU sowie VM-spezifische Informationen), der bei Aktivierung des Prozessorzustandes geladen wird.

Gegenüber dem Native-Betrieb entsteht VM2000-Overhead:

- Hypervisor-Overhead (Programmablauf im Hypervisor-Modus)
- indirekter Overhead (reduzierte Server-Leistung durch wechselnde VM-Kontexte)

Bei sorgfältiger Konfiguration liegt der VM2000-Overhead zwischen 5% und 12% der Server-Leistung. Wie hoch der VM2000-Overhead auf einem Server tatsächlich sein wird, hängt von der VM2000-Konfiguration ab. Beachten Sie die Hinweise im [Abschnitt „Optimale Konfiguration von VM2000“ auf Seite 132](#), um den VM2000-Overhead möglichst klein zu halten.

Die folgenden Abschnitte geben Hinweise zur optimalen Gestaltung des VM2000-Betriebes. Ausführliche Informationen zu VM2000 finden Sie im Handbuch „VM2000“ [33].

6.1 Verwaltung der CPU-Leistung mit VM2000

Ablaufbeispiele zu den Themen dieses Abschnitts finden Sie im Abschnitt „CPUs verwalten“ des Handbuchs „VM2000“ [33].

6.1.1 VM2000-Scheduling (S-Server)

Beim Scheduling einer VM unter VM2000 bringt der VM2000-Hypervisor eine ablaufbereite virtuelle CPU einer VM auf einer freien realen CPU zum Ablauf. Für das Scheduling benutzt VM2000 zum Ablaufzeitpunkt einer VM zwei unterschiedliche Verfahren:

- CPU-Zuteilung im Zeitscheibenverfahren (siehe unten)
- Feste CPU-Zuordnung (Dedizierte CPUs, siehe [Seite 120](#))

Dabei wird die CPU-Leistung des Servers abhängig von den gewählten Einstellungen für die VMs optimal auf die ablaufbereiten virtuellen Maschinen verteilt.



Auf SQ-Servern übernimmt der Xen-Hypervisor das Scheduling. Er verwendet dabei auch ein Zeitscheibenverfahren, das aber von VM2000 nicht beeinflusst wird. Die VM-spezifischen Steuergrößen Multiprozessorgrad, CPU-Quote und Begrenzung der CPU-Leistungsaufnahme stehen auch hier zur Verfügung.

CPU-Zuteilung im Zeitscheibenverfahren

Im Normalfall ist die Anzahl zugeschalteter realer CPUs in einem CPU-Pool **kleiner** als die Summe der zugeschalteten virtuellen CPUs aller laufenden VMs, die diesem CPU-Pool zugeordnet sind. VM2000 bringt eine virtuelle CPU auf einer realen CPU aus dem CPU-Pool nach einem Zeitscheiben-Verfahren zum Ablauf.

Die maximale Größe der Zeitscheibe wird für jede virtuelle CPU einer VM von VM2000 dynamisch im Intervall von 0,1 bis 8,0 Millisekunden festgelegt. VMs mit einer „sehr kleinen“ CPU-Quota erhalten dann auch eine kleinere Zeitscheibe. Die maximale Größe der Zeitscheibe wird auf den aktuellen BS2000/OSD-Servern nur in wenigen Fällen erreicht. Zumeist wird eine Zeitscheibe dadurch beendet, dass die virtuelle CPU der VM in den unterbrechbaren Wartezustand („Idle“) übergeht.

Der VM2000-Administrator kann über VM-spezifische Steuergrößen Einfluss auf die Verteilung der CPU-Leistung nehmen, siehe [Seite 119](#). Daneben verbessert VM2000 die Performance, z.B. durch die so genannte CPU-Affinität, siehe [Seite 120](#).

Die Scheduling-Priorität der einzelnen Gastsysteme errechnet sich aus den vorgegebenen CPU-Quoten und der in jüngster Vergangenheit verbrauchten CPU-Zeit.

VM-spezifische Steuergrößen

Die performance-relevanten Steuergrößen für eine VM im Zeitscheibenverfahren sind:

- Multiprozessorgrad der VMs/Gastsysteme (siehe [Seite 129](#))
- CPU-Quote der VM-Gruppe, CPU-Quoten bzw. Mitglieds-CPU-Quoten der VMs (siehe „[CPU-Anteil einer VM](#)“ unten)
- Leistungsbegrenzung der VM oder VM-Gruppe (MAX-CPU-UTILIZATION, siehe [Seite 120](#))
- VM-Privileg IO-PRIORITY (siehe [Seite 127](#))

Der Zeitverlauf, in dem ein Gastsystem den ihm zugewiesener Anteil an der CPU-Leistung aufnehmen kann, hängt sehr stark vom Nutzungsgrad der VM2000-Zeitscheibe ab, und zwar vom eigenen und von dem der anderen Gastsysteme. Dieser Nutzungsgrad ist bei CPU-intensiven Lasten hoch, bei Ein-/Ausgabe-intensiven Lasten niedrig.

Nutzt ein Gastsystem die ihm zustehende VM2000-Zeitscheibe nicht voll aus, so wirkt sich dies auf das Ein-/Ausgabe-Zeitverhalten der anderen Gastsysteme günstig aus. Bei einer Multiprozessor-VM versucht VM2000 den nicht genutzten Anteil der CPU-Leistung einer virtuellen CPU vorrangig auf die anderen virtuellen CPUs der VM zu verteilen.

Wird dagegen von einem Gastsystem die Zeitscheibe voll ausgenutzt, so kann es vor allem für Gastsysteme, deren Last fast nur aus Ein-/Ausgabe-intensiven Tasks besteht, zu einer deutlichen Verlängerung der Ein/Ausgabezeiten kommen. Mögliche Maßnahmen sind dann:

- Beschränkung des CPU-intensiven Gastsystems durch MAX-CPU-UTILIZATION (siehe [Seite 120](#)) und/oder
- Vergabe des Privilegs IO-PRIORITY (siehe [Seite 127](#))

`/SHOW-VM-STATUS INFORMATION=*SCHEDULE` informiert über den mittleren Wert der genutzten Zeitscheibe eines Gastsystems. Dabei wird die Wartezeit bis zur Aktivierung der virtuellen CPU als Maß für die „Dehnung der VM“ ausgegeben.

CPU-Anteil einer VM

VM2000 gibt in den Informationskommandos verschiedene CPU-Quoten für eine VM oder VM-Gruppe aus, mit denen der VM2000-Administrator die geplante und aktuelle Verteilung der CPU-Leistung beobachten kann:

- CPU-Q (CPU-Quote, `/SHOW-VM-ATTRIBUTES/-RESOURCES`)
ist die bei `/CREATE-VM` oder `/MODIFY-VM-ATTRIBUTES` eingestellte (MEMBER-)CPU-QUOTA der VM

- **EFF-Q** (effektive CPU-Quote, `/SHOW-VM-ATTRIBUTES/-RESOURCES INFORMATION=*CPU`) ist die normierte CPU-Quote der VM unter Berücksichtigung der Randbedingungen Multiprozessorgrad, Leistungsbegrenzung der VM und CPU-Pools (Summe der normierten CPU-Quoten aller **einggerichteten** VMs = 100%). EFF-Q zeigt den langfristig zu erwartenden CPU-Anteil der VM oder VM-Gruppe bei CPU-intensiver Last in allen Gastsystemen und wenn alle realen und virtuellen CPUs zugeschaltet sind.
- **CUR-Q** (aktuelle CPU-Quote, `/SHOW-VM-STATUS`) ist die normierte CPU-Quote der VM unter Berücksichtigung der Randbedingungen Multiprozessorgrad, Leistungsbegrenzung der VM und CPU-Pools (Summe der normierten CPU-Quoten aller **laufenden** VMs = 100%). CUR-Q zeigt den aktuell zu erwartenden CPU-Anteil der VM oder VM-Gruppe bei CPU-intensiver Last in allen Gastsystemen und in Bezug auf die aktuell zugeschalteten realen und virtuellen CPUs.

Leistungsbegrenzung einer VM

Wenn eine VM oder eine VM-Gruppe nie mehr als einen bestimmten Prozentsatz der CPU-Leistung des Servers erhalten soll (auch nicht bei freien CPUs), dann kann dies durch den Parameter `MAX-CPU-UTILIZATION` erreicht werden. Eine begrenzte CPU-Leistung kann z.B. im Rahmen von Service-Levels nötig werden. Der Parameter ermöglicht die Begrenzung überhöhter CPU-Leistungsaufnahme durch ein Gastsystem mit CPU-intensiver Last.

CPU-Affinität

Im Zeitscheibenverfahren strebt VM2000 beim Scheduling an, dass eine virtuelle CPU beim nächsten Scheduling-Vorgang auf der **gleichen** realen CPU zum Ablauf kommt. CPU-Caches und Initialisierungsdaten der VM bleiben erhalten und reduzieren den indirekten Overhead. Vorrangiges Ziel bleibt aber die Optimierung der Antwortzeit.

Feste CPU-Zuordnung (Dedizierte CPUs)

Die Anzahl zugeschalteter realer CPUs in einem CPU-Pool ist **größer oder gleich** der Summe der zugeschalteten virtuellen CPUs aller laufenden VMs, die diesem CPU-Pool zugeordnet sind. VM2000 ordnet jeder virtuellen CPU einer VM in diesem CPU-Pool genau eine reale CPU fest zu.

Der VM2000-Administrator kann in diesem Fall mit dem VM-Attribut `VM-ACTIVE-IDLE` noch bestimmen, ob eine VM auch dann die Kontrolle über eine reale CPU behält, wenn die darauf ablaufende virtuelle CPU der VM untätig ist (unterbrechbarer Wartezustand, „Idle“), siehe [Seite 123](#).

6.1.2 VM-Gruppen (S-Server)

Mehrere VMs können in **VM-Gruppen** zusammengefasst werden.

Die CPU-Nutzung wird dann von VM2000 zweistufig gesteuert. Mit diesem Konzept können logisch zusammengehörende Gastsysteme in Bezug auf die CPU-Nutzung als Einheit betrachtet werden. Z.B. können für einen Kunden mehrere VMs bereitgestellt und mit einem Service-Level versehen werden.

Über folgende Parameter der VM-Gruppen kann der VM2000-Administrator Einfluss auf die Verteilung der CPU-Leistung nehmen:

- die Priorisierung einer VM-Gruppe gegenüber anderen VM-Gruppen oder (Einzel-)VMs erfolgt in der gleichen Weise wie bei (Einzel-)VMs, d.h. über die Parameter `CPU-QUOTA` und `MAX-CPU-UTILIZATION` der VM-Gruppe
- die `MEMBER-CPU-QUOTA` steuert die Priorisierung der VMs, die die VM-Gruppe bilden, untereinander. Für die Scheduling-Priorität wird die `MEMBER-CPU-QUOTA` mit der `CPU-QUOTA` der VM-Gruppe relativiert
- neben der CPU-Leistungsaufnahme VM-Gruppe kann auch die CPU-Leistungsaufnahme jeder VM einer VM-Gruppe durch `MAX-CPU-UTILIZATION` begrenzt werden
- bei VM-Gruppen versucht VM2000 den nicht genutzten Anteil der CPU-Leistung einer VM vorrangig auf die anderen Mitglieder der VM-Gruppe zu verteilen

VM-Gruppen erzeugen keinen zusätzlichen VM2000-Overhead.

6.1.3 CPU-Pools

Die realen CPUs eines Servers, die für den VM2000-Betrieb zur Verfügung stehen, können unter VM2000 in unterschiedliche, disjunkte **CPU-Pools** aufgeteilt werden. Mit diesem Konzept kann eine bessere Hardware-Effizienz und CPU-Affinität bei Servern mit vielen CPUs erreicht werden. Eine VM wird genau einem CPU-Pool zugeordnet.

VM2000 versucht die CPU-Leistung der realen CPUs eines CPU-Pools proportional zu den CPU-Quoten auf die einzelnen Gastsysteme, die einem CPU-Pool zugeordnet sind, aufzuteilen. Eine obere Schranke ergibt sich aus der maximalen CPU-Leistungsaufnahme einer VM.

Folgende Eigenschaften von CPU-Pools haben Einfluss auf die Performance des Servers:

- CPU-Pools reduzieren den VM2000-Overhead, insbesondere bei einer größeren Anzahl realer CPUs
- CPU-Pools erlauben eine gezielte, aber strikt begrenzte Leistung (Service-Level) für eine Menge von VMs („begrenzte VM-Gruppe“) bei zusätzlich reduziertem VM2000-Overhead
- CPU-Pools auf S-Servern ermöglichen es, für kritische Produktivsysteme eine feste CPU-Zuordnung (dedizierte CPUs) herzustellen

Kriterien für die Gestaltung von CPU-Pools sind:

- Die Größe der CPU-Pools (die Anzahl der einem CPU-Pool zugeordneten realen CPUs) sollte geplant werden, so dass die zur Verfügung gestellte CPU-Leistung der Summe der geplanten Leistungsanteile für die dem CPU-Pool zugewiesenen VMs entspricht. Wie bei der CPU-Auslastung des ganzen Servers sollte auch die CPU-Auslastung der einzelnen CPU-Pools beim OLTP-Betrieb nicht höher als 75% liegen.
- Die Auslastung der realen CPUs des Servers sollte möglichst gleich sein. Dementsprechend sollten die CPU-Pools möglichst gleich ausgelastet sein.
- Bei CPU-Pools mit nur einer realen CPU ist zu erwarten, dass die Dehnung der Hardware-Bedienzeit (siehe [Seite 127](#)) höher ist als bei CPU-Pools mit mehreren realen CPUs.

Wenn sich der Leistungsbedarf der Gastsysteme im Laufe der Session ändert, dann können dem CPU-Pool dynamisch mit `/SWITCH-VM-CPU` neue CPUs hinzugefügt oder CPUs entzogen werden.

Kriterien für die Zuordnung von VMs zu den CPU-Pools sind:

- Bei Gastsystemen mit Anwendungen, deren Antwortzeitverhalten stark von der Dauer der Ein-/Ausgaben abhängt, sollten die konkurrierenden VMs bzw. Gastsysteme im gleichen CPU-Pool sorgfältig ausgewählt werden, damit sich die Dehnung der Hardware-Bedienzeit nicht ungünstig auswirkt.
Beispielweise sollte vermieden werden, dass einem CPU-Pool viele CPU-intensive Gastsysteme zugeordnet werden.
- Bei Gastsystemen mit besonders hoher Anforderung an die Performance kann der CPU-Pool mit so vielen realen CPUs ausgestattet werden, dass für jede aktive virtuelle CPU eine reale CPU zur Verfügung steht. VM2000 arbeitet auf S-Servern dann beim Scheduling mit einer festen CPU-Zuordnung.
- Mit dem Parameter `VM-ACTIVE-IDLE=*AT-DEDICATED-CPUS` (S-Server) behält eine VM mit fester CPU-Zuordnung auch dann die Kontrolle über die reale CPU, wenn die darauf ablaufende virtuelle CPU der VM untätig ist (unterbrechbarer Wartezustand, „Idle“). Damit werden auch die Kontextwechsel beim Gastsystem-Wartezustand („Idle“) vermieden. Auf S-Servern wird damit nahezu eine Performance wie im Native-Betrieb erreicht.



Für VMs mit `VM-ACTIVE-IDLE` ist keine Auswertung der `VM-ACTIVE`-Zeiten möglich (`/SHOW-VM-STATUS` oder `VM-Report` von `openSM2`).

Bei Bedarf können einzelne VMs oder ganze VM-Gruppen mit `/ASSIGN-VM(-GROUP)-TO-CPU-POOL` dynamisch anderen CPU-Pools zugeordnet werden.

6.2 Hauptspeicher

Jedes Gastsystem besitzt exklusiv einen Anteil am Hauptspeicher des Servers. Der Zugriff auf den Hauptspeicher erfolgt so performant wie im Native-Betrieb. Es ergeben sich also die gleichen Zugriffszeiten.

Die Hauptspeicherverteilung sollte geplant vorgenommen werden. Die benötigte Hauptspeichergöße für eine VM wird zunächst geschätzt oder aus der bisherigen Systemumgebung abgeleitet. Nach Messung der Paging-Rate mit SM2 muss eventuell eine Korrektur erfolgen. Die Paging-Rate sollte, entsprechend dem Lastprofil (TP- oder Dialog-orientiert), niedrig sein. Die aktuelle Hauptspeichergöße einer VM (und auf S-Servern auch die Lage) kann mit `/SHOW-VM-RESOURCES INFORMATION=*MEMORY` ermittelt werden.

Der Hauptspeicherbedarf für eine VM auf einem SQ-Server ist wegen der Code-Expansion und (ab SQ200) wegen dem (residenten) Speicherbedarf für JIT-Puffer höher als bei S-Servern. Auf einem SQ-Server werden standardmäßig 40% des Hauptspeichers für JIT-Puffer reserviert.

Die Hauptspeichergöße für den VM2000-Hypervisor (S-Server) wird von VM2000 abhängig vom Umfang der Peripherie berechnet.

Die minimale Hauptspeichergöße (`MIN-MEMORY-SIZE`) für eine VM sollte nur festgelegt werden, wenn die Funktionen der Hauptspeicher-Rekonfiguration (`/EXTEND- /REDUCE-VM-MEMORY`) verwendet werden sollen. Sie ist dann so hoch zu wählen, dass mindestens die residenten Speicheranforderungen im Gastsystem befriedigt werden können.

Der Bedarf an residentem Speicher ist abhängig vom Einsatz des Softwareprodukts DAB.

Der JIT-Speicher der SQ-Server (ca. 40% des Hauptspeichers) liegt ebenfalls im residenten Speicher. Die `MIN-MEMORY-SIZE` einer VM darf trotzdem mit einem kleineren Wert definiert werden, weil bei `/REDUCE-VM-MEMORY` eine entsprechende Größe des residenten JIT-Speichers dynamisch freigegeben wird, um die 40%-Regel für JIT zu erhalten. Bei `/EXTEND-VM-MEMORY` kehrt sich der Vorgang um.

Die maximale Hauptspeichergöße (`MAX-MEMORY-SIZE`) bestimmt auf SQ-Servern die Obergrenze, auf die der Hauptspeicher der VM erweitert werden kann (`/EXTEND-VM-MEMORY`). Standardwert ist die doppelte Größe der `MEMORY-SIZE` der VM. Wenn der Hauptspeicher einer VM nicht vergrößert werden soll, dann sollte für die maximale Hauptspeichergöße der Wert der `MEMORY-SIZE` gewählt werden.

Der aktuelle Verbrauch von residentem Speicher kann aus den Werten der Reportgruppe `MEMORY` von SM2 ermittelt werden: Residenter Speicher = `TOTAL - Pageable Frames`.

6.3 Peripherie

Kanäle

Auf S-Servern stehen die Kanäle allen VMs zur Verfügung. Deren Sharing erfolgt über die Firmware. Die Auslastung der Kanäle kann VM-übergreifend mit SM2 beobachtet werden.

Auf SQ-Servern stehen keine Kanäle im ursprünglichen Sinne zur Verfügung. X2000 emuliert Geräte mit einem virtuellen Ein-/Ausgabepfad, der an allen VMs verfügbar ist.

Geräte

Die vorhandenen Geräte werden beim Startup des Monitorsystems und der Gastsysteme dynamisch ermittelt. Damit kennen und verwalten VM2000, das Monitorsystem und alle Gastsysteme die gleichen Geräte.

Periphere Geräte müssen den einzelnen Gastsystemen vor der dortigen Nutzung zugeordnet werden. Dies erfolgt:

- **explizit** durch den VM2000-Administrator mit `/ADD-VM-DEVICES` bzw. `/SWITCH-VM-DEVICES` oder
- **implizit** durch ein Gastsystem mit dem Privileg `ASSIGN-BY-GUEST=*YES(...)` mit `/ATTACH-DEVICE`.
Der VM2000-Administrator muss dazu für die betreffenden VMs und Geräte diese Art der Zuordnung erlauben.



Bei der impliziten Zuordnung von Platten wird standardmäßig die Benutzungsart „Shared“ eingetragen.

Periphere Geräte können exklusiv einem einzelnen Gastsystem zugeordnet werden oder als „Shared“-Geräte mehreren Gastsystemen zur Verfügung stehen:

- **Exklusive Zuweisung** = das Gerät ist einem Gastsystem fest zugeordnet

Die Ein-/Ausgaben werden direkt eingeleitet, also wie im Native-Betrieb.

Für Gastsysteme im Wartezustand („Idle“) überprüft der VM2000-Hypervisor (S-Server) das Eintreffen von Ein-/Ausgabe-Termination-Interrupts und initialisiert das entsprechende Gastsystem gemäß seiner Scheduling-Priorität. Dieser Hypervisor-Aufwand ist minimal.

- **„Shared“ Zuweisung** = das Gerät steht mehreren Gastsystemen zur Verfügung
Die Art der Abwicklung der Ein-/Ausgabe auf S-Servern ist von folgender Bedingung abhängig:
 - a) Ist das Gerät nur einem Gastsystem zugeordnet, so werden die Ein-/Ausgaben wie bei der exklusiven Zuweisung direkt, d.h. ohne Zwischenschaltung des VM2000-Hypervisors, durchgeführt („Direct-I/O“).
In den Informationskommandos von VM2000 wird das Gerät durch die Anzeige SH(D) gekennzeichnet.
 - b) Ist das Gerät mehreren Gastsystemen zugewiesen, so unterbricht der VM2000-Hypervisor auf S-Servern alle Ein-/Ausgaben der Gastsysteme und serialisiert sie („Indirect-I/O“).
In den Informationskommandos von VM2000 wird das Gerät durch die Anzeige SH(I) gekennzeichnet.

Der CPU-Bedarf des VM2000-Hypervisors steigt für gemeinsam benutzte Geräte linear mit der Ein-/Ausgaberate. Der VM2000-Overhead steigt um ungefähr 0,4% pro 100 Ein-/Ausgaben pro Sekunde.

Auf SQ-Servern werden Platten-Ein-/Ausgaben als logische Aufträge über die RSC-Schnittstelle ausgeführt. Der Hypervisor-Overhead entfällt.

Bedienzeiten von Geräten (Messprogramm SERVICETIME von SM2) können zu einem Zeitpunkt nur von einer VM ermittelt werden.

Privileg IO-PRIORITY (S-Server)

Anwendungen führen meist synchrone Ein-/Ausgaben aus. Nach deren Start wartet die Task auf das Ende der Ein-/Ausgabe. Wenn kein weiterer Task ablaufbereit ist, erfolgt der Übergang in den Wartezustand („Idle“) und die reale CPU wird durch den Hypervisor neu verteilt. Nach Ende der Ein-/Ausgabe muss das Gastsystem auf die erneute Zuordnung einer realen CPU warten, bevor es das Ende der Ein-/Ausgabe bearbeiten kann. Die Verzögerung zwischen dem tatsächlichen Ende der Ein-/Ausgabe und ihrer Bearbeitung im Gastsystem wird als **Dehnung der Hardware-Bedienzeit** oder als **IO-Dehnung** bezeichnet.

CPU-intensive Gastsysteme können dadurch den Betrieb Ein-/Ausgabe-intensiver Gastsysteme stören. Hinweise darauf gibt `/SHOW-VM-STATUS INF=*SCHEDULE`, Ausgabespalten `%RUNOUT` und `%WAIT`.

Die IO-Dehnung ist um so geringer, je rascher dem Gastsystem wieder die CPU zugeteilt wird. Dazu kann auf S-Servern einer VM das Privileg `IO-PRIORITY=*YES` zugeordnet werden. Befindet sich eine virtuelle CPU einer solchen VM im Wartezustand, so wird diese beim Eintreffen eines Ein-/Ausgabe-Termination-Interrupts vom Hypervisor sofort auf einer realen CPU zum Ablauf gebracht, um das Ergebnis der Ein-/Ausgabe zu bearbeiten. Die Laufzeiten von Ein-/Ausgabe-kritischen Jobs verbessern sich deutlich.

Dem positiven Effekt, dass gleich gute Hardware-Bedienzeiten wie im Native-Betrieb erreicht werden, steht eine erhöhte Scheduling-Rate gegenüber. Dadurch steigt auch der VM2000-Overhead.

Es wird daher empfohlen, die Gastsysteme zunächst ohne Privileg einzurichten (also mit `IO-PRIORITY=*NO`) und erst im Bedarfsfall das Privileg `IO-PRIORITY=*YES` zu vergeben.

Das Privileg `IO-PRIORITY=*YES` bringt bei dedizierten CPUs keine Vorteile.

Unterstützung des Dienstprogramms IORM (I/O Ressource Manager)

Das Dienstprogramm IORM (siehe Handbuch „Dienstprogramme“ [6]) wird durch VM2000 unterstützt. Im VM2000-Betrieb sollte IORM auf dem Monitorsystem und allen Gastsystemen im Einsatz sein.

Folgende IORM-Funktionen werden im Zusammenspiel mit VM2000 ausgeführt:

- Dynamische I/O-Lastverteilung für Platten am FC-Kanal mit DPAV im VM2000-Betrieb (nur S-Server)

Dynamisches PAV (DPAV) weist autonom denjenigen Volumes Alias-Geräte zu, die am meisten davon profitieren. Im VM2000-Betrieb wird das eigentliche Umschalten von Alias-Geräten von DPAV im Monitorsystem koordiniert und durchgeführt.

- Optimierte Geräteauswahl im ETERNUS CS HE-Betrieb unter VM2000 mit DDAL (Dynamic Device Allocation)

Alternativ zur Nutzung der Geräte gemäß ihrer Generierung (wie in den Vorversionen) kann das Betriebssystem eine gleichmäßige Auslastung aller verfügbaren ICPs (Integrated Channel Processor) sicherstellen.

Unter VM2000 weitet die IORM-Funktion DDAL die optimierte (lokale) Geräteauswahl auf alle BS2000/OSD-Gastsysteme ab V7.0 eines Servers aus.

- Begrenzung der I/O-Leistungsaufnahme einzelner VM2000-Gastsysteme mit IOLVM (I/O Limit for Virtual Machines, S-Server)

Im VM2000-Betrieb können weniger wichtige, jedoch Ein-/Ausgabe-intensive Gastsysteme andere, wichtigere Gastsysteme im Ein-/Ausgabe-Betrieb behindern.

Die Funktion IOLVM des Dienstprogramms IORM kann solche Konfliktsituationen erkennen und bremst gezielt den Ein-/Ausgabebetrieb des eigenen Gastsystems, wenn sein spezifisches I/O-Limit für eine der gemeinsam benutzten I/O-Ressourcen (Kanal, Port, Pfad, gemeinsam genutzte Platte) überschritten wird.

Das I/O-Limit für IOLVM wird als maximale I/O-Leistungsaufnahme der VM im Operanden `MAX-IO-UTILIZATION` in den VM2000-Kommandos `/CREATE-VM` bzw. `/MODIFY-VM-ATTRIBUTES` festgelegt.

6.4 Gastsystem-Planung

6.4.1 Anzahl und Multiprozessorgrad der VMs

Gesamtanzahl und Multiprozessorgrad der VMs beeinflussen den VM2000-Overhead:

- Gesamtzahl der VMs bzw. Gesamtzahl der virtuellen CPUs
 - Je mehr unterschiedliche virtuelle CPUs auf einer realen CPU zum Ablauf kommen, desto weniger optimal wirken die CPU-Caches.

VM2000 (S-Server) reduziert durch CPU-Affinität (siehe [Seite 120](#)) die Anzahl der CPU-Wechsel, bei denen eine virtuelle CPU auf einer anderen realen CPU als zuvor zum Ablauf gebracht wird.
Mit CPU-Pools kann die Menge der ablaufenden virtuellen CPUs pro realer CPU eingeschränkt werden.
 - Der VM2000-Hypervisor-Overhead zur Verwaltung der CPU-Queues auf S-Servern hängt direkt von der Anzahl virtueller CPUs in einem CPU-Pool ab.



Das Verhältnis der Anzahl der aktiven virtuellen CPUs (der Gastsysteme mit Produktionslast) zur Anzahl der realen CPUs sollte in allen CPU-Pools kleiner oder gleich 3 sein.

Die Anzahl der Gastsysteme sollte dementsprechend klein sein.

Bei geringeren Leistungsanforderungen an die einzelnen Gastsysteme kann auch eine höhere Anzahl VMs (max. 15) eingerichtet werden. In jedem Fall ist zu prüfen, ob der verfügbare Hauptspeicher für die benötigten Gastsysteme ausreichend groß ist.

- Multiprozessorgrad einer VM (= maximale Leistungsaufnahme der VM)

Die Auslastung der virtuellen CPUs einer VM beeinflusst den Nutzungsgrad der Zeitscheibe und damit die Häufigkeit der Scheduling-Vorgänge. Hoher Nutzungsgrad der Zeitscheibe und niedrige Häufigkeit der Scheduling-Vorgänge sind performance-günstig.

Der Multiprozessorgrad einer VM sollte sich am Spitzenbedarf des Gastsystems orientieren. Er sollte möglichst niedrig gewählt werden. Bei geringer Last sollten virtuelle CPUs im Gastsystem weggeschaltet werden.

6.4.2 Einstellung der CPU-Quoten

Die Einstellung der CPU-Quoten für die VMs sollte in mehreren Schritten erfolgen:

1. Grobeinstellung der CPU-Quoten

In erster Näherung wird empfohlen, die CPU-Quoten proportional zum erwarteten CPU-Verbrauch der einzelnen Gastsysteme festzusetzen.

Beispiel

Auf einem Bi-Prozessor sind zwei Gastsysteme (VM1, VM2) geplant, welche schätzungsweise 50% bzw. 20% der Rechnerleistung benötigen.

Es ergeben sich die folgenden CPU-Quoten:

VM1: CPU-Q=50 (normierte EFF-Q=71,43)

VM2: CPU-Q=20 (normierte EFF-Q=28,57)

Da das Gastsystem VM2 den Server nur zu ca. 20% auslasten wird, soll es als Mono-Gastsystem betrieben werden. Das Gastsystem VM1 benötigt auf jeden Fall beide CPUs.

2. Leistungsbegrenzung

Die Begrenzung der CPU-Leistung für eine VM (siehe [Seite 120](#)) ist so zu wählen, dass deren normale Leistungsaufnahme bei etwa 75% der Leistungsbegrenzung liegt.

3. Messung

Mit `/SHOW-VM-STATUS` und `SM2` (siehe [Seite 139](#)) werden die VM und das Gastsystem beobachtet. Anhand der Messwerte ist zu prüfen, ob der tatsächliche CPU-Zeitverbrauch und das Performance-Verhalten den Erwartungen entsprechen.

Wenn die Gastsysteme annähernd die erwartete Leistung aufnehmen, kann die Einstellung so belassen werden.

4. Leistungsbedarf prüfen

Wenn die Gastsysteme **nicht** die erwartete Leistung aufnehmen, dann ist zu prüfen, wie hoch der CPU-Bedarf der einzelnen Gastsysteme tatsächlich ist. Dazu sollten die Gastsysteme nur einzeln gestartet werden, d.h. ohne Konkurrenz durch eine andere VM. Wenigstens sollten die Gastsysteme nacheinander mit einer jeweils sehr hohen effektiven CPU-Quote priorisiert werden.

Man kann dann davon ausgehen, dass der gemessene Prozentwert `VM-ACTIVE` dem tatsächlichen CPU-Bedarf des Gastsystems entspricht. Die CPU-Quoten können jetzt anhand dieser Werte neu berechnet werden.



Im Hinblick auf die Belastung des Servers durch sämtliche Gastsysteme zusammen, empfiehlt es sich, auf S-Servern CPU-Auslastungen > 75% bei OLTP-Betrieb zu vermeiden.

5. Feineinstellung der CPU-Quoten

Auch nach korrekter Grobeinstellung (d. h. CPU-Quoten entsprechen dem Leistungsbedarf der Gastsysteme), kann die Leistungsaufnahme einzelner Gastsysteme stark vom erwarteten Wert abweichen. Ein-/Ausgabe-intensive Gastsysteme nehmen i.d.R. unter Soll, CPU-intensive Gastsysteme über Soll an Leistung auf.

Ferner kommt es vor, dass die Antwortzeiten wichtiger Anwendungen nicht den Anforderungen entsprechen. In solchen Fällen muss nachgebessert werden, d. h. einzelne CPU-Quoten werden höher („Überplanung“) bzw. niedriger eingestellt, als die Grobeinstellungsempfehlung vorgibt.

Die Korrekturen der CPU-Quoten können durchaus eine Größenordnung von Faktor 2 annehmen. Ob und wie stark nachgebessert wird, hängt von den Performance-Anforderungen an die Gastsysteme ab. In diesem Schritt ist eine enge Zusammenarbeit mit den Gastsystem-Verwaltern unerlässlich.

6. VM-Gruppen bilden (S-Server)

Abschließend können VMs zu VM-Gruppen zusammengefasst werden.

Bei Nutzung von VM-Gruppen ist eine Feineinstellung der `MEMBER-CPU-QUOTA` für die Gruppenmitglieder ebenfalls zu empfehlen.

Eine Überplanung der `CPU-QUOTA` der VM-Gruppe wird nicht empfohlen. Falls eine Überplanung notwendig sein sollte, so sollten anstelle von VM-Gruppen einzelne VMs geplant werden.

6.4.3 Optimale Konfiguration von VM2000

Mit der stetig steigenden Leistungsfähigkeit der BS2000/OSD-Server haben sich die Zeitverhältnisse geändert, unter denen das Scheduling der VM2000-Gastsysteme stattfindet:

- Bei wenig CPU-intensiven Anwendungen (z.B. OLTP-Betrieb oder Batch-Betrieb mit vielen Ein-/Ausgaben oder sonstigen Unterbrechungen) entsteht ein Bedarf nach vielen, meist nur sehr kurzen Zeitintervallen, in denen CPU-Leistung benötigt wird („CPU-Zeitscheiben“).
- Bei Anlagenauslastungen von deutlich unter 100% werden normalerweise nur wenige dieser kurzen CPU-Zeitscheiben unmittelbar nacheinander verwendet; dann geht die CPU wieder in den Wartezustand („Idle“) über.
- Bei TP-Lasten ist somit mit einer starken Zersplitterung des CPU-Bedarfs in kleine CPU-Zeitscheiben sowie häufigen, kurzen Wartezuständen der CPU zu rechnen.
- Die Länge dieser CPU-Zeitscheiben sinkt mit zunehmender Leistung der CPU.
- Als Folge davon steigt die Zahl der Scheduling-Vorgänge von VM2000 mit wachsender CPU-Leistung (bzw. wachsender Server-Leistung) und mit steigendem Multiprozessorgrad der VMs.
Die maximale Größe einer CPU-Zeitscheibe wird nur noch in Ausnahmefällen genutzt. Je höher der Multiprozessorgrad einer VM ist, umso weniger wird die maximale Größe der CPU-Zeitscheiben genutzt.

Folge der steigenden Zahl der Scheduling-Vorgänge von VM2000 ist damit auch ein geringfügig gestiegener VM2000-Overhead - dies aber bei sehr guten Durchsatzwerten und guten Antwortzeiten.

Für OLTP-Anwendungen wird daher empfohlen, die Prioritäten-Steuerung zwischen den DB-Server- und DB-Client-Tasks so einzustellen, dass das Auftragsbuch der DB-Server-Tasks immer gefüllt ist. Damit wird Overhead durch zusätzliche Kommunikation vermieden.

Darüber hinaus gibt es eine Reihe von Maßnahmen, mit denen Sie den VM2000-Overhead auf das notwendige Minimum begrenzen können:

- Halten Sie die Anzahl der virtuellen CPUs (den Multiprozessorgrad) der VMs so klein wie möglich (siehe [Abschnitt „Anzahl und Multiprozessorgrad der VMs“ auf Seite 129](#)):
 - ▶ Legen Sie den Multiprozessorgrad einer VM individuell fest. Verwenden Sie nicht einfach den Multiprozessorgrad des Servers.
 - ▶ Wählen Sie den Multiprozessorgrad einer VM nur so groß, dass die erforderliche (Spitzen-)Last bearbeitet werden kann.
 - ▶ Verteilen Sie, wenn möglich, die Last auf mehrere VMs mit kleinerem Multiprozessorgrad.
 - In Messungen auf einem S-Server mit 4 CPUs wurde eine Last zunächst auf 2 Gastsysteme mit je 4 CPUs und anschließend auf 4 Gastsysteme mit je 2 CPUs verteilt. Es zeigte sich, dass die Last-Verteilung auf 4 Gastsysteme mit je 2 CPUs nur knapp die Hälfte an VM2000-Overhead verursachte.
 - ▶ Starten Sie keine Gastsysteme „auf Vorrat“. Solche Gastsysteme verursachen auch ohne Last VM2000-Overhead.
 - ▶ Sie können Gastsysteme mit einem geeigneten Multiprozessorgrad für Lastspitzen einrichten. Schalten Sie aber in Zeiten ohne Lastspitzen die nicht benötigten CPUs im Gastsystem weg (/DETACH-DEVICE).
- Verwenden Sie auf S-Servern nach Möglichkeit dedizierte CPUs (siehe [Abschnitt „VM2000-Scheduling \(S-Server\)“ auf Seite 118](#)):

Im VM2000-Betrieb ohne zusätzliche CPU-Pools werden alle verfügbaren realen CPUs für das Scheduling der (virtuellen) CPUs aller Gastsysteme verwendet. Für Systeme, an die besonders hohe Performance-Anforderungen gestellt werden (z.B. für Produktivsysteme, im Gegensatz zu Test- oder Entwicklungssystemen), empfiehlt sich daher ein Betrieb, der einem Native-Betrieb sehr nahe kommt.

- ▶ Stellen Sie solchen Systemen so viele reale CPUs zur Verfügung, wie es (in Summe) zugeschaltete virtuelle CPUs in den betreffenden Gastsystemen gibt.

Dies führt dazu, dass jede virtuelle CPU eine eigene reale CPU zur exklusiven Nutzung zur Verfügung hat („dedizierte CPU“). Somit wird unnötiger Overhead durch vermeidbare Cache-Verluste und Wartezeiten auf eine freie, reale CPU oder auf Multiprozessor-Locks verhindert.

Folgende Konfiguration ist beispielsweise dafür geeignet:

- ▶ Richten Sie neben dem Standard-CPU-Pool einen weiteren CPU-Pool für kritische Produktivsysteme mit dedizierten realen CPUs ein. Alle anderen Gastsysteme verwenden den Standard-CPU-Pool mit den verbleibenden realen CPUs.

- ▶ Nutzen Sie auf S-Servern für VMs mit dedizierten CPUs zusätzlich das VM-Attribut `VM-ACTIVE-IDLE`.

Diese Funktion verhindert, dass virtuelle CPUs im Wartezustand („Idle“) die reale CPU verlassen. Es gibt kaum Hypervisor-Overhead.

Eine Messung auf einem S-Server mit 4 CPUs und nur einer VM hat gezeigt, dass sich bei Verwendung dedizierter CPUs und Nutzung des VM-Attributs `VM-ACTIVE-IDLE` der VM2000-Overhead um 75% gegenüber der Verwendung dedizierter CPUs **ohne** Nutzung des VM-Attributs `VM-ACTIVE-IDLE` reduziert.



Für VMs mit `VM-ACTIVE-IDLE` ist keine Auswertung der VM-ACTIVE-Zeiten möglich (`/SHOW-VM-STATUS` oder VM-Report von openSM2).

- Vermeiden Sie „shared“ zugewiesene Platten nach Möglichkeit:

Siehe dazu die Ausführungen zur exklusiven und zur „shared“ Zuweisung im [Abschnitt „Peripherie“ auf Seite 125](#).

Messungen haben ergeben, dass sich für je 100 IO/s auf „shared“ zugewiesene Platten mit mehr als einem Gastsystem der VM2000-Overhead um 0,4% erhöht.

- ▶ Beschränken Sie die Nutzung von „shared“ zugewiesenen Platten unter VM2000 auf das nötige Maß.
- Achten Sie bei OLTP-Betrieb darauf, dass die gesamte CPU-Auslastung des Servers den von uns empfohlenen Wert von ca. 75% **nicht** überschreitet.
- Verwenden Sie KVP-Konsolen. Sie verursachen weniger VM2000-Overhead als virtuelle Konsolen.

6.5 Systembetreuung im Gastsystem

VMs bzw. Gastsysteme mit Batch-orientierter, CPU-intensiver Last werden vom VM2000-Scheduling bevorzugt. Dialog-orientierte Gastsysteme mit häufigen Idle-Zeiten werden durch die IO-Dehnung in ihren Antwortzeiten behindert. Nötigenfalls muss eine Feineinstellung der CPU-Quoten erfolgen, siehe [Seite 131](#).

Die Task-Prioritäten in einem Dialog-orientierten Gastsystem wirken weniger stark. Die Prioritäten müssen ggf. neu eingestellt werden, siehe folgenden Abschnitt.

Ein Gastsystem, das seine Zeitscheibe nur in geringem Umfang nutzt, wird einerseits durch die IO-Dehnung, andererseits durch schlechtere Cache-Hitratens gegenüber dem Native-Betrieb behindert. Insbesondere auf S-Servern kann der Ablauf in einem CPU-Pool mit dedizierten CPUs und dem VM-Attribut `VM-ACTIVE-IDLE` helfen, siehe [Seite 120](#).

Bei Einsatz von PCS kann die Dehnung einer Kategorie in beide Richtungen beeinflusst werden, siehe [Abschnitt „PCS unter VM2000“ auf Seite 137](#).

6.5.1 Einstellung der Prioritäten

Teilt man eine IT-Last auf mehrere Gastsysteme auf, so verändert sich die Konkurrenzsituation der einzelnen Lastkomponenten. Daher müssen manchmal Prioritäten neu eingestellt werden. Dies soll an folgendem Beispiel erläutert werden.

Auf einem Server im Native-Betrieb laufen folgende Lastkomponenten:

- eine UDS/UTM-Anwendung
- mehrere Batch-Datenbankanwendungen
- einige Dialog-Tasks mit Programmentwicklungstätigkeit ohne Datenbankzugriff

Prioritäten und CPU-Verbrauch waren bisher wie folgt:

	UDS-Tasks	UTM-Tasks	Batch	Dialog1	Dialog2
Task-Anzahl	1	8	2 - 6	3	10
Priorität	150	150	255	180	210
Auslastung (%)	20	30	10	5	10

Jetzt soll die IT-Last folgendermaßen auf zwei Gastsysteme aufgeteilt werden: Die Entwicklungstätigkeiten sollen auf ein eigenes Gastsystem ausgelagert werden. Alle anderen Komponenten sollen wegen der gemeinsam genutzten Datenbanken auf einem anderen System laufen.

VM	Bezeichnung	CPU-Quote	Last
VM1	TP-System	80	UDS/UTM + Batch
VM2	Dialog-System	20	Dialog1 + Dialog2

Infolge der wegfallenden Dialog-Last verbessert sich die Situation für die Batch-Tasks, da diese nun immer dann laufen können, wenn die TP-Last auf Ein-/Ausgaben wartet. Der Durchsatz CPU-intensiver Batch-Tasks verbessert sich, da diese die reale CPU an das Gastsystem binden.

Reine Dialogsysteme haben gewöhnlich einen hohen Idle-Anteil. Wegen der niedrigen CPU-Quote muss das Dialog-System nach jedem IDLE-Zustand relativ lange warten. Dadurch können sich die Antwortzeiten der Dialoge verschlechtern.

Auch die Bevorzugung der höher prioren Dialoge gegenüber den niedriger prioren nimmt ab. Um nach der Lasttrennung dieselben Ergebnisse zu erhalten, sind folgende Maßnahmen durchzuführen:

- Bei gravierender Verschlechterung der Dialog-Antwortzeiten muss die CPU-Quote des Dialogsystems erhöht werden. Dabei sind immer die TP-Antwortzeiten und der Batch-Durchsatz zu beobachten.
- Der Prioritätsabstand zwischen Dialog1 und Dialog2 muss vergrößert werden.
- Die Prioritäten der UDS- und UTM-Tasks müssen noch etwas verbessert werden, um den ursprünglichen Abstand zu den jetzt besser gestellten Batch-Tasks wieder herzustellen.

6.5.2 PCS unter VM2000

Läuft PCS (Performance Control Subsystem) unter VM2000, so wird intern die CPU-Quota des entsprechenden Gastsystems berücksichtigt, d.h. innerhalb eines Gastsystems sollte die Summe des S-Q MAX-Wertes für antwortzeitkritische Kategorien bei 100 liegen (\cong 100% der Gastsystem-Kapazität).

Gegenüber dem Native-Betrieb ist i.d.R. eine Korrektur der Parametrisierung der SERVICE-Bereiche (Bereich zwischen SERVICE-QUOTA MIN und MAX, gesteuert über die Dehnungsparameter REQUEST-DELAY MIN und MAX) erforderlich.

Die Dehnung (REQUEST-DELAY, siehe auch [Abschnitt „Einführung in das PCS-Konzept“ auf Seite 222](#)) ist folgendermaßen definiert:

$$\text{Dehnung} = \text{Laufzeit im Multiprogramming-Betrieb} / \text{Allein-Laufzeit}$$

Unter VM2000 können sich nun beide Laufzeiten verlängern. Die Alleinlaufzeit kann zum Beispiel durch die Verlängerung der Ein-/Ausgabezeiten durch andere Gastsysteme oder durch die Verminderung der Hardware-Leistung infolge geringerer Cache-Hitrate zunehmen.

Abhängig vom Lastprofil der einzelnen Kategorien eines Gastsystems und den Lastprofilen der anderen Gastsysteme können sich daher die PCS-Dehnungen von Kategorien gegenüber einem vergleichbaren Native-Betrieb sowohl vergrößern (was grundsätzlich zu erwarten wäre) als auch verkleinern.

Bei der Überprüfung bzw. Korrektur der Parameter SERVICE-QUOTA und REQUEST-DELAY sollte in folgender Reihenfolge vorgegangen werden:

- SERVICE-QUOTA MIN
Damit wird die Service-Quote eingestellt, die eine Kategorie im Normalbetrieb benötigt, also ohne Berücksichtigung von Lastspitzen.
- SERVICE-QUOTA MAX
Maximaler Prozentanteil am Leistungsvermögen des Gastsystems, der einer Kategorie für den Lastspitzenfall zugestanden wird.
- REQUEST-DELAY MIN
Der Systemverwalter sollte durch wiederholte Messungen herausfinden, ab welchem Wert REQUEST-DELAY ACTUAL ein Engpass bzw. unbefriedigendes Kategorieverhalten vorliegt, d. h. eine Erhöhung der Service-Rate erwünscht ist. Auf diesen Wert wird dann REQUEST-DELAY MIN eingestellt.
- REQUEST-DELAY MAX
Hiermit wird festgelegt, wie heftig einer Lastspitze entgegen gesteuert werden soll. Je näher der Wert bei REQUEST-DELAY MIN liegt, je kleiner er also ist, desto stärker ist die Reaktion auf Lastspitzen.

6.5.3 Gastsystem-Bedienung

Es wird aus Performance-Gründen empfohlen, die Gastsysteme über KVP-Konsolen zu bedienen.

Gastsysteme können auch über virtuelle Konsolen (über \$VMCONS) bedient werden. In diesem Fall sollte nur die Basisbedienung (Startup) über virtuelle Konsole abgewickelt werden. Die eigentliche Bedienung sollte über logische Konsolen mittels OMNIS durchgeführt werden.

OMNIS schließt sich direkt an \$CONSOLE des entsprechenden Gastsystems an und umgeht somit \$VMCONS. Die Synchronisation der Aufträge an \$VMCONS erfolgt über eine Börse. Diese Börse kann bei vielen Gastsystemen bzw. vielen Konsol-Ein-/Ausgaben zum Engpass werden.

6.6 Messwerkzeuge für VM2000

Es gibt zwei Messwerkzeuge für den VM2000-Betrieb:

- `/SHOW-VM-STATUS` führt im Monitorsystem Messungen auf VM2000-Ebene für VMs, VM-Gruppen, CPU-Pools und reale CPUs durch. Dabei werden zum Teil auch VM2000-internen Performance-Kenngrößen ausgegeben. Gemessen werden können die Hypervisor-Aktivitäten, Plan- und Ist-Werte der CPU-Auslastung und Daten des VM2000-Scheduling.
Mit dem Kommando können sowohl periodischen Messwerte als auch Messwerte der unmittelbaren Vergangenheit ausgegeben werden.
- das Mess-System openSM2 mit Online-Reports (INSPECTOR) und Langzeitauswertungen (ANALYZER).

Für die Beobachtung der Leistungsaufnahme der Gastssysteme und des Hypervisors mit openSM2 sind folgende ANALYZER-Reportgruppen zu empfehlen.

- In den Gastsystemen
CPU: CPU-Utilization
CPU-Auslastung in den Funktionszuständen TU, TPR, SIH. Diese Werte sind die vom jeweiligen Gastsystem ermittelten Auslastungswerte (bezogen auf die Anzahl der dem Gastsystem zur Verfügung stehenden virtuellen CPUs!).
- Im Monitorsystem
VM2000: VM2000-Utilization/VM2000-Hypervisor
CPU-Verbrauch, aktuelle CPU-Quota und aktuelle maximale Leistungsaufnahme pro VM. CPU-Verbrauch des Hypervisors und Hypervisor-Idle.
VM2000: CPU-Pool
CPU-Auslastung und Dimensionierung der CPU-Pools.
VM2000: VM-Gruppen
CPU-Verbrauch, aktuelle CPU-Quota und aktuelle maximale Leistungsaufnahme pro VM-Gruppe.



Einige SM2-Messwerte müssen bei VM2000-Betrieb neu interpretiert werden. Eine detaillierte Beschreibung befindet sich im Abschnitt „SM2-Einsatz bei VM2000-Betrieb“ des Handbuchs „openSM2“ [18].

7 Leistungsverhalten der Software

Dieses Kapitel behandelt die Performance-Kriterien beim Einrichten der Systemumgebung. Unter dem Begriff „Systemumgebung“ ist das gesamte Umfeld der Anwenderprogramme zu verstehen. Die Systemumgebung setzt sich zusammen aus dem Systemkern und den Subsystemen.

Systemkern

Der Systemkern von BS2000/OSD besteht aus Basissteuermechanismen und unterstützenden System-Tasks zur optimalen Nutzung der Betriebsmittel und zur Verständigung zwischen Anwender und Maschine sowie aus Systemroutinen zur Erleichterung der Programmierung. Die Aufgaben des Systemkerns, dem die höchste Priorität zugeordnet ist, lassen sich wie folgt zusammenfassen:

- Job-Management

Stellt die Verbindung nach außen dar:

- Logon und Logoff-Behandlung für die Benutzer der Terminals
- Verständigung mit dem Operator
- Verarbeitung des Auftrags-Eingabestroms

- Task-Management

Steuert und überwacht mit Hilfe der Basissteuermechanismen die Ausführung der User- und System-Tasks:

- Unterbrechungsanalyse
- Erzeugen und Beenden von Tasks
- Verwaltung der Betriebsmittel Hauptspeicher und CPU
- Steuerung der Tasks über Warteschlangen

- Data-Management

Übernimmt die Maßnahmen, die zur Behandlung von Datenbeständen erforderlich sind:

- Abwicklung der Ein-/Ausgabe zwischen Hauptspeicher und Ein-/Ausgabe-Einheiten
- Verwaltung der Datenträger
- Logische Zugriffsmethoden (SAM, ISAM, PAM, FASTPAM, DIV, BTAM)
- Unterstützung der HIPERFILEs zusammen mit DAB
(siehe [Abschnitt „Arbeiten mit HIPERFILEs“ auf Seite 176](#))

Um den Systemkern werden Subsysteme je nach den zu bewältigenden Aufgaben angeordnet (z.B. Editierhilfen, Compiler, DFÜ-Komponenten, Datenbanken).

Subsysteme

Die Subsysteme gliedern sich in zwei Gruppen:

- Privilegierte Subsysteme

Diese laufen im Funktionszustand TPR ab (z.B. SPOOL, NDM) und sind mit dem Systemkern so eng verbunden, dass sie nicht mit den für den Anwender bestimmten Ladeprogrammen geladen werden können.

- Nichtprivilegierte Subsysteme

Sie haben gegenüber BS2000/OSD den Status von normalen Anwenderprogrammen und bedienen sich daher auch nur der für den Anwender offenen Schnittstellen (z.B. Compiler, Editoren, Datenbanken, Dienstprogramme).

Einrichten der Systemumgebung

Ausgehend vom Systemkern bezeichnet man alle Tätigkeiten zur Erstellung eines aufgabenbezogenen Betriebssystems als Einrichten der Systemumgebung. Die Systemeinrichtung beinhaltet:

- Einstellen der Größe des Benutzeradressraums für S-Server
(siehe [Abschnitt „Größe des Benutzeradressraums“ auf Seite 143](#))
- Einrichten der Dateien
(siehe [Abschnitt „Performance-Kriterien beim Einrichten von Dateien“ auf Seite 145](#))
- Festlegen der HIPERFILEs
(siehe [Abschnitt „Arbeiten mit HIPERFILEs“ auf Seite 176](#))

Die nachfolgenden Hinweise sollen helfen, diese Arbeiten so auszuführen, dass die Leistungsfähigkeit von BS2000/OSD voll genutzt werden kann.

7.1 Größe des Benutzeradressraums

SQ-Server

Der Benutzeradressraum für SQ-Server ist in der Größe von 2 GB generiert und kann nicht verändert werden.

S-Server

Bei S-Servern setzt sich der Gesamtadressraum zusammen aus dem Benutzeradressraum (inklusive des Platzbedarfs der Shared-Produkte, Parameter SHRSIZE) und dem Systemadressraum. Für die Größe des Gesamtadressraums stehen auf S-Servern die Werte 1 GB oder 2 GB zur Verfügung.

Durch die Größe des Benutzeradressraums ergibt sich automatisch die Größe des Systemadressraums als Differenz zum nächsthöheren Wert des Gesamtadressraums. Die Angabe des Parameters SYSSIZE im Parameterservice ist bei der Systemeinleitung nicht mehr erforderlich.

Für S-Server wird standardmäßig ein Benutzeradressraum von 1.808 MB generiert. Diese Größe kann mit der Prozedur `SYSPRC.BS2000-EXEC.<version>` geändert werden (siehe Handbuch „Systeminstallation“ [31]).

Bei Konfigurationen mit höherem Bedarf an Systemadressraum muss der Benutzeradressraum auf 1.792 MB (SYSSIZE=256 MB), 1.664 MB (SYSSIZE=384 MB) oder 1.536 MB (SYSSIZE=512 MB) reduziert werden.

Die Unterteilung des Gesamtadressraums wurde u.a. zur Minderung des Hauptspeicherbedarfs für die Adressraumverwaltung (Segmenttafel) eingeführt. Abhängig von der Größe des Gesamtadressraums (1 oder 2 GB) wird folgender Hauptspeicher pro Task für die Segmenttafeln belegt:

- 4 KB bei 1.024 MB Gesamtadressraum
- 8 KB bei 2.048 MB Gesamtadressraum

Bei einer kleinen Anzahl von Tasks ist die Auswahl des größtmöglichen Adressraums (2 GB) unkritisch. Bei einer großen Anzahl von Tasks muss geprüft werden, ob der Mehrbedarf an Hauptspeicher durch einige Großanwendungen gerechtfertigt ist.

Pro MB belegtem Adressraum und Task werden für die Seitentafel-Verwaltung 3 KB Hauptspeicher benötigt. Belegen z.B. 10 Tasks vollständig den maximal zur Verfügung stehenden Benutzeradressraum von 1.808 MB, so wird allein für die Adressraumverwaltung (Segmenttafel und Seitentafel) folgender Hauptspeicher (residenter Klasse-3-Systemadressraum) benötigt:

$$\begin{array}{rcll} 10 * & 8 \text{ KB} & = & 80 \text{ KB} \quad \text{für Segmenttafel} \\ 10 * & 1.808 \text{ MB} * 3 \text{ KB/MB} & = & \underline{54.240 \text{ KB}} \quad \text{für Seitentafel} \\ & & & \underline{\underline{54.320 \text{ KB}}} \end{array}$$

Bei großen virtuellen Adressräumen ist auf ausreichend dimensionierte Seitenwechsellbereiche zu achten. Die benutzerspezifische Größe (/ADD-USER, /MODIFY-USER-ATTRIBUTES) ist sorgfältig festzulegen. Die Nutzung vieler großer Adressräume setzt einen entsprechend groß dimensionierten Hauptspeicher voraus.

7.2 Performance-Kriterien beim Einrichten von Dateien

BS2000-Benutzer und Systemadministratoren können zumeist nur die Lage der Dateien bzgl. der logischen Volumes beeinflussen. Auf physikalischer Ebene können sich mehrere logische Volumes auf einer gemeinsamen physikalischen Platte befinden. Die daraus resultierenden Auswirkungen auf die Performance und die Performance-Maßnahmen sind im [Abschnitt „Konfigurierung in BS2000/OSD“ auf Seite 52](#) beschrieben.

Informationen zur Performance von Net-Storage finden Sie im [Abschnitt „Net-Storage“ auf Seite 86](#).

In den folgenden Abschnitten dieses Kapitels wird für den (aktuellen) Begriff „logisches Volume“ als Synonym weiterhin der (bisherige) Begriff „Platte“ verwendet.

7.2.1 Logische Betriebsarten von Platten

Platten werden in BS2000/OSD logisch über die Volume-Serial-Number (VSN) angesprochen. Der Betrieb eines Volumes ist möglich als Public Volume oder als Private Volume. Public Volumes werden innerhalb von SF-Pubsets (Single-Feature-Pubsets), SM-Pubsets (System-Managed-Pubsets) oder Shared-Pubsets (bei Mehrrechnerzugriff) betrieben.

Gleichgültig, welche logische Betriebsart zum Einsatz kommt, die Zeit für die Abwicklung des einzelnen Zugriffs zu den Daten ist – bei gleichem Plattentyp – etwa gleich lang.

Durch den unterschiedlichen Aufwand bezüglich der Speicherplatzverwaltung bzw. der Koordinierung des Mehrrechnerzugriffs ergeben sich jedoch gravierende Unterschiede hinsichtlich der Anzahl durchzuführender Ein-/Ausgaben bei Aufruf von Funktionen mit Katalogzugriff.

Zusätzlich gibt es zur Erhöhung der Verfügbarkeit die Betriebsarten:

- Dual Recording by Volume (DRV)
Datenspiegelung, gesteuert durch BS2000/OSD
(Softwareprodukt DRV, siehe dazu Handbuch „DRV“ [7])
- Replikation (Plattenspiegelung)
Datenspiegelung, gesteuert durch die Plattenspeichersysteme ETERNUS DX und Symmetrix



Die Datenspiegelung ist ein Teil des RAID-Konzepts.

RAID (Redundant Arrays of Independent Disks) ist eine Architektur, die die Abbildung mehrerer physikalischer Plattenlaufwerke auf ein logisches Volume definiert, um Datenredundanz zu erzielen, siehe [Seite 48](#)).

Public Volumes

Die Speicherplatzverwaltung erfolgt durch das System und bietet die optimale Unterstützung für Aufrufe mit Katalogzugriff:

- Hoher Komfort für den Anwender durch stark vereinfachte Definitionen beim Einrichten, Erweitern und Löschen von Dateien.
- Geringere Anzahl von Verwaltungs-Ein-/Ausgaben gegenüber den Private Volumes (Verwaltung der F5-Kennsätze im Hauptspeicher, Unterstützung durch das Softwareprodukt SCA, siehe [Seite 150](#)).

Die explizite Vorgabe der Lage einer Datei kann für alle Pubsets durch die Systemverwaltung erfolgen. Für den Anwender ist dies nur möglich, wenn im pubset-spezifischen Benutzerkatalog das Recht zur physikalischen Allokierung eingetragen ist (/MODIFY-USER-ATTRIBUTES, Operand PHYSICAL-ALLOCATION). Dies ist wichtig beim Einrichten spezieller Systemdateien (siehe [Abschnitt „Einrichten von Systemdateien“ auf Seite 156](#)).

Pubsets sind die strategische Plattform für Weiterentwicklungen im Plattenbereich. Private Volumes bieten nicht die volle funktionale Unterstützung neuer Funktionen, z.B. im HIPERFILE-Konzept.

Ein Umstieg von privaten Volumes auf Pubsets wird daher empfohlen.

SF-Pubsets

Für die Speicherplatzverwaltung gilt das gleiche wie bei Public Volumes beschrieben. Zusätzlich zur Erhöhung der Verfügbarkeit ergeben sich aus Performance-Sicht folgende Vorteile (besonders wichtig bei Dialogbetrieb):

- Die Splittung des Dateikatalogs TSOSCAT auf die einzelnen Pubsets verbessert die Katalog-Performance bei intensiver Nutzung von Katalogfunktionen.
- Durch die Möglichkeit zur Steuerung der Ein-/Ausgabe-Verteilung über die Benutzerkatalog-Einträge durch die Systembetreuung können Anwendern mit besonders zugriffsintensiven Dateien getrennte Daten-Pubsets zugewiesen werden.

SM-Pubsets

SM-Pubsets bestehen aus mehreren Volume-Sets, die verschiedene, von der Systembetreuung vorgegebene Eigenschaften hinsichtlich Formatierung, Verfügbarkeit und Performance aufweisen können.

Im Vergleich zu SF-Pubsets, die aus genau einem Volume-Set bestehen, bieten SM-Pubsets eine bessere Steuerung der Ein-/Ausgabe-Verteilung durch die Systembetreuung bei gleichzeitig höherem Komfort für den Anwender:

- Durch die Angabe von Direktattributen oder einer Storage-Klasse kann der Anwender beim Einrichten einer Datei ihren Ablageort auf einem SM-Pubset bestimmen.
- Die Systemverwaltung ordnet jeder Storage-Klasse eine Volume-Set-Liste zu und definiert pro Volume-Set die gewünschten Performance-Anforderungen.

Ausführliche Informationen zu SMS (System Managed Storage) bzw. SM-Pubsets können dem Handbuch „SMS“ [32] entnommen werden.

Shared-Pubsets

Zwei oder mehrere gleichzeitig schreibberechtigte Server (oder VM2000-Gastsysteme) können auf gemeinsame Pubsets zugreifen (Ausnahme: Home- bzw. Paging-Pubsets). Die Server sind über eine BCAM-Verbindung gekoppelt und arbeiten nach dem Master-/Slave-Prinzip. Dabei werden sämtliche Katalogoperationen (RDCAT, WRCAT, ...) immer vom Master-Server durchgeführt. Der Einsatz von SCA ist deshalb nur für den Master sinnvoll.

Bei Berücksichtigung folgender Hinweise ist das Performance-Verhalten annähernd gleich wie bei „normalen“ Public Volumes:

- Derjenige Server, von dem die meisten Zugriffe erfolgen, wird als Master gewählt (Minimierung der serverübergreifenden Aktionen).
- Um den Aufwand für die serverübergreifenden Katalogaktionen in vertretbarem Rahmen zu halten, sollten pro RPF (Relativer Performance Faktor) nicht mehr als zwei OPEN/CLOSE-Operationen pro Sekunde auf dem Slave-Server durchgeführt werden.



Dateien auf Shared-Pubsets können mit DAB von jedem Server aus mit lokalem Caching unterstützt werden.

Private Volumes

Die Speicherplatzverwaltung erfolgt durch den Anwender. Durch exakte Vorgabe der Lage der Dateien auf den jeweiligen Volumes kann der Anwender Zugriffskonflikte bei häufig benutzten Dateien vermeiden.

Bei Dateibearbeitungsaufrufen mit Katalogzugriff ergibt sich eine deutlich höhere Anzahl von Verwaltungs-Ein-/Ausgaben gegenüber Public Volumes durch zusätzliche Zugriffe auf den VTOC-Bereich (Volume Table of Contents). Z.B. erfolgen pro OPEN/CLOSE-Paar 6 logische Zugriffe auf die F1-Kennsätze mit jeweils einer oder mehreren physikalischen Ein-/Ausgaben pro logischem Zugriff.

Ein günstiges Performance-Verhalten ist also nur dann erzielbar, wenn die Anzahl der Verwaltungs-Ein-/Ausgaben klein ist im Verhältnis zu den Nutz-Ein-/Ausgaben.

DRV: Dual Recording by Volume

DRV erhöht die Verfügbarkeit der auf den Platten gespeicherten Information durch doppelte Datenaufzeichnung auf zwei physikalischen Plattenlaufwerken. Dabei erfolgt die Verdopplung im Ein-/Ausgabesystem von BS2000/OSD.

Bei einer synchronen Schreib-Operation wird zuerst eine asynchrone Ausgabe auf das erste, anschließend eine synchrone Ausgabe auf das zweite Plattenlaufwerk gestartet.

Zur Durchführung einer Lese-Operation wird das Plattenlaufwerk mit der kleineren Warteschlange ausgewählt, wodurch eine unsymmetrische Auslastung der Original-Platte und der Spiegel-Platte entsteht. Durch die Wahl des schwächer ausgelasteten Plattenlaufwerks verkürzt sich die Ein-/Ausgabezeit (Software-Duration) bei lesenden Zugriffen im Vergleich zum SRV-Betrieb (Single Recording by Volume).

Gegenüber dem SRV-Betrieb kann etwa das gleiche Performance-Verhalten erwartet werden.

Kopiergeschwindigkeit

Sowohl bei der Rekonstruktion als auch bei der Egalisierung kann die Kopiergeschwindigkeit pro Pubset dynamisch mit /SET-DRV-PARAMETER, Operand COPY-SPEED=*LOW/*MEDIUM/*HIGH in drei Stufen eingestellt werden. Bei COPY-SPEED= *LOW werden die DRV-Kopieroperationen stark verzögert, sodass der Pubset für Ein-/Ausgaben von Anwendungen besser erreichbar ist. Die Voreinstellung ist *MEDIUM.

Kopierumfang

Bei der Rekonstruktion werden nicht alle Blöcke der Platte kopiert, sondern nur noch die im F5-Label als belegt gekennzeichneten Blöcke.

Nach einem Systemausfall oder nach einem abnormal beendeten Pubset-Export führt DRV eine Egalisierung durch, um die Plattenpaare wieder abzugleichen.

Neben den bereits bestehenden Varianten, alle Blöcke oder nur die Systemdaten zu kopieren, bietet DRV die Möglichkeit, nur die Systemdaten und die geöffneten Dateien zu egalisieren. Diese Option ist nur für SF-Pubsets oder Volume-Sets eines SM-Pubsets möglich. Sie kann nur vor dem Importieren des Pubsets bzw. Volume-Sets mit /SET-DRV-PARAMETER UNIT=*PUBSET(...)/*VOLUME-SET(...), EQUALIZE-DATA= *OPEN-FILES eingestellt werden.

Schnelle Rekonstruktion nach getrennter Verarbeitung

Bei getrennter Verarbeitung (z.B. Sicherung der Original-Daten, Batch-Verarbeitung der Spiegel-Daten) merkt sich DRV die veränderten Bereiche. Dadurch kann die Rekonstruktionszeit ggf. drastisch verkürzt werden, wenn nur der Änderungsumfang seit Beginn der getrennten Verarbeitung (im Mbyte-Bereich) statt der gesamten Dateien (im Gbyte-Bereich) kopiert werden muss.

Replikation (Plattenspiegelung)

Bei den Plattenspeichersystemen ETERNUS DX und Symmetrix steht das RAID1-Level hardwaremäßig zur Verfügung. Hier erfolgt die doppelte Datenaufzeichnung ohne weitere Software-Unterstützung. Siehe auch [Seite 48](#).

7.2.2 Beschleunigung der Katalogzugriffe mit SCA

SCA (Speed Catalog Access) ist ein empfohlenes Softwareprodukt zur Beschleunigung der Katalogdienste, das beim Suchen von Datei- und Jobvariablen-Einträgen die sequenzielle Suche nach Katalogblöcken durch den direkten Zugriff ersetzt.

Realisiert wird diese Vorgehensweise mit Hilfe von zwei Tabellen. Eine Tabelle gibt Aufschluss über den freien Platz in den Katalogblöcken, die andere ordnet den Datei- bzw. Jobvariablenamen die logische Blocknummer des Katalogblocks zu (Verweistabelle). Beide Tabellen werden von der SCA-Task aufgebaut.

SCA darf nur für SF-Pubsets aufgerufen werden. Bei SM-Pubsets ist der beschleunigte Zugriff auf Katalogeinträge bereits intern realisiert.

SCA wird in zwei Varianten angeboten: mit und ohne Task-Wechsel. Empfohlen wird der Einsatz der Variante „SCA ohne Task-Wechsel“.

- SCA ohne Task-Wechsel

Die Verweistabelle wird im Klasse-4-Speicher aufgebaut und ist so für alle Tasks erreichbar. Alle SCA-Routinen (außer dem Aufbau der Tabellen) laufen unter der zugehörenden Task ab. Der Aufwand für die Task-Wechsel entfällt, der Aufwand für Serialisierungsmaßnahmen ist minimal.

- SCA mit Task-Wechsel

Die Verweistabelle wird von der SCA-Task in ihrem Klasse-5-Speicher verwaltet. Jede Katalogdienst-Anforderung einer Task führt zu einem Task-Wechsel.

Welche Variante für einen Pubset verwendet werden soll, bzw. ob SCA beim Importieren des Pubsets automatisch gestartet werden soll (Standard für BS2000/OSD ab V8.0), kann im MASTER-Katalog MRSCAT hinterlegt werden (Operand `START-SPEEDCAT` in `/ADD-` und `/MODIFY-MASTER-CATALOG-ENTRY`).

Bei Einsatz der Variante „SCA ohne Task-Wechsel“ ist ein CPU-Minderbedarf von ca. 1 - 2 % bei gleichzeitig höherem Klasse-4-Speicherbedarf gegenüber der Variante „SCA mit Task-Wechsel“ zu erwarten.

Automatischer Start von SCA

Wenn SCA installiert ist, dann wird seit BS2000/OSD V8.0 beim Importieren eines Pubsets standardmäßig SCA im Modus `*SPEEDCAT-TASK` gestartet, um einen performanten Katalogzugriff bei SF-Pubsets zu gewährleisten.

Im Kommando `ADD-MASTER-CATALOG-ENTRY` wird deshalb seit BS2000/OSD V8.0 der neue Standardwert `START-SPEEDCAT=*AUTOMATIC` verwendet.

Vorteile durch den Einsatz von SCA

1. Erhöhung des Gesamtdurchsatzes des Systems bei katalogintensiver Last

Die verminderte Anzahl physikalischer Zugriffe zur Datei TSOSCAT pro logischem Katalogaufruf (OPEN, CLOSE, ...) führt zu einer deutlichen Verkürzung der Katalogzugriffszeit bei gleichzeitiger Verringerung der Systembelastung.

Weiterhin werden die Plattenlaufwerke entlastet, auf denen sich die Katalogdateien befinden. Bei SF-Pubsets sind das i.d.R. die Volumes xxxx.0. Für das Gesamtsystem bedeutet dies, dass der Durchsatz der zugreifenden Tasks erhöht werden kann. Die Laufzeit für katalogintensive Programme (z.B. ARCHIVE) sowie das Antwortzeitverhalten bei Kommandos mit hohem Anteil an Katalogzugriffen (z.B. /CREATE-FILE, /MODIFY-FILE-ATTRIBUTES, /IMPORT-FILE) kann erheblich verbessert werden.

Bei schnellen Servern und Dialogbetrieb ist SCA unabdingbar. Der Einsatz führt zu einer deutlichen Durchsatzsteigerung.

2. Höhere Flexibilität der RZ-Organisation

Eine der bisherigen organisatorischen Maßnahmen zur Verbesserung der Katalog-Performance kann mit Einsatz von SCA entfallen. Nicht mehr erforderlich ist dann, häufig benutzte Katalogeinträge an den Anfang einer Katalog-Blockkette zu legen.

3. Volle Kompatibilität

SCA verändert die Katalogstruktur nicht. Damit bleiben die Anwenderschnittstellen wie auch die internen Systemschnittstellen zum Katalogverwaltungssystem erhalten.



Wesentliche Performance-Verbesserungen sind nur zu erwarten, wenn große Katalogbelegungen der Anwender vorliegen (größer 60-100 Datei-Einträge pro Benutzererkennung).

SCA kann für verschiedene Kataloge gleichzeitig gestartet und beendet werden, und zwar entweder automatisch beim Importieren des Pubsets (abhängig vom MRSCAT-Eintrag) oder per Prozeduraufruf (/ENTER-JOB FROM-FILE=SPEEDCAT.ENTER.START) nach dem Importieren des Pubsets.

7.2.3 Einsatz von SF-Pubsets

Für den Ablauf von Anwendungen mit bestimmten Ein-/Ausgabe-Anforderungen ist die Zusammenfassung gleichartiger Public Volumes zu Pubsets grundsätzlich sinnvoll.

Ein Pubset soll 4 oder weniger Public Volumes umfassen. Dies hat den Vorteil, dass bei einem Ausfall nur der betreffende Pubset blockiert wird und mit den restlichen Pubsets noch ein eingeschränkter Betrieb weitergeführt werden kann. Das Einrichten eines sogenannten Stand-by-Pubsets (= Kopie des Home-Pubsets) für den raschen Wiederanlauf ist zu empfehlen (Handhabung siehe Handbuch „Einführung in die Systembetreuung“ [10]).

Durch den Einsatz von mehreren SF-Pubsets wird die Vielfalt der Verteilungsmöglichkeiten von Dateien im Sinne eines günstigen Performance-Verhaltens erhöht.

Hinweise zur Verteilung der Systemdateien TSOSCAT, Seitenwechselbereiche und SYSEAM werden im [Abschnitt „Einrichten von Systemdateien“ auf Seite 156](#) gegeben.

Um die Ein-/Ausgabe-Anforderungen der Anwender, die unter verschiedenen Benutzerkennungen arbeiten, durch Systemvorgaben zu verteilen, sind bei /ADD-USER bzw. /MODIFY-USER-ATTRIBUTES folgende Parameter zu beachten:

```
/ADD-USER USER-IDENTIFICATION=xx, DEFAULT-PUBSET = *HOME / <catid>      (1)
```

```
PUBSET = *HOME / <catid>      (2)
```

```
PUBLIC-SPACE-LIMIT = <max>/<integer> (3)
```

- (1) <catid> kennzeichnet denjenigen Pubset, auf dem der Anwender Dateien ohne Angabe einer catid ansprechen kann, z.B. in /CREATE-FILE, /ADD-FILE-LINK, /MODIFY-FILE-ATTRIBUTES.
- (2) <catid> gilt für den Pubset, auf den der Anwender zugreifen darf. Der Zugriff zu Dateien eines bestimmten Pubsets ist für einen Anwender nur möglich, sofern er im Benutzerkatalog des jeweiligen Pubsets eingetragen ist.
- (3) <max> gibt das Pubspace-Limit für diese Benutzerkennung des unter (2) angegebenen Pubsets an.

Ein Sonderfall ist, wenn für eine Benutzerkennung der Pubspace-Wert Null für einen Pubset vergeben wird: Der Benutzer hat Zugriff auf alle Dateien, deren Zugriffsrechte entsprechend gesetzt sind; er selbst kann jedoch keine Dateien in diesem Pubset einrichten.

7.2.4 Einsatz von SM-Pubsets

Während die Formatierungseigenschaften eines Volume-Sets, die bei der Initialisierung festgelegt werden, über die gesamte Dauer des Bestehens unveränderbar sind, können die Eigenschaften Verfügbarkeit und Performance im laufenden Betrieb geändert werden. Im Folgenden sollen nur die Performance-Eigenschaften weiter betrachtet werden.

SM-Pubsets aus der Sicht des Anwenders

Mit `/CREATE-FILE` und `/MODIFY-FILE-ATTRIBUTES` (Operand `STORAGE-CLASS`) bzw. dem Makro `FILE` (Operand `STOCLAS`) können für eine Datei so genannte Storage-Services angefordert werden:

Operand	Bedeutung
<code>STORAGE-CLASS</code>	Dateiattribute, die für den Ablageort relevant sind
= <code>*STD</code>	Zuordnung der Default-Storage-Klasse aus dem Benutzerkatalog des SM-Pubsets
= <code><name></code>	Zuordnung einer bestimmten Storage-Klasse
= <code>*NONE</code>	keine explizite Zuordnung einer Storage-Klasse. Der Anwender legt die Performance-Attribute (<code>*STD</code> / <code>*HIGH</code> / <code>*VERY-HIGH</code> / <code>*USER-MAX</code>) selbst fest.

Informationen über Namen und Eigenschaften der verfügbaren Storage-Klassen (d.h. bei Schutz durch `GUARDS` muss der Zugriff erlaubt sein) erhält der Anwender über `/SHOW-STORAGE-CLASS`.

Mit `/SHOW-PUBSET-FILE-SERVICES` kann sich der Anwender über die Storage-Services der vorhandenen Volume-Sets informieren. Er kann somit kontrollieren, ob es einen Volume-Set mit den für seine Datei gewünschten Eigenschaften gibt.

Die Storage-Services können auch über Direktattributierung angefordert werden.

SM-Pubsets aus der Sicht der Systembetreuung

Damit die Performance-Anforderungen des Anwenders an die Volume-Sets eines SM-Pubsets wirksam werden können, müssen entsprechende Berechtigungen und Speicherplatz-Kontingente in den Benutzerkatalog eingetragen werden (`/ADD-USER`, `/MODIFY-USER-ATTRIBUTES` bzw. `/MODIFY-USER-PUBSET-ATTRIBUTES`):

Operand	Bedeutung
<code>DEF-STORAGE-CLASS = <name></code>	Default-Storage-Klasse für die Anwender-Angabe <code>STORAGE-CLASS=*STD</code>

Operand	Bedeutung
DMS-TUNING-RESOURCES = *NONE = *CONCURRENT-USE = *EXCLUSIVE-USE	für die explizite Anforderung über Direktoperand oder implizit über eine Storage-Klasse: kein Caching zugelassen Anforderung PERFORMANCE=*HIGH Anforderung PERFORMANCE=*VERY-HIGH
PERM-/TEMP-/WORK-SPACE-LIMIT = *PARAMETERS(...) HIGH-PERF-SPACE= ... VERY-HIGH-PERF-SPACE= ...	Speicherplatz-Kontingente der Anwender festlegen (jeweils für permanente, temporäre und Arbeitsdateien): für hochperformanten Speicherplatz für sehr hochperformanten Speicherplatz

Mit /CREATE- bzw. /MODIFY-STORAGE-CLASS richtet die Systembetreuung Storage-Klassen ein bzw. ändert sie.

Indem der Anwender einer Datei eine Storage-Klasse zuordnet, erhält diese Datei implizit alle in der Storage-Klasse festgelegten Performance-Attribute.

Operand	Bedeutung
PERFORMANCE = *STD = *HIGH = *VERY-HIGH	bestimmt das Caching-Verhalten (Hauptspeicher, GS): keine Cache-Nutzung Caching mit Verdrängung nach LRU Caching ohne Verdrängung
USAGE = *READ-WRITE = *READ = *WRITE	erhöhte Performance-Anforderungen gelten: für Lese- und Schreiboperationen nur für Leseoperationen nur für Schreiboperationen
DISK-WRITE = *IMMEDIATE = *BY-CLOSE = *STD	legt den Zeitpunkt der Datenkonsistenz fest: Datenkonsistenz nach jeder Schreiboperation (Voraussetzung: nichtflüchtiges Cache-Medium) Datenkonsistenz erst nach CLOSE-Verarbeitung für permanente Dateien gilt *IMMEDIATE, für temporäre Dateien gilt *BY-CLOSE
VOLUME-SET-LIST = <name> = *NONE	bestimmt die Zuordnung einer Volume-Set-Liste Name der Liste keine Zuordnung

Zur Erfüllung der erhöhten Performance-Anforderungen (*HIGH / *VERY-HIGH) muss von der Systembetreuung die Zuweisung des Cache-Mediums Haupt- oder Globalspeicher erfolgen (/MODIFY-PUBSET-CACHE-ATTRIBUTES; siehe [Abschnitt „User-PFA-Caching“ auf Seite 184](#)).

Durch die Angabe einer Volume-Set-Liste pro Storage-Klasse lassen sich RZ-spezifische Strategien zur Steuerung der Ein-/Ausgabe-Verteilung realisieren. Ordnet ein Anwender einer Datei eine Storage-Klasse mit angeschlossener Volume-Set-Liste zu, wird die Datei bevorzugt auf einem zur Liste gehörigen Volume-Set angelegt.

Die Definition einer Volume-Set-Liste erfolgt mit /CREATE-VOLUME-SET-LIST.

Die Performance-Eigenschaften eines Volume-Sets werden mit /MODIFY-PUBSET-DEFINITION-FILE festgelegt:

Operand	Bedeutung
PERFORMANCE = *STD = *HIGH = *VERY-HIGH = list-poss(3) ...	bestimmt das Caching-Verhalten (Hauptspeicher, GS): keine Cache-Nutzung Caching mit Verdrängung nach LRU Caching ohne Verdrängung Werteliste ermöglicht einen Performance-Bereich
WRITE-CONSISTENCY = *IMMEDIATE = *BY-CLOSE	legt den Zeitpunkt der Datenkonsistenz fest: Datenkonsistenz nach jeder Schreiboperation (Voraussetzung: nichtflüchtiges Cache-Medium) Datenkonsistenz erst nach CLOSE-Verarbeitung

Die ausreichende Beschreibung des Performance-Profiles ist Voraussetzung für das Funktionalisieren der Volume-Set-Selektion. Das Performance-Profil wird nicht automatisch aus der physikalisch vorhandenen Konfiguration ermittelt. Es ist die Aufgabe der Systembetreuung, die realen Verhältnisse (Berücksichtigung eines flüchtigen oder nichtflüchtigen Cache-Mediums bei der Vergabe von WRITE-CONSISTENCY) richtig zu beschreiben.

7.2.5 Einrichten von Systemdateien

Systemdateien werden vom Betriebssystem zur Abwicklung seiner Aufgaben benötigt. Wegen dieser exponierten Stellung zählen sie, was die Leistungsfähigkeit eines IT-Systems betrifft, zu den sensitivsten Dateien. Aus der Vielzahl der Systemdateien wird auf die folgenden am häufigsten zugegriffen:

Funktion	DVS-Name
Dateikatalog (SF-Pubset)	\$TSOS.TSOSCAT
Dateikatalog (SM-Pubset)	\$TSOS.TSOSCAT.<volset-id>
Seitenwechselbereiche	\$TSOS.SYS.PAGING.<vsn>
Hintergrundspeicher für die Zugriffsmethode EAM	\$TSOS.SYSEAM

Bei der Einrichtung dieser Dateien ist das Hauptaugenmerk auf die **Vermeidung von Zugriffskonflikten** zu legen.

Primär lässt sich die Wahrscheinlichkeit von Zugriffskonflikten durch die Aufteilung der Dateien auf mehrere Volumes verringern. Befinden sich aus wirtschaftlichen Gründen (z.B. Nutzung der Plattenkapazität) Anwender- und Systemdateien auf demselben Volume, so ist die Zusammenlegung mit schwach frequentierten Anwenderdateien anzustreben.

Als charakteristische Merkmale der Systemdateien sind die **Dateigröße** und die **Zugriffshäufigkeit** anzusehen.



Wenn ein System eingerichtet ist, kann die Lage der Seitenwechselbereiche nur mit hohem Aufwand verändert werden. Deshalb muss bereits beim Konfigurieren besonderes Augenmerk auf Lage und Größe dieser Systemdatei gerichtet werden.

Bei Programmentwicklungen ist die Häufigkeit der Zugriffe auf katalogisierte Modulbibliotheken (TASKLIBs) und Makrobibliotheken oft so groß, dass beim Konfigurieren deren Lage berücksichtigt werden muss.

7.2.5.1 Dateikatalog (SF-Pubset)

Auf den Dateikatalog (die Datei TSOSCAT) wird nur von den Routinen des CMS (Catalog Management System) zugegriffen. Die PAM-Blöcke dieser Datei enthalten entweder:

- Dateieinträge
- Jobvariableneinträge
- globale Verwaltungsdaten
(MRS-Katalog und Einträge zur Beschreibung des aktuellen Zustands des TSOSCAT)

Die Zugriffe auf die Datei erfolgen blockweise. Durch den Systemparameter BMTNUM (siehe [Seite 337](#)) oder `/IMPORT-PUBSET` bzw. `/ADD-` und `/MODIFY-MASTER-CATALOG-ENTRY` wird festgelegt, wie viele Ein-/Ausgabe-Puffer das CMS zur Verfügung hat. Simultananforderungen der Anwender auf denselben Katalogblock werden mit Hilfe einer „Buffer Management Table“ (BMT) und einer Börse synchronisiert (je eine BMT und eine Börse pro Katalog-Ein-/Ausgabe-Puffer).

Die Katalogdatei kann in mehrere Bereiche gegliedert sein. Aus technischen Gründen muss der erste Bereich auf dem Volume xxxx.0 liegen. Sind mehrere Pubsets vorhanden, gibt es auf jedem Pubset einen Katalog. Vermieden werden muss, dass Teile von TSOSCAT auf stark belasteten Plattenlaufwerken liegen. Als solche gelten u.a. alle Volumes mit Seitenwechselbereichen und SYSEAM-Bereichen.

Dateikatalogformate

Es gibt drei Dateikatalogformate:

NORMAL und LARGE (implizit, für Pubsets mit „großen“ Objekten) sowie EXTRA LARGE.

TSOSCAT-Typ	max. Größe (Katalogblöcke)	max. Größe (PAM-Seiten)	kompatibel zu
NORMAL	8.192	16.384	BS2000/OSD-BC V1.0
LARGE	16.384	32.368	BS2000/OSD-BC V5.0
EXTRA LARGE	32.368	64.016	BS2000/OSD-BC V6.0B

Die Katalogformate LARGE und EXTRA LARGE sind inkompatibel:

Werden sie für einen Pubset genutzt, so ist mit diesem Pubset ein Rückstieg in eine ältere BS2000-Version nicht mehr möglich.

Für SM-Pubsets mit Katalogen des Typs EXTRA LARGE können für jeden der Spezialkataloge #MIN, #JVC und #PTV bis zu 100 Teilkataloge mit jeweils 64.016 PAM-Seiten angelegt werden (`/ADD-CATALOG-FILE`).

Das CMS erkennt, wenn ein Dateikatalog zu 90% belegt ist, und nimmt automatisch eine Katalogvergrößerung vor, sofern das ohne Wechsel des Katalogformats möglich ist. Für Spezialkataloge des Typs EXTRA LARGE wird dann ein neuer Teilkatalog angelegt, wenn keiner der bereits existierenden Teilkataloge vergrößert werden kann.

Außerdem kann die Datei manuell mit dem Kommando `/MODIFY-FILE-ATTRIBUTES` (Operand `SPACE`) vergrößert werden. Die Veränderung wird sofort wirksam und nicht erst nach dem nächsten Import des Pubsets.

Einrichten des TSOSCAT

Die Größe des TSOSCAT wird bestimmt durch

- die Anzahl der zugelassenen Anwender (repräsentiert durch die mit `/ADD-USER` eingetragenen Benutzerkennungen)
- die Anzahl der Dateien und Jobvariablen im System
- die Größe der Katalogeinträge, die bei Dateien von der Anzahl ihrer Extents, bei Jobvariablen von der Größe des JV-Werts abhängt
- das Zersplittern der Katalogeinträge, das durch selektives Löschen entsteht
Der Platz, der durch Löschen „einzelner“ Einträge frei wird, steht zwar zum Anlegen neuer Einträge zur Verfügung, bleibt aber sonst ungenutzt. Erst nach dem Löschen aller Einträge wird ein Katalogblock freigegeben.

Die Einträge im TSOSCAT sind in Katalogblöcken (zu je 4 KB) abgelegt:

- Ein Katalogblock für Dateien, der so genannte „Primary Block“, wird für jeden Benutzer auch dann reserviert, wenn er keine Dateien anlegt.
- Ein Katalogblock enthält nur Dateieinträge eines Benutzers oder nur JV-Einträge eines Benutzers.
- In einen Katalogblock passen 1 bis 13 Dateieinträge.
Auf einem ausreichend großen und mit `SPACEOPT` gepflegten Pubset kann im Durchschnitt mit 10 Dateieinträgen pro Katalogblock gerechnet werden.
- In einen Katalogblock passen 8 bis 18 JV-Einträge.
Ein Durchschnitt lässt sich nur schwer angeben, weil die Größe der JV-Werte durch den Benutzer frei gewählt werden kann. Beispielsweise passen bei `MONJVs`, die eine Standardgröße von 128 Byte haben, 11 JV-Einträge in einen Katalogblock.

Die durchschnittliche Anzahl der durch „Primary Blocks“ und Dateieinträge belegten Katalogblöcke lässt sich durch folgende Formel abschätzen:

Anzahl Katalogblöcke = Anzahl Benutzerkennungen + (Gesamtanzahl Dateien / 10)

Die Formel berücksichtigt die Zersplitterung der Katalogeinträge und die sinkende Kapazität bei steigender Größe der Einträge.

Für Jobvariablen werden zusätzliche Katalogblöcke benötigt.

Beispiel

Berechnung der Kataloggröße für 150 Benutzerkennungen mit 65 Dateien pro Benutzerkennung: $TSOSCAT = 150 + (150 * 65 / 10) = 1125$ Katalogblöcke

Ohne das Softwareprodukt SCA (Speed Catalog Access) sollten pro Benutzerkennung nicht mehr als 65 Dateien bzw. Jobvariablen katalogisiert werden. Mit SCA kann die Anzahl der Dateien bzw. Jobvariablen höher sein.

Lage des TSOSCAT

Beim Einrichten des Systems wird standardmäßig auf dem Volume xxxx.0 ein TSOSCAT in der Größe von 2000 PAM-Blöcken eingerichtet.

Unter der Voraussetzung, dass der auf Systemresidenz (xxxx.0) eingerichtete Seitenwechselfeldbereich **im späteren Betrieb nicht benutzt wird**, sollte der TSOSCAT in seiner erforderlichen Größe auf dem Volume xxxx.0 angelegt werden (bei Installation mit SIR).

Wird das Volume xxxx.0 neben den Anwenderaktivitäten (max. 15% Auslastung) von der Systemseite her im Wesentlichen nur durch Katalogzugriffe belastet (ebenfalls max. 15% Auslastung), so schafft es bei 70% Read-Hit-Rate ca. 50 Katalog-Ein-/Ausgaben pro Sekunde. Eine Reduzierung durch die Erhöhung der Anzahl CMS-Ein-/Ausgabe-Puffer und dem Einsatz von SCA ist anzustreben.

Bei Einsatz mehrerer Pubsets befindet sich der TSOSCAT automatisch auf jedem Pubset. Die Empfehlungen gelten dann sinngemäß für das Volume xxxx.0 jedes Pubsets.

Nach Aufnahme des Echtbetriebs wird die Überprüfung der Anzahl Katalog-Ein-/Ausgaben pro Sekunde mit SM2 empfohlen (Reports „CMS Queues“, „CMS IO'S“ der Reportgruppe CATALOG-MANAGEMENT).

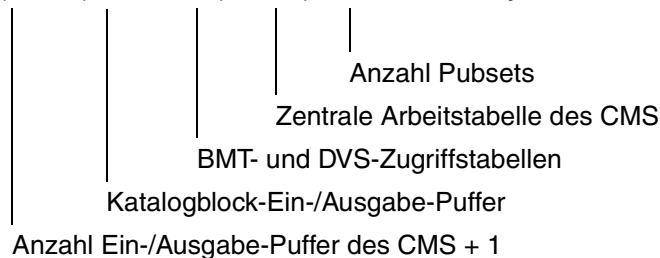
Für den Produktivbetrieb sind bei der Kalkulation der Leistung des CMS neben der Katalog-Ein-/Ausgabe-Intensität auch der Adressraumbedarf für Tabellen und Puffer sowie Behinderungen als Folge von Synchronisation zu beachten. Beim Einsatz von mehreren Pubsets werden Katalog-Ein-/Ausgabe-Puffer und deren Verwaltungstabellen (BMT) pro Pubset angelegt. Bei vielen Pubsets kann deshalb der Adressraumbedarf des CMS erheblich ansteigen.

Mit /ADD-, /MODIFY-MASTER-CATALOG-ENTRY und /IMPORT-PUBSET ist die pubset-spezifische Festlegung der Anzahl CMS-Puffer möglich.

Beispiel

Im Startup-Parameterservice wurden die Systemparameter BMTNUM=32 und CATBUFR=N angegeben. Für 2 Pubsets benötigt CMS damit als Arbeitsbereich:

$(33 * (4096 + 178) + 388) * 2 = 282.860$ Bytes, d.h. 277 KB Klasse-4-Speicher



Für Pubsets mit EXTRA LARGE Katalogen wird ein zusätzlicher Ein-/Ausgabepuffer für den statischem MRS-Katalog (ca. 4 KB) benötigt.

Zum Schutz simultaner Anforderungen wird pro Katalogpuffer eine Börse angelegt. Zum Schutz gegen konkurrierende Aufrufe der CMS-Routinen werden darüber hinaus weitere Börsen benutzt.

Um die Pfadlängen beim Suchen von Datei- und Jobvariablen-Einträgen kurz zu halten und bei diesem Suchen möglichst wenige Katalogpuffer gegen Simultanaufrufe anderer Anwender zu sperren, muss von Zeit zu Zeit die Belegung der Platten mit Dateien und der Katalogblöcke mit Datei- und JV-Einträgen reorganisiert werden. Dabei müssen alle Dateien und Jobvariablen gesichert, gelöscht und wieder eingerichtet werden. Beim Wiedereinrichten sind zwei eventuell gegenläufige Forderungen zu optimieren:

1. Zum einen sollten große Dateien zuerst wieder ins System übernommen werden. Denn dann kann deren Verteilung nach Gesichtspunkten der Zweckmäßigkeit durchgeführt werden. Kleine Dateien können auf Lückenplätzen untergebracht werden.
2. Zum anderen wird die Leistung des CMS effizient genutzt, wenn Dateieinträge von hochfrequentierten Dateien möglichst am Anfang der Kette von Katalogblöcken jeder Benutzerkennung liegen. Das CMS beginnt die Suche nach Datei- oder JV-Einträgen jeweils im ersten Block der Anwender-Katalogblock-Kette.

Der zweiten Forderung ist der Vorrang zu geben.

Gibt es zu viele Benutzerkennungen und Datei- und JV-Einträge pro Benutzerkennung, sollte das Softwareprodukt SCA eingesetzt werden (Details siehe [Abschnitt „Beschleunigung der Katalogzugriffe mit SCA“ auf Seite 150](#)).

7.2.5.2 Dateikatalog (SM-Pubset)

Ein SM-Pubset besteht aus einem Control-Volume-Set und mehreren „normalen“ Volume-Sets.

Der Dateikatalog eines SM-Pubsets ist so auf die einzelnen Volume-Sets verteilt, dass jeder Teilkatalog die Informationen für die Dateien auf dem entsprechenden Volume-Set enthält.

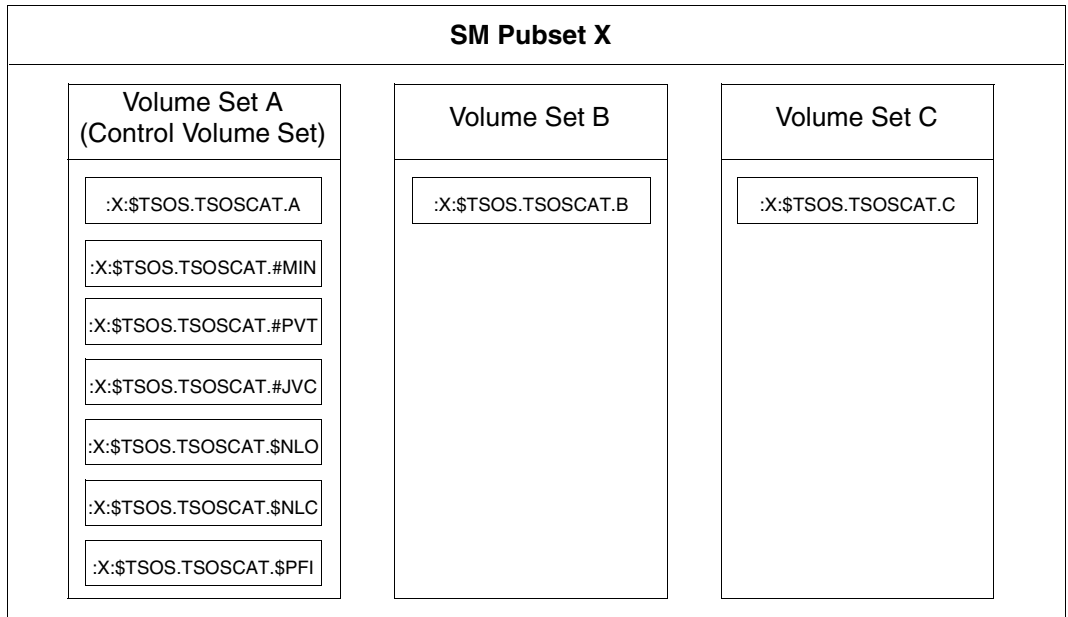


Bild 5: Aufbau des Dateikatalogs eines SM-Pubsets

Bei einer Zusammenlegung mehrerer SF-Pubsets zu einem SM-Pubset ist darauf zu achten, dass im MRSCAT die Werte für die CMS-Puffer angepasst werden.

Bei einfacher Addition der Werte wird sich i.A. durch den besseren Ausgleich von Lastschwankungen am Katalog eine bessere Katalog-Performance ergeben.

Folgende weitere globale Teilkataloge liegen auf dem Control-Volume-Set und sollten dort auf mehrere Volumes aufgeteilt werden:

- Teilkatalog für migrierte und leere Dateien
- Teilkatalog für Dateien auf privaten Datenträgern
- Teilkatalog für Jobvariablen

Weitere Teilkataloge auf dem Control-Volume-Set (\$NLO, \$NLC, \$PFI) dienen ausschließlich der Datensicherheit.

Der zentrale Benutzerkatalog liegt ebenfalls auf dem Control-Volume-Set.

Die einzelnen Teilkataloge entsprechen im Aufbau dem TSOSCAT eines SF-Pubsets. Für die Einrichtung dieser Teilkataloge gelten daher die gleichen Empfehlungen wie für die Einrichtung des TSOSCAT auf SF-Pubsets.

Der Einsatz von SCA ist nicht möglich, der beschleunigte Katalogzugriff ist bereits intern realisiert.

7.2.5.3 Seitenwechselbereiche

Seitenwechselbereiche werden als Dateien unter folgendem Namen katalogisiert:

```
:<catid>: $TSOS.SYS.PAGING.<vsn>
```

Bei jedem Systemlauf kann über den Parameterservice eine Auswahl aus der Menge aller existierenden Seitenwechseldateien getroffen werden. Damit ist eine Anpassung an unterschiedliche Lastprofile von Session zu Session möglich.

Alle Pubsets, die für den kommenden Systemlauf mindestens einen Seitenwechselbereich enthalten, müssen bereits vor dem Einleiten des Systems (Startup) online sein. Sie werden vom Startup exklusiv belegt, jedoch nicht implizit importiert (d.h. die Seitenwechselbereiche werden benutzt, auch wenn der Pubset nicht importiert ist). Diese Pubsets können während des Systemlaufs von anderen BS2000-Systemen (im Sinne von „Shared-Pubsets“) nicht verwendet werden.

Seitenwechselbereiche dürfen auch auf SM-Pubsets liegen.

Einrichten der Seitenwechselbereiche

Beim Einrichten eines Pubsets werden auf allen für Seitenwechsel vorgesehenen Volumes dieses Pubsets die Seitenwechsel-Dateien reserviert und für jede ein Eintrag im Dateikatalog dieses Pubsets angelegt. Ein Volume kann nur eine Seitenwechsel-Datei (allerdings mit mehreren Extents) enthalten.

Die Größe des Seitenwechselbereichs hängt von der Anzahl und der virtuellen Programmgröße der Programme ab, die gleichzeitig ablaufen. Hinzu kommt der jeweils generierte Systemadressraum.

Für die Erstellung wird folgende Formel empfohlen:

$$\text{Größe} = (\text{Anzahl Anwender} * \text{virt. Programmgröße} + \text{Systemadressraum}) * 2$$

Nach Aufnahme des Echtbetriebs sollte eine Überprüfung (Report „Paging Area Utilization“ der Reportgruppe MEMORY von SM2) erfolgen, um die tatsächlich benötigte Dateigröße festzustellen.

Für die gleichzeitig ablaufenden Programme sollte die Größe des Seitenwechselbereichs ein Mehrfaches des mittleren Bedarfs, jedoch nicht mehr als das Doppelte des maximalen Bedarfs betragen:

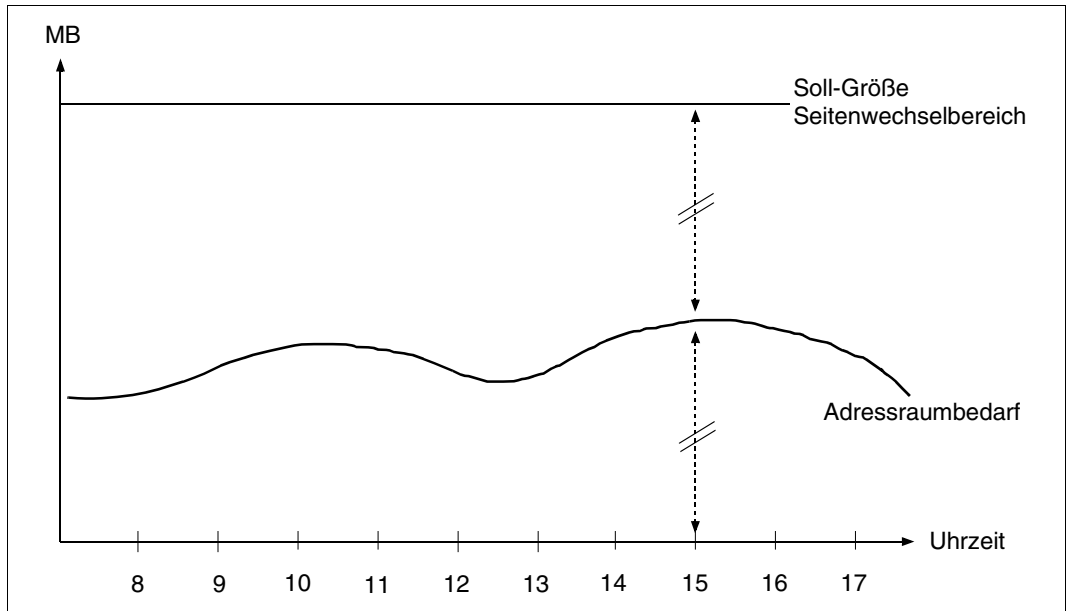


Bild 6: Belegung des Seitenwechselbereichs

Die **Zugriffshäufigkeit** zum Seitenwechselbereich ist abhängig von:

- der Betriebsart (TP-, Dialog- oder Batch-Betrieb)
- der Größe des Hauptspeichers
- der Geschwindigkeit des Servers

Dialogbetrieb

Der Dialogbetrieb ist charakterisiert durch viele, vom Working-Set-Bedarf her aber relativ kleine Tasks (50 - 150 4-KB-Seiten).

Da auf die einzelne User-Task i.d.R. nur wenige Seitenanforderungen pro Dialogschritt entfallen (5 - 10), ergibt sich auch bei einer höheren Paging-Gesamtrate keine wesentliche Beeinträchtigung der Leistung, die der einzelne Anwender sieht.

Die Durchführung des Paging-Vorgangs ist aber nicht nur mit einer zeitlichen Verzögerung, sondern auch mit einem Systemaufwand verbunden.

Aus Effizienzgründen sollte der Aufwand für Paging 3% der Leistungsfähigkeit des Servers nicht überschreiten. Bei leistungsstärkeren Servern ist selbst dieser Prozentsatz zu hoch und erfordert eine unwirtschaftlich hohe Anzahl von Volumes mit Seitenwechselbereichen.

TP-Betrieb

Der TP-Betrieb ist gekennzeichnet durch wenige, vom Working-Set-Bedarf aber relativ große Tasks (mehrere MB groß).

Zur Erfüllung einer gewünschten Anwenderfunktion werden viele Seiten angesprochen, oft über mehrere Tasks hinweg (speziell bei DB/DC-Anwendungen).

Eine höhere Paging-Gesamtrate führt hier meist zu einer schweren Beeinträchtigung der Leistung, die der einzelne Anwender sieht. Wegen der hohen Seitenanforderungen zur Erfüllung einer Funktion steigt die Paging-Rate schon bei geringem Hauptspeicher-Mangel stark an. In diesem Betriebspunkt genügt eine geringfügige weitere Lasterhöhung, um einen Leistungseinbruch hervorzurufen (exponentieller Anstieg der Paging-Rate: das System „kippt“).

Der TP-Betrieb reagiert auf Hauptspeicher-Mangel wesentlich empfindlicher als die anderen Betriebsarten. Daher müssen die Richtwerte für die obere Grenze der Paging-Rate entsprechend tiefer liegen (siehe [Abschnitt „Richtwerte für die BS2000/OSD-Server“ auf Seite 331](#)).

Vom wirtschaftlichen Standpunkt aus ist stets die Senkung der Paging-Rate auf Werte unterhalb der angegebenen Richtwerte anzustreben.



Um auch auf plötzliche Überlasten besser reagieren zu können, sollte ein Volume, welches ausschließlich für Paging-Aktivitäten verwendet wird, nur zu maximal 10% ausgelastet sein. Dabei schafft es bei 70% Read-Hit-Rate ca. 30 Paging-Ein-/Ausgaben pro Sekunde. Da normalerweise nicht mehr als 10 - 20 Paging-Ein-/Ausgaben durchgeführt werden sollen (siehe [Abschnitt „Richtwerte für die BS2000/OSD-Server“ auf Seite 331](#)), ist es i.d.R. nicht nötig die Paging-Area auf mehr als 3 bis 6 physikalische Geräte aufzuteilen.

Lage der Seitenwechselbereiche

Seitenwechseldateien sollten auf allen Volumes die gleiche Größe haben (möglichst nur ein Extent), um eine gleichmäßige Ein-/Ausgabe-Auslastung aller Volumes zu erreichen. In der Regel korreliert die Ein-/Ausgaberate mit der Größe der Seitenwechseldatei.

Nach Möglichkeit sollte kein Seitenwechselbereich auf Volumes aktiviert werden, auf denen sich der Dateikatalog oder ein SYSEAM-Bereich befindet. Dies gilt auch für Volumes, die durch Anwenderdateien hoch ausgelastet sind.

Weitere Einzelheiten sind im Handbuch „Einführung in die Systembetreuung“ [10], Abschnitt „Speicherverwaltung“, und im Handbuch „Systeminstallation“ [31], Abschnitt „Behandlung wichtiger Systemdateien“, beschrieben.

Beispiel zur Berechnung von Größe und Verteilung der Seitenwechselbereiche

Annahme

TP-Betrieb:	1200 Benutzer in über 30 UTM-Tasks und 5 Datenbank-Tasks mit einer jeweiligen durchschnittlichen Programmgröße von 120 MB
Systemadressraum:	240 MB
CPU:	S-Server (Mono, 240 RPF)
Plattenspeichersystem:	Symmetrix DMX-3
Paging-Rate:	maximal 100 Ein-/Ausgaben pro Sekunde Wie auf Seite 164 erläutert, sollte ein Paging-Volume nicht mehr als 30 Paging-Ein-/Ausgaben durchführen, d.h. (3) - 4 Volumes mit Seitenwechselbereichen sind erforderlich

Berechnung

$$(35 * 120 \text{ MB} + 240 \text{ MB}) * 2 = 8880 \text{ MB}$$

Zur Erreichung einer gleichmäßigen Volume-Belastung muss eine Aufteilung in gleich große Teilbereiche erfolgen, d.h. 2220 MB pro Volume.

Auswahl bestimmter Volumes für den Seitenwechsel zum Startup-Zeitpunkt

Ohne Auswahl der Seitenwechselbereiche über den Parameterservice werden die auf dem Home-Pubset eingerichteten Bereiche verwendet.

Durch Angaben in der Systemparameterdatei wird festgelegt, welche Seitenwechselbereiche für den kommenden Systemlauf tatsächlich benutzt werden sollen. Wenn durch die Lastplanung des IT-Systems das Lastprofil bereits vor der Systemeinleitung bekannt ist, können die Paging-Parameter nach diesem Profil mit der Parameterdatei realistisch eingestellt werden.

Grundsätzlich sollte auf dem jeweils ersten Volume eines Pubsets (VSN=xxxx.0) kein Seitenwechselbereich reserviert werden, weil auf diesem immer der pubsetspezifische Dateikatalog liegt und ggf. vom CMS her eine hohe Ein-/Ausgaberate zu erwarten ist.

Erweiterung und Reduzierung des Seitenwechselbereichs

Der Seitenwechselbereich kann sowohl erweitert als auch reduziert werden. Damit ist die Verlagerung des Seitenwechselbereiches von dem performance-kritischen Home-Pubset auf einen weniger belasteten Pubset möglich. Weitere Informationen können dem Abschnitt „Seitenwechselbereich (Paging-Area)“ im Handbuch „Einführung in die Systembetreuung“ [10] entnommen werden.

Zur Erweiterung des Seitenwechselbereiches bzw. Reduzierung der Paging-Aktivitäten auf ausgewählten Volumes siehe Kommandobeschreibungen zu `/EXTEND-PAGING-AREA` und `/MODIFY-PAGING-AREA-ATTRIBUTES` (Handbuch „Kommandos“ [15]).

7.2.5.4 SYSEAM-Datei

Die Datei SYSEAM dient der Speicherung von temporären Daten (Zugriffsmethode EAM). Sie kann auf jedem Pubset eingerichtet werden.

Benutzer greifen auf die SYSEAM-Datei zu, die auf ihrem Benutzer-Default-Pubset liegt. Ist auf dem Benutzer-Default-Pubset keine SYSEAM-Datei vorhanden, so erfolgen die Zugriffe auf die SYSEAM-Datei des Home-Pubsets.

Die optimale Größe von SYSEAM kann nur nach Beobachtung der mittleren Belegung während eines ausreichend langen Zeitraums ermittelt werden (Hinweise zur Ersterstellung siehe im Anhang „[Performance-relevante Systemparameter](#)“, Parameter EAMMEM, EAMMIN, EAMSEC, EAMSIZ, ab [Seite 344](#)).

Die Beobachtung wird mit dem Programm SPCNTRL (Space Control) durchgeführt. An die Datei SYSEAM werden – besonders bei interaktiver Programmentwicklung im Dialogbetrieb – hohe Anforderungen gestellt. Häufige Übersetzungsläufe beeinflussen sowohl die Größe als auch die Zugriffshäufigkeit beträchtlich.

Für die Überwachung der Zugriffshäufigkeit wird der Einsatz von SM2 empfohlen (Dateiüberwachung).

Lage der Datei SYSEAM

Damit bei einer größeren Anzahl von Anwendern, zusammen mit leistungsstarken Servern, die Datei SYSEAM nicht zum Engpass wird, ist die Aufteilung auf mehrere Pubsets sinnvoll.

Nach Möglichkeit sollten keine SYSEAM-Bereiche auf Volumes gelegt werden, auf denen sich Seitenwechselbereiche befinden.

Da bei Plattenspeichersystemen sämtliche Ein-/Ausgabe-Anforderungen dem Caching unterliegen (Write-Hit = 100%), ist die Aufteilung der Datei SYSEAM auf mehrere Volumes innerhalb eines Pubsets nicht erforderlich.

Eine starke Beanspruchung der Datei SYSEAM ist meist mit einer entsprechenden Größe der Datei gekoppelt. Es empfiehlt sich in diesen Fällen, die erforderlichen Volumes ausschließlich für SYSEAM zu reservieren und hinsichtlich der Ein-/Ausgaberate bis an die obere Grenze der Auslastung (30%) zu gehen. In diesem Fall schafft ein Volume ca. 300 Ein-/Ausgaben pro Sekunde.

Lastspitzen für SYSEAM

Als Hauptanwender von SYSEAM sind die Compiler, das Softwareprodukt LMS sowie die Binder BINDER und DBL anzusehen.

Der Systembetreuung sollten die Zeiten bekannt sein, zu denen Programmdienste aufgerufen werden. Dann können Entscheidungen getroffen werden, ob zusätzliche Pubsets einzurichten und wann diese zu importieren sind.

Bei Lasten mit großem Anteil an Programmdiensten sollten die TSOS-Makrobibliotheken, die Anwender-Makrobibliotheken und die MODLIB-Dateien nicht auf Volumes liegen, auf denen es auch SYSEAM-Bereiche gibt.

Steht genügend Hauptspeicher zur Verfügung, so ist es sinnvoll, mit Hilfe des Systemparameters EAMMEM (siehe im Anhang „[Performance-relevante Systemparameter](#)“, Seite 344) einen Teil des SYSEAM-Bereiches in den Klasse-4-Speicher zu legen (gilt nur für den Home-Pubset) und damit SYSEAM-Ein-/Ausgaben zu sparen. Übersetzungs- und Binderläufe können damit entsprechend beschleunigt werden.

Maßnahmen nach SYSEAM-Secondary-Allocation

Wenn in einem Systemlauf die Konsolmeldung `DMS0852 (&00): (&01) PAM SEITEN AND (&02) EXTENTS` ausgegeben wird (mit &00 als Name der SYSEAM-Datei), dann überschreitet die Dateigröße den Wert von EAMMIN bzw. den pubsetspezifischen Eintrag für MINIMAL-SIZE im MRSCAT.

Diese Meldung, die bei jeder Erweiterung oder Reduzierung ausgegeben wird, sollte nur bei Lastspitzen auftreten. Tritt diese Meldung häufig auf, so ist dies ein Hinweis, dass der Systemparameter EAMMIN bzw. der Wert für MINIMAL-SIZE im MRSCAT-Eintrag zu klein gewählt wurde. Zur Vermeidung von Secondary-Allocations müssen diese Parameter angepasst und die Datei entsprechend erweitert werden. Die Dateien können nach der Systemeinführung gelöscht und wieder neu eingerichtet werden. Zu diesem Zeitpunkt dürfen keine Benutzer auf die SYSEAM-Dateien zugreifen. Ein erneuter Startup ist danach nicht erforderlich.

Durch einen System-Kaltstart werden die SYSEAM-Dateien auf die Größe des Systemparameters EAMMIN zurückgesetzt.

7.2.5.5 Meldungsdateien

Um das Erzeugen und Ausgeben von Systemmeldungen in BS2000/OSD effizient zu gestalten, sind zwei Gesichtspunkte zu beachten:

- Verwaltung der Meldungen

Alle Meldungen, die bei einem Einsatz des BS2000-Systems gebraucht werden könnten, sind in mehreren ISAM-Dateien abgelegt. Beim Startup werden die für den Systemlauf unbedingt benötigten und die lt. Parameter vorgegebenen Meldungsdateien verfügbar gemacht (Systemparameter `MSGFILEi` und `MSGNOFL`, siehe im Anhang „[Performance-relevante Systemparameter](#)“, Seite 343). Weitere Meldungsdateien können in der MIP-Parameterdatei (`SYSPAR.MIP.<version>`) angegeben werden. Sie werden aber erst nach dem Laden des MIP-Subsystems aktiviert. Im Systemlauf können weitere Meldungsdateien hinzugenommen oder entfernt werden. Siehe Handbuch „Systembetreuung“ [10].

- systeminterne Behandlung der Meldungen

Jede mit dem Dienstprogramm MSGMAKER erzeugte Meldungsdatei enthält den Meldungstext und einige Attribute (zum Suchen des Meldungstextes) sowie die Bedeutung und den Antworttext (für `/HELP-MSG-INFORMATION`).

Beim Bereitstellen wird die Datei eröffnet und der Verwaltungsteil in einen Arbeitsbereich im Klasse-4-Speicher eingelesen.

Wenn Anwender aufgrund ihrer Verarbeitung Anforderungen zur Ausgabe von Systemmeldungen stellen, so werden bei der Behandlung dieser Meldungsausgaben folgende Aktionen durchgeführt:

- Prüfung aufgrund des mitgelieferten Meldungsschlüssels, ob die Meldung bereits im Meldungspuffer steht oder ob sie mit dem ISAM-Zugriff GETKY gelesen werden muss.
- Prüfung aufgrund des Meldungsgewichts (Weight-Code), ob diese Meldung beim Ausgabemedium Bedienstation ausgegeben werden muss oder ob sie zu unterdrücken ist.
- Aktualisierung variabler Textteile und Einleitung der Ausgabe (falls keine Unterdrückung gefordert wird). Dabei ist mindestens ein Task-Wechsel notwendig.

Ausgabemedien können sein:

- die Haupt- und/oder Nebenbedienstation
- und/oder SYSOUT/SYSLST der User-Task
- und/oder ein User-Bereich (Puffer)
- oder eine spezielle BCAM-Applikation (z.B. OMNIS)

Jede angeforderte Meldungsausgabe an die Haupt- und/oder Nebenbedienstation (auch solche, deren Ausgabe zu unterdrücken ist) wird außerdem als Datensatz in die CONSLOG-Datei geschrieben.

Ablage von Meldungsdateien

Meldungsdateien, die nur bei speziellen Anwendungen benötigt werden, sollten nicht auf den Platten des Home-Pubsets stehen (sie belegen dort nur unnötig Platz), sondern auf dem zu diesen Anwendungen gehörenden Pubset. Für den aktuellen Bedarf können die Platten mit `/IMPORT-PUBSET` zugeschaltet oder die Dateien aus Sicherheitsbeständen (z.B. mit HSMS) rekonstruiert werden. Die Meldungsdateien werden dann mit `/MODIFY-MSG-FILE-ASSIGNMENT` oder `/MODIFY-MIP-PARAMETERS` in das laufende System übernommen.

Einzelne Meldungen (oder die Inhalte ganzer Meldungsdateien) werden in die für den Systemlauf notwendigen Meldungsdateien integriert.

Anzahl der Meldungsdateien

Die Meldungen spezieller Softwareprodukte können in der Meldungsdatei des Softwareprodukts verbleiben. Sie werden mit IMON automatisch installiert.

Wieviele Meldungsdateien als Bestandteil des Gesamtsystems verwaltet und welche davon in welchen Systemläufen verfügbar gemacht und eventuell nach Ende des Bedarfs entfernt werden, liegt im Ermessen der Systembetreuung.



Wenige zusätzliche Meldungsdateien vereinfachen deren Verwaltung und benötigen weniger Systemadressraum. Bei vielen Meldungsdateien ist eine genauere Anpassung an den aktuellen Lastfall möglich.

Vorschläge zur Verringerung bzw. Beschleunigung der Meldungs Ausgabe

1. Unterdrücken bestimmter Systemmeldungen

Das Bereitstellen, Aktualisieren und Ausgeben von Meldungen ist für den ordnungsgemäßen Systemablauf notwendig, stellt aber auch eine Belastung für das System dar. Die Art der Behandlung der Meldungen soll dem Bedarf angepasst sein. Zum Beispiel kann bereits die Anforderung zur Meldungs Ausgabe abgeschaltet werden, wenn das Laden von Programmen (`/START-EXECUTABLE-PROGRAM`) für die Tätigkeit des Operators ohne Bedeutung ist und auch sonst ein Protokoll dazu nicht ausgewertet wird (Systemparameter EACTETYP auf 0 setzen, siehe [Seite 343](#)).

Werden im IT-Betrieb unterschiedliche Anwendungen zeitlich nacheinander abgewickelt, sollte auch die Menge der zu unterdrückenden Ausgaben auf den Bedienstationen dem Einzelfall angepasst werden.

Dazu gehört:

Alle Meldungen, die hinsichtlich der Ausgabeunterdrückung unterschiedlich zu behandeln sind, werden in ihren Meldungsdateien mit unterschiedlichen Meldungsgewichten versehen. Mit `/ADD-CONSOLE-FILTER` wird für jeden Lastfall die jeweils günstigste Filterstufe eingestellt. Mögliche Lastfälle sind z.B. Produktiveinsatz, Programmentwicklung, Testbetrieb, Diagnose nach Störfällen oder Versionsumstellungen und Mischformen.

Folgende Empfehlungen sollten beachtet werden:

Unnötige Ausgaben auf den Bedienstationen belasten das System und machen das Protokoll unübersichtlich. Ein einmal erstelltes Konzept für die Vergabe und das Setzen der Meldungsgewichte in die einzelnen Meldungssätze macht sich im Laufe der Zeit bezahlt.

Zur Ausgabeunterdrückung anhand des Meldungsgewichtes gibt es eine Alternative durch den Einsatz der Funktion Meldungsbestellung bzw. Meldungsabbestellung (/MODIFY-MSG-SUBSCRIPTION, /SET-MSG-SUPPRESSION). Hiermit ist es einer Meldungsempfängerinstanz möglich, einzelne Meldungen zu bestellen bzw. abzubestellen. Beide Möglichkeiten können auch kombiniert genutzt werden:

- a) Zuerst legt man über die Zuordnung von Berechtigungsschlüsseln und die Festlegung von Meldungsgewichten die Meldungsmenge grob fest.
- b) Dann führt man über den Meldungsbestell-/abbestell-Mechanismus eine Feineinstellung durch.

2. Gezielte Ausgabe von Systemmeldungen

Zur Vergabe von Berechtigungsschlüsseln (Parameterservice, Anweisung SET-CODE) gilt:

- Mit entsprechenden Berechtigungsschlüsseln können Bedienstationen Bedienungsmeldungen empfangen.
- Das Durchreichen von Meldungen wird über einen systeminternen Verständigungsmechanismus (mit Hilfe von Börsen) abgewickelt und verursacht einen nicht unerheblichen Systemaufwand.
- Bei der Vergabe der Berechtigungsschlüssel sollte grundsätzlich geprüft werden, welche Meldungen bzw. Kommandos an den Bedienstationen zur Verfügung stehen müssen, um durch eine restriktive Vergabe überflüssige Meldungen zu vermeiden.
- Durch den Einsatz des Meldungsbestell-/abbestell-Mechanismus kann eine Optimierung der Meldungszustellung erreicht werden.

3. Beschleunigung der Meldungsabgabe mit DLAM (Dynamic Loadable Access Method)

Die Systemverwaltung kann für jede Meldung das DLAM-Kennzeichen setzen, das im Inhaltsverzeichnis jeder Meldungsdatei hinterlegt ist. Beim Startup wird für alle Meldungsdateien, die laut Startparameter bei „System Ready“ zur Verfügung stehen sollen (Systemparameter MSGFILi, siehe [Seite 343](#)), das Inhaltsverzeichnis ausgewertet und in den Bereichszuordnungen (im zentralen Tabellenwerk des Message-Handlers) abgelegt. Danach werden alle Meldungen mit dem DLAM-Kennzeichen aus den jeweiligen Dateien gelesen.

(Dienstprogramm MSGMAKER, Anweisung/Maske: ADD-MSG und MODIFY-MSG)

Vorteil

Während des Systemlaufs ändert sich der Bedarf an Adressraum für gepufferte Meldungstexte nicht (statisches Bild ab Startup). Bei gesetztem DLAM-Kennzeichen verkürzen sich die Pfadlängen, da die Prüfung entfällt, ob ein gerade benötigter Meldungstext bereits gepuffert ist. Die physikalische Input-Operation wird aus dem Systemlauf in den Startup-Abschnitt vorverlegt:

Der Aufbau der Meldungen erfolgt im Klasse-5-Speicher zum Startup-Zeitpunkt. Es wird keine physikalische Ein-/Ausgabe bei Aufruf der Meldung während des Systemlaufs durchgeführt.

Aufwand

Das Selektieren und Kennzeichnen der einzelnen Meldungen kostet Bearbeitungszeit. Auf großen Servern mit hohem Durchsatz ist DLAM zu empfehlen.

4. Beschleunigung der Meldungs Ausgabe mit LOCAL-DLAM (Local Dynamic Loadable Access Method)

Falls eine Meldung mit der Zugriffsmethode LOCAL-DLAM (vergeben mit dem Dienstprogramm MSGMAKER) versehen ist, wird sie bei Aktivierung in den Klasse-4-Speicher kopiert.

Vorteil

Wie beim Zugriff mit DLAM ist keine physikalische Ein-/Ausgabe nötig. Zusätzlich fällt der Zugriff auf die zentrale MIP-Task weg, alle Aktionen laufen unter der Auftraggeber-Task ab. Damit wird nicht nur der Systemaufwand reduziert, sondern es werden zusätzlich mögliche Lock-Situationen zwischen der zentralen MIP-Task und anderen Auftraggeber-Tasks verhindert.

5. Beschleunigung der Meldungs Ausgabe mit Hilfe der Systemparameter MSGCENTL und MSGCENTN (siehe [Seite 343](#))

MSGCENTL legt die max. Länge von Meldungen, MSGCENTN die Anzahl der Meldungen fest, die im Klasse-4-Speicher zwischengespeichert werden, um Dateizugriffe zu sparen.

Die Größe G des benötigten Klasse-4-Speichers errechnet sich mit:

$$G = \text{MSGCENTL} * \text{MSGCENTN} \text{ [Byte]}$$

7.2.6 Einrichten von Anwenderdateien

Anwenderdateien werden i.A. von den Anwendern im Verlauf ihrer Verarbeitung eingerichtet und bearbeitet. Dabei ist zu beachten, dass jedes Einrichten und Löschen einer Datei, jedes Eröffnen und Schließen einer Datei und jedes Verändern der Größe einer Datei den Dateikatalog belastet.

Wesentliche Kriterien für die Wahl des Datenträgers bzw. der logischen Betriebsart sind die Dateigröße, die Zugriffshäufigkeit und die Katalogintensität im Rahmen der Dateibearbeitung.



Seit BS2000/OSD V8.0 beträgt die Ein-/Ausgabegröße beim Kopieren von Dateien mit `/COPY-FILE` 128 KByte (davor: 64 KByte). Beim Kopieren großer Dateien ergeben sich damit Verbesserungen von bis zu 20% beim CPU-Bedarf und bis zu 15% bei der Laufzeit.

Bei kleinen Dateien kann sich die Laufzeit erhöhen. Beim Einsatz von SCA (siehe [Seite 150](#)) verbessert sich die Laufzeit um ca. 5%.

Dateigröße und Zugriffshäufigkeit

Die Zugriffshäufigkeit auf Dateien hat bei der Konzeption der Datenbestände erheblichen Einfluss auf die Leistung des Systems.

- Um die Auslastung der Volumes möglichst gering und gleichmäßig zu halten, sollten zugriffsarme (große) Dateien auf Volumes gelegt werden, auf denen bereits zugriffsintensive Dateien installiert sind und umgekehrt.
- Generell sollten stark frequentierte Dateien auf getrennte Volumes verteilt werden (insbesondere bei schreibintensiven Anwendungen). Dies ist nur sinnvoll bei Anwendungen mit mehreren Tasks.
- Ein- und Ausgabedateien, die innerhalb einer Verarbeitung abwechselnd angesprochen werden, sollten auf getrennten Volumes liegen.
- Auf Private Volumes kann der Anwender die Lage der Dateien durch explizite Vorgaben festlegen. Auf Public Volumes können dies nur die Systemverwaltung bzw. ausgewählte Anwender (`/MODIFY-USER-ATTRIBUTES`, Operand `PHYSICAL-ALLOCATION`) tun. Die Lage großer Dateien sollte nach Möglichkeit durch explizite Vorgaben festgelegt werden.
- Für zugriffsintensive Dateien zeitkritischer Anwendungen ist die Nutzung dieser Dateien als HIPERFILES sinnvoll (siehe [Abschnitt „Arbeiten mit HIPERFILES“ auf Seite 176](#)).
- In der Regel werden Lasten mit normalen bis hohen Schreibraten durch die gängigen Caching-Mittel in den Plattensteuerungen gut bewältigt. Bei Lasten mit sehr hohen Schreibraten auf einzelne Hotspots/Dateien kann es aber zu Staus kommen („Write delay“).

Zur Bewältigung dieses Effektes gibt es folgende Tuning-Möglichkeiten, die auch miteinander kombiniert werden können:

- durch den Einsatz von Hiperfiles auf Globalspeicher lassen sich kürzeste Ein-/Ausgabezeiten mit maximalem Durchsatz realisieren (siehe auch [Seite 176](#))
- durch Verteilung der Hotspots auf mehrere logische Volumes bzw. physikalische Geräte lässt sich die Ein-/Ausgabe-Bandbreite vervielfachen. Dazu gibt es
 - a) Hardware-Maßnahmen:
Einsatz von RAID1/0 oder RAID5 (siehe [Seite 49](#)) mit zusätzlicher Software-Unterstützung durch den Einsatz von PAV (siehe [Seite 55](#)) und
 - b) Software-Maßnahmen:
Verteilung des Hotspots bzw. der Datei auf mehrere Dateien, Extents oder Volumes. Als Beispiel sei hier die Verteilung der KDCFILE von openUTM auf mehrere Dateien erwähnt (siehe [Seite 193](#)).

Katalogintensität

Wie schon im [Abschnitt „Logische Betriebsarten von Platten“](#) (ab [Seite 145](#)) erwähnt, ist der systeminterne Aufwand für Dateibearbeitungsaufrufe mit Katalogzugriff für Public Volumes bzw. Pubsets am geringsten, gefolgt von Private Volumes. Am höchsten ist der Aufwand für Shared-Private-Disk.

Der Durchsatz bei den einzelnen logischen Betriebsarten hängt also ab vom Verhältnis der Verwaltungs-Ein-/Ausgaben zu den Nutz-Ein-/Ausgaben.

Für Dateien mit hoher Katalogintensität (Prozeduren, ENTER-Dateien, viele rasch aufeinanderfolgende OPEN-/CLOSE-Operationen) sind Public Volumes bzw. Pubsets am besten geeignet.

Dateien mit geringer Katalogintensität können auf Private Volumes gespeichert werden.

Beim gemeinsamen Zugriff von mehreren Servern ist hohe Katalogintensität nur bei Shared-Pubsets vertretbar.

Die Beachtung folgender Hinweise führt zur Reduzierung der Anzahl der Verwaltungs-Ein-/Ausgaben:

- Primary- und Secondary-Space-Zuweisungen (Primary-/Secondary-Allocation) bei /CREATE-FILE sollten an die erwartete Dateigröße angepasst werden.
- Unnötiges Löschen und Einrichten einer Datei sollte verhindert werden (Job-Kommandofolgen: /CREATE-FILE, /DELETE-FILE, /CREATE-FILE auf denselben Dateinamen).
- Mit OPEN-/CLOSE-Operationen sollte sparsam umgegangen werden. (Auch die Kommandos /COPY-FILE, /CALL-PROCEDURE, /INCLUDE-PROCEDURE, /ENTER-JOB, /ENTER-PROCEDURE, /START-(EXECUTABLE-)PROGRAM, /LOAD-(EXECUTABLE-)PROGRAM und /ASSIGN-SYSDTA lösen OPEN-/CLOSE-Operationen aus.)

Reorganisation

Bei größeren Systemen empfiehlt sich eine regelmäßige Reorganisation des gesamten Dateibestandes (z.B. alle 3 Wochen). Die Reorganisation wirkt der Zersplitterung von Dateien in viele kleine Extents entgegen. Die Reorganisation wird durch das Softwareprodukt SPACEOPT wesentlich vereinfacht.

SPACEOPT bereinigt die Fragmentierung durch optimale Verlagerung der Datei-Extents auf den Volumens eines Pubsets. Die Reorganisation von Dateien ist auch dann möglich, wenn der verfügbare freie Platz kleiner ist als die Größe der Datei, d.h. SPACEOPT ermöglicht auch lokale Verbesserungen auf Basis des verfügbaren freien Plattenspeichers.

SPACEOPT kann in den Plattenspeichersystemen ETERNUS DX und Symmetrix die Größe von BS2000-Volumen an die Größe der LUNs (Logical Units), durch die sie realisiert werden, anpassen.

Eine derartige Anpassung kann nach einer Plattenmigration mit DRV bei der Quell- und Zielplatte die gleiche Kapazität haben müssen, erforderlich werden. Informationen zur Plattenmigration, z.B. von D3475 nach D3435, finden Sie im Handbuch „DRV“ [7].

Ein weiterer Anwendungsfall ist die Vergrößerung eines BS2000-Volumens im Anschluss an die Vergrößerung einer LUN durch Hinzufügen zusätzlicher Platten im Plattenspeichersystem ETERNUS DX oder Symmetrix. Die Funktion ist nur für das Platten D3435 (FBA-Plattenformat) verfügbar.

SPACEOPT arbeitet volume-bezogen, d.h. Aufträge können für einzelne oder alle Volumens eines Pubsets spezifiziert werden. SPACEOPT bietet Möglichkeiten zur Bewertung des Belegungs- und Fragmentierungszustandes und kann im laufenden Betrieb eingesetzt werden.

SPACEOPT kann eine Verlagerung von Dateien auch dann vornehmen, wenn die Dateien geöffnet sind.

Zweckmäßigerweise erfolgt der Einsatz von SPACEOPT in betriebsärmeren Zeiten, da abhängig vom Fragmentierungsgrad der Volumens eine höhere Ein-/Ausgabe-Last anfallen kann.



Die Performance anderer BS2000-Gastsysteme oder anderer Server, die Ein-/Ausgaben auf die gleiche physikalische Platte ausführen, kann bei der Reorganisation mit SPACEOPT negativ beeinflusst werden.

Siehe auch [Abschnitt „Konfigurierung in BS2000/OSD“ auf Seite 52](#).

Ausführliche Informationen zu SPACEOPT finden Sie im Handbuch „SPACEOPT“ [30].

7.3 Arbeiten mit HIPERFILEs

Ziel des HIPERFILE-Konzepts (High Performant Files) ist die Beschleunigung von Datei-zugriffen und die Vermeidung bzw. Beseitigung von Ein-/Ausgabe-Engpässen durch Zwischenpufferung der Daten im Haupt- oder Globalspeicher mit dem Softwareprodukt DAB (Disk Access Buffer).

DAB (Disk Access Buffer) hat das Ziel, die Zugriffe auf Plattendateien durch Caching in den schnellen Speichermedien Hauptspeicher oder Globalspeicher zu beschleunigen.

Der Hauptspeicher ist ein flüchtiges Speichermedium, das sich in erster Linie für das Caching von Lesezugriffen sowie zusätzlich für Schreibaufträge auf temporäre Dateien eignet. Nicht gesicherte Schreibdaten im Zwischenpuffer sind bei einem flüchtigen Speichermedium nach einem Systemabbruch verloren.

Der Globalspeicher (an S-Servern verfügbar) ist ebenfalls ein flüchtiges Speichermedium, das aber durch optionale Erweiterungen (Batterie bzw. unterbrechungsfreie Stromversorgung) zu einem nichtflüchtigen Speichermedium aufgerüstet werden kann. Der Globalspeicher eignet sich daher besonders für ausfallsicheres, schnelles Schreiben.

Mit ADM-PFA-Caching und User-PFA-Caching gibt es zwei Ausprägungen des HIPERFILE-Konzepts:

- **ADM-PFA-Caching (Administrator controlled Performant File Access)**
bezeichnet eine Variante des HIPERFILE-Konzepts, bei der die Systembetreuung über Kommandos festlegt, welche Dateien bzw. Volumes in welchem Cache-Medium gepuffert werden sollen und welcher Caching-Modus (Read-, Write- oder Read-Write-Mode) dabei angewendet wird.
- **User-PFA-Caching (User controlled Performant File Access)**
bezeichnet eine zweite Variante des HIPERFILE-Konzepts, die die Einbettung der HIPERFILEs in das Data-Management (DVS) beinhaltet. Hier kann der Anwender entscheiden, welche seiner Dateien eines Pubsets HIPERFILEs werden und damit von einem, von der Systembetreuung eingerichteten Cache profitieren sollen.

Darüber hinaus können im Globalspeicher Platten emuliert werden, deren Zugriffszeiten gegenüber einem Plattenspeichersystem erheblich kleiner sind. Die Verwaltung dieser Platten-Emulation wird vom Subsystem GSVOL (GS-Volumes im GS) geleistet.

7.3.1 Caching-Modi

Befinden sich die Daten, auf die zugegriffen wird, im Cache, so spricht man von einem Cache-Hit, andernfalls von einem Cache-Miss. Das Verhältnis der Hits zur Gesamtanzahl der Zugriffe wird als Cache-Hit-Rate bezeichnet. Je höher die Hit-Rate ist, desto mehr Nutzen bringt ein Cache. Hier wird der Cache im Haupt-/Globalspeicher betrachtet; der im Plattenspeichersystem integrierte Cache arbeitet „unsichtbar“ im Hintergrund.

Folgende Caching-Modi stehen zur Verfügung (die Lesezugriffe sind in den Bildern jeweils links, die Schreibzugriffe jeweils rechts durch Pfeile gekennzeichnet):

Read-Cache

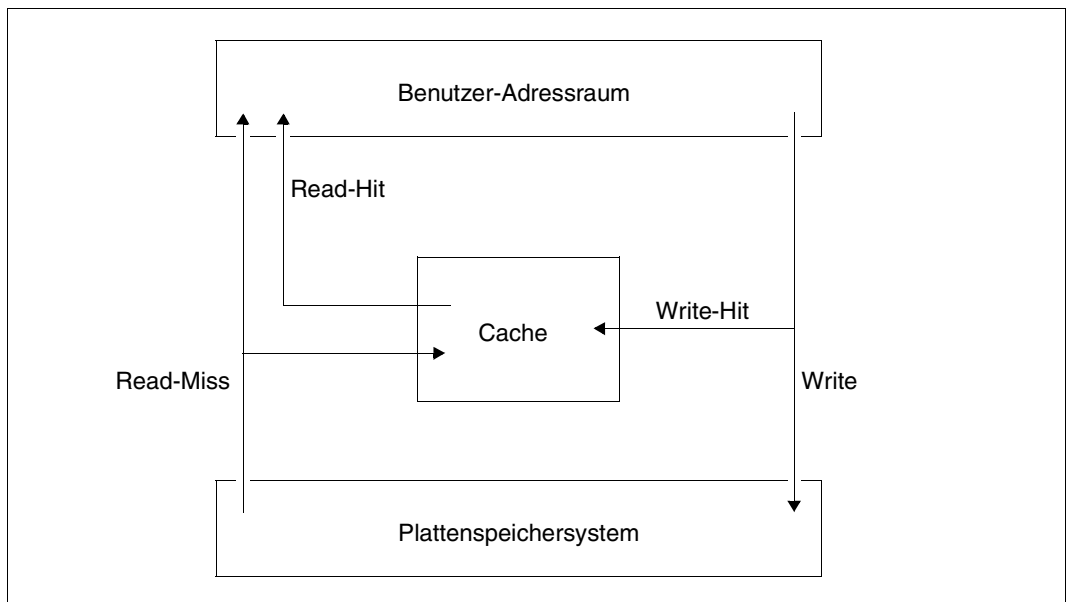


Bild 7: Ein-/Ausgabe-Abläufe bei einem Read-Cache

- Read-Hit:** Die Daten befinden sich zum Zugriffszeitpunkt im Cache und werden von dort gelesen.
- Read-Miss:** Die Daten werden vom Plattenspeichersystem gelesen und gleichzeitig in den Cache eingetragen, damit nachfolgende Lesezugriffe von dort befriedigt werden können.
- Write:** Schreibzugriffe erfolgen grundsätzlich auf das Plattenspeichersystem. Befindet sich der entsprechende Datenblock bereits im Cache („Write-Hit“), wird der Inhalt aktualisiert, ohne dass der Schreibvorgang beschleunigt wird.

Write-Cache

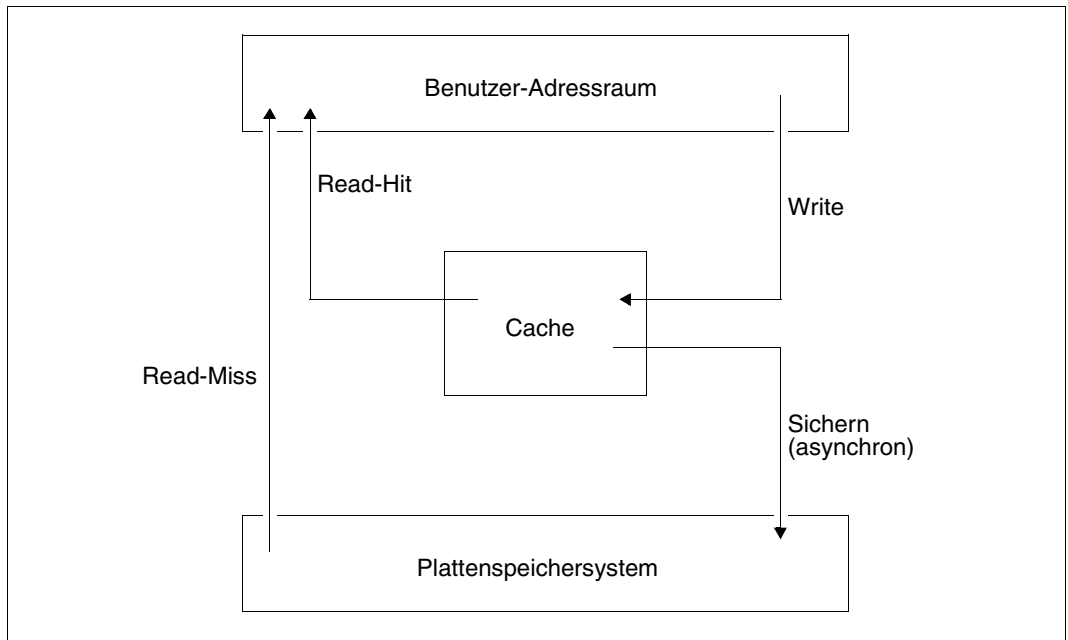


Bild 8: Ein-/Ausgabe-Abläufe bei einem Write-Cache

- Read-Hit:** Der Lesezugriff erfolgt direkt aus dem Cache (wie beim Read-Cache).
- Read-Miss:** Die Daten werden vom Plattenspeichersystem gelesen, jedoch nicht in den Cache eingetragen.
- Write:** Sowohl beim Haupt- als auch beim Globalspeicher wird immer in den Cache geschrieben, unabhängig davon, ob der Datenblock dort vorhanden ist oder nicht. Für die Anwendung ist der Ausgabevorgang beendet, sobald die Daten im Cache stehen. Das Schreiben auf das Plattenspeichersystem wird i.d.R. unmittelbar danach angestoßen.

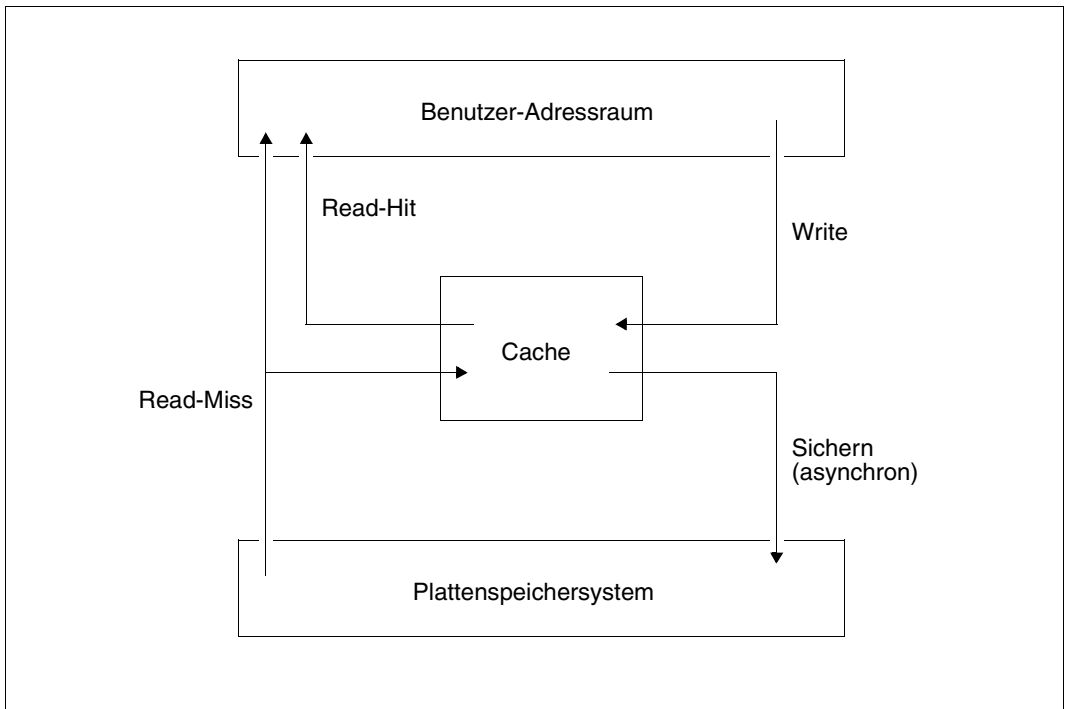
Read-Write-Cache

Bild 9: Ein-/Ausgabe-Abläufe bei einem Read-Write-Cache

Read-Hit: wie Read-Cache

Read-Miss: wie Read-Cache

Write: wie Write-Cache

7.3.2 ADM-PFA-Caching

Beim **ADM-PFA-Caching** steuert die Systembetreuung die Einrichtung von Cache-Bereichen und die Zuordnung von ausgewählten Datenbereichen (ein Cache-Bereich ist stets ein Teilbereich eines gesamten Cache-Speichers, der eine eigene Verwaltungseinheit darstellt). Mit `/START-DAB-CACHING` kann die Systembetreuung voneinander unabhängige Cache-Bereiche einrichten und damit ausgewählte Daten verschiedener Anwendungen gegeneinander abschotten. Auch für ganze Volumes kann ein Cache-Bereich eingerichtet werden. Eine ausführliche Beschreibung ist im Handbuch „DAB“ [5] zu finden.

Mit dem Begriff **AutoDAB** wird eine Variante des DAB-Cachings bezeichnet, bei der zum einen durch automatische und intelligente Verfahren die Systemverwaltung bei der Administration des Cachings entscheidend entlastet wird, und zum anderen dadurch auch die Effektivität des DAB-Cachings nochmals gesteigert werden kann.

Ein DAB-Cache-Bereich kann entweder als „automatischer“ oder „nicht-automatischer“ Cache-Bereich eingerichtet werden.

Für die Nutzung des Cache-Bereichs kann eine der drei folgenden Möglichkeiten eingestellt werden:

- a) Der berechtigte Anwender spezifiziert, welche Dateien mit welchem Cache-Modus den eingerichteten Cache-Bereich nutzen sollen.
- b) Der Systemverwalter legt fest, dass alle Anwender-Dateien eines Pubsets den Cache-Bereich nutzen.
- c) Der Systemverwalter richtet einen automatischen Cache-Bereich ein, bei dem die optimale Auswahl der zu unterstützenden Dateien automatisch erfolgt. Diese Auswahl wird permanent überwacht und ggf. korrigiert. Mit AutoDAB können auch ganze Pubsets durch einen Cache-Bereich abgedeckt werden.

DAB verwaltet jeden Cache-Bereich in Form von so genannten Cache-Segmenten. Die Standardgröße eines Cache-Segmentes beträgt bei automatischen Cache-Bereichen 32 KB. Sie kann auch bei nicht automatischen Cache-Bereichen von der Systembetreuung mit 4, 8, 16 oder 32 KB festgelegt werden (das automatische Caching muss während der Änderung gestoppt werden).

Beim ersten Lesezugriff auf einen mit DAB unterstützten Bereich (Datei) werden von DAB mit Seitenkettung benachbarte Datenblöcke, deren Anzahl von der eingestellten Cache-Segmentgröße abhängt, vom Plattenspeichersystem gelesen.

Räumliche Lokalität

Bei Lesezugriffen mit guter räumlicher Lokalität (vor allem bei sequenziellen Zugriffen) ermöglichen große Segmente (= 32 KB) einen so hohen Durchsatz von Ein-/Ausgabe-Aufträgen, dass in diesem Fall eine große Anzahl von Read-Hits auftritt und sich gleichzeitig die Zugriffe auf das Plattenspeichersystem auf ein Minimum reduzieren.

Bei Lesezugriffen mit geringerer räumlicher Lokalität (Random-Zugriff, im Datenbank-Betrieb die zumeist benutzte Zugriffsart) empfiehlt es sich, kleine Segmente (= 4 KB) zu wählen.

AutoDAB erkennt gute Lokalität und führt bei guter Lokalität automatisch einen Prefetch durch (auch wenn ein kleineres Cache-Segment eingestellt wurde).

Zeitliche Lokalität

Die in den Cache eingelesenen Blöcke bleiben im Cache stehen, bis DAB beendet oder dieser Speicherbereich für ein anderes Cache-Segment benutzt wird.

Verdrängungen werden nach dem LRU-Prinzip (Least Recently Used) verwaltet, wodurch bei guter zeitlicher Lokalität der Zugriffe schon mit einem relativ kleinen Cache-Bereich eine hohe Read-Hit-Rate erreicht werden kann.

Für performance-kritische Anwendungen, deren Lesezugriffe eine schlechte zeitliche Lokalität aufweisen, bietet DAB die Möglichkeit der residenten Zwischenpufferung.

Schreibaufträge werden bei einem Read-Write-Cache (bzw. Write-Cache) in jedem Fall direkt in den Cache eingetragen, unabhängig davon, ob der Datenblock bereits im Cache ist oder nicht. Für die Anwendung ist der Ausgabevorgang beendet, sobald die Daten in den Cache geschrieben sind. Die Daten werden von DAB zu einem späteren Zeitpunkt auf das Plattenspeichersystem gesichert.

Um mit DAB das Cache-Medium Globalspeicher nutzen zu können, müssen DAB-Partitionen im Globalspeicher eingerichtet werden (siehe Handbücher „Systembetreuung“ [10] und „DAB“ [5]).

Bei Verwendung des Cache-Mediums Hauptspeicher ist zu beachten, dass DAB die Systempuffer im Hauptspeicher resident anlegt, d.h. der Hauptspeicher muss entsprechend dimensioniert sein.

Mit den folgenden Kommandos steuert die Systembetreuung den Einsatz von ADM-PFA-Caching mit DAB:

Kommando	Funktion
START-DAB-CACHING	Cache-Bereiche attributieren und einrichten
SHOW-DAB-CACHING	Informationen über die aktuelle DAB-Konfiguration und Zugriffszähler (Hit-Raten) ausgeben
STOP-DAB-CACHING	Cache-Bereiche auf Platte sichern und auflösen
MODIFY-DAB-CACHING	Parameter eines DAB-Cache-Bereichs dynamisch ändern
FORCE-STOP-DAB-CACHING	Auflösung eines DAB-Cache-Bereichs ohne Sicherung der Schreibdaten durchführen

Einsatzempfehlungen

- Genereller Einsatz von AutoDAB

Die Auswahl der zu puffernden Datenbasis und die Einstellung des Prefetching-Faktors sollte durch das automatisierte Caching erfolgen. Es ist deshalb empfehlenswert, die performance-kritischen Datenträger oder Pubsets mit AutoDAB zu bedienen.

Im Regelfall ist die folgende Vorgehensweise einfach und zielführend:

1. Bestimmung aller performance-kritischen Anwendungen
2. Bestimmung aller Volumes/Pubsets, die von diesen Anwendungen angesprochen werden
3. Zusammenfassung aller dieser Volumes/Pubsets zu einem automatischen ADM-PFA-Cache-Bereich (günstig für den Fall, dass über die Zugriffsintensität auf einzelne Pubsets nichts Näheres bekannt ist) oder Definition eines automatischen PFA-Cache-Bereichs für jedes einzelne Pubset (siehe auch [Abschnitt „User-PFA-Caching“ auf Seite 184](#))

- Festlegung der Cache-Bereich-Größe

Die Größe des Cache-Bereichs sollte mindestens 1% der Größe der zu unterstützenden Datenbasis (parallel bearbeitete Dateien) betragen. Bei überwiegend sequenzieller Verarbeitung kann dieser Wert unterschritten werden, bei überwiegend Random-Verarbeitung ist dieser Wert u.U. zu klein. Ist der Cache zu klein gewählt, so wird dies von AutoDAB als Konsolmeldung mitgeteilt.

- Beschleunigung aller Schreibzugriffe:

Unabhängig von der Lokalität beschleunigt DAB bei einem Read-Write-Cache (bzw. Write-Cache) alle Schreibzugriffe, solange freie Puffer zur Verfügung stehen. Damit lassen sich kurzzeitige Lastspitzen abfangen und schnelle Dateizugriffszeiten erreichen. Da die im Cache zwischengelagerten Schreibdaten aber von DAB auf das Plattenspeichersystem gesichert werden müssen, ist bei Schreibenanwendungen nur bei ausreichend hoher Lokalität der Zugriffe eine Entlastung des Ein-/Ausgabe-Systems möglich.

- Home-Pubset

Der Home-Pubset kann nicht mit PFA in den DAB-Cache-Medien Haupt- und Global-speicher zwischengepuffert werden. Dies ist mit ADM-PFA-Caching zwar möglich, sollte aber entweder nur mit reinem Read-Cache erfolgen oder im nichtflüchtigen Globalspeicher mit Schreib-Cache nur für solche Dateien, auf die vor „System Ready“ nicht zugegriffen wird.

- SM-Pubsets

Daten eines SM-Pubsets können nicht mit ADM-PFA-Caching in einem DAB-Cache-Medium zwischengepuffert werden, da der Ablageort der Datei, d.h. der Volume-Set, dynamisch geändert werden kann.

- Datenbank-Pubsets

Bei reinen Datenbank-Pubsets kann es zur Verbesserung der „zeitlichen Lokalität“ besser sein, eine kleinere Segmentgröße zu wählen: z.B. 8 KB statt dem Standardwert 32 KB (/START-DAB-CACHING, Parameter CACHE-SEGMENT-SIZE).



In der Praxis führen die verkürzten Ein-/Ausgabezeiten i.d.R. zu einer deutlichen Durchsatzerhöhung und damit zu Verschiebungen der Last und der CPU-Auslastung.

Bei Anwendungen, die eine höhere Write-IO-Rate verursachen, sollten zur Steigerung des Durchsatzes zusätzliche Platten verwendet werden.
(Der Einsatz von DAB für Write-IOs wird nicht generell empfohlen.)

Ausführliche Informationen zum Einsatz des DAB sind im Handbuch „DAB“ [5] zu finden.

7.3.3 User-PFA-Caching

Beim User-PFA-Caching (PFA: Performant File Access) entscheidet die Systembetreuung, welche Pubsets einen Cache-Bereich zugeordnet bekommen, welche Eigenschaften dieser Cache-Bereich hat und welche Anwender vom Caching dieser Pubsets profitieren sollen.

Eine ausführliche Beschreibung ist im Handbuch „Einführung in die Systembetreuung“ [10] zu finden.

PFA aus der Sicht des Anwenders

Abhängig von der Definition des Cache-Bereichs kann die Nutzung eines PFA-Caches für einen Pubset entweder global für alle Dateien oder gezielt für spezielle Dateien erfolgen:

- Ist der Cache-Bereich mit der Eigenschaft `CACHED-FILES=*BY-SYSTEM` (d.h. AutoDAB ist im Einsatz) oder `CACHED-FILES=*ALL` (alle Anwender-Dateien des Pubsets unterliegen dem Caching) definiert, so muss der Anwender keine weiteren Aktionen durchführen, um vom Caching auf diesem Pubset zu profitieren.
- Ist der Cache-Bereich mit `CACHED-FILES=*BY-USER-SELECTED` definiert, so wirkt das Caching wie anschließend beschrieben, nur für entsprechend attributierte Dateien.

Mit `/CREATE-FILE`, `/MODIFY-FILE-ATTRIBUTES`, `/ADD-FILE-LINK` bzw. den Makros `FILE` und `CATAL` können folgende Dateieigenschaften vergeben werden:

Operand	Bedeutung
PERFORMANCE = *STD = *HIGH = *VERY-HIGH = *USER-MAX	bestimmt das Caching-Verhalten (Hauptspeicher/Globalspeicher): kein Caching Caching mit Verdrängung nach LRU Caching ohne Verdrängung höchstes Performance-Attribut, zu dem der Anwender berechtigt ist
USAGE = *READ-WRITE = *READ = *WRITE	die erhöhten Performance-Anforderungen gelten: für Lese- und Schreiboperationen nur für Leseoperationen nur für Schreiboperationen
DISK-WRITE = *IMMEDIATE = *BY-CLOSE = *STD	Datenkonsistenz wird hergestellt: nach jeder Schreiboperation (Voraussetzung: nichtflüchtiges Cache-Medium!) erst nach CLOSE-Verarbeitung für permanente Dateien gilt *IMMEDIATE, für temporäre Dateien gilt *BY-CLOSE

Werden im Rahmen einer Folgeverarbeitung (i.d.R. bei Batch-Betrieb) mehrmals die gleichen Dateien geöffnet bzw. geschlossen, so kann der Anwender den Durchsatz weiter verbessern, indem er beim Schließen der Datei

- die Invalidierung der Lesedaten im Cache verhindert bzw.
- das Sichern der im Cache befindlichen Schreibdaten unterbindet (bei flüchtigem Speichermedium nur für temporäre Dateien zu empfehlen).

Zusätzlich wirkt sich günstig aus, dass wiedereröffnete Dateien nicht erneut in den Cache eingelagert werden müssen.

Ermöglicht wird diese Variante des PFA-Cachings (HIPERBATCH = High Performance Batch Processing) über `/ADD-FILE-LINK, Operand CLOSE-MODE= *KEEP-DATA-IN-CACHE,` bzw. den Makro `CLOSE, Operand KEEP-DATA-IN-CACHE.`

PFA aus der Sicht der Systembetreuung

Damit die Performance-Anforderungen des Anwenders wirksam werden können, müssen entsprechende Berechtigungen in den Benutzerkatalog eingetragen werden (`/ADD-USER,` `/MODIFY-USER-ATTRIBUTES` bzw. `/MODIFY-USER-PUBSET-ATTRIBUTES`):

Operand	Bedeutung
DMS-TUNING-RESOURCES = *NONE = *CONCURRENT-USE = *EXCLUSIVE-USE	vereinbart Performance-Maßnahmen Caching nicht zugelassen für Anforderung PERFORMANCE=*HIGH für Anforderung PERFORMANCE=*VERY-HIGH
PERM-/TEMP-/WORK-SPACE-LIMIT = *PARAMETERS(...) HIGH-PERF-SPACE= ... VERY-HIGH-PERF-SPACE= ...	Speicherplatz-Kontingente der Anwender festlegen (jeweils für permanente, temporäre und Arbeitsdateien): für hochperformanten Speicherplatz für sehr hochperformanten Speicherplatz

Es ist darauf zu achten, dass Berechtigung und Speicherplatz-Kontingente im Benutzerkatalog desjenigen Pubsets eingetragen werden, auf dem die Datei liegt.

Mit den folgenden Kommandos steuert die Systembetreuung den Einsatz von PFA-Caching:

Kommando	Funktion
MODIFY-PUBSET-CACHE-ATTRIBUTES	Cache-Bereichseigenschaften im MRSCAT eines Pubsets oder Volume-Sets festlegen
IMPORT-/EXPORT-PUBSET	bewirkt implizit die Einrichtung bzw. Auflösung eines im MRSCAT beschriebenen Cache-Bereichs
START-/STOP-PUBSET-CACHING	dynamische Einrichtung bzw. Auflösung eines im MRSCAT beschriebenen Cache-Bereichs während einer Pubset-Session
SHOW-CACHE-CONFIGURATION	PFA-Cache-Bereichs-Konfiguration ausgeben
SHOW-PUBSET-CACHE-ATTRIBUTES	Cache-Bereichs-Beschreibung im statischen und dynamischen MRSCAT ausgeben
FORCE-DESTROY-CACHE	Auflösung eines PFA-Cache-Bereichs erzwingen
MODIFY-PUBSET-DEFINITION-FILE	(logische) Performance-Eigenschaften eines Volume-Sets eines SM-Pubsets festlegen
MODIFY-PUBSET-PROCESSING	bewirkt implizit die Einrichtung bzw. Auflösung von Cache-Bereichen bei der aktiven Hinzunahme bzw. Entfernung von Volume-Sets zu bzw. aus einem SM-Pubset

Die Definition des Cache-Mediums zum entsprechenden Pubset erfolgt über die Operanden von /MODIFY-PUBSET-CACHE-ATTRIBUTES:

Operand	Bedeutung
CACHE-MEDIUM =*MAIN-MEMORY =*GLOBAL-STORAGE	bestimmt das Cache-Medium: Hauptspeicher Globalspeicher
CACHE-SIZE=n	Größe des Cache-Bereichs
FORCE-OUT =*AT-HIGH-FILLING =*AT-LOW-FILLING =*NO	legt fest, ab welchem Cache-Füllungsgrad beschriebene Inhalte auf Platte gesichert werden: 75% 25% keine periodische Sicherung

Bezüglich des Operanden `CACHE-MEDIUM` sind noch folgende Angaben wichtig:

Operand	Bedeutung
<code>CACHE-SEGMENT-SIZE</code> <code>=*4KB/*8KB/*16KB/</code> <code>*32KB</code>	Größe eines Cache-Segments, d.h. minimale Größe des Datenbereichs, der bei einem Read-Miss eingelagert wird: 4, 8, 16 oder 32 KB (nur für nicht-automatische Cache-Bereiche relevant)
<code>CACHED-FILES</code> <code>=*BY-USER-SELECTED</code> <code>=*ALL</code> <code>=*BY-SYSTEM</code>	bestimmt, welche Dateien den Cache nutzen sollen: über Performance-Attribute bestimmt der Anwender, welche seiner Dateien den Cache nutzen sollen alle Anwender-Dateien nutzen den Cache automatischer Cache-Bereich: durch das automatisierte, intelligente Caching des AutoDAB werden die performance-relevanten Dateien automatisch ermittelt; Für die ausgewählten Dateien wird der zu ihrem Zugriffsprofil passende Prefetch-Faktor eingestellt, die Dateien werden zyklisch überwacht, um eine optimale Cache-Performance zu gewährleisten.

Der Einsatz von User-PFA-Caching ist erforderlich, wenn bei einem Shared-Pubset ein lokales Write-Caching für jeden Server unterstützt werden soll (bei ADM-PFA-Caching ist nur Read-Caching möglich).

7.4 Transaktionsbetrieb und BCAM

Dieser Abschnitt beschreibt Maßnahmen zur Antwortzeitoptimierung im Transaktionsbetrieb und bei BCAM.



Die Netzqualität hat in vielen Fällen den größten Einfluss auf das Antwortzeitverhalten. Paketverluste, Paketverfälschungen und Reihenfolgevertauschungen zwingen das Transportsystem zu Fehlerbehandlungen, die zu Paketwiederholungen und einer verminderten Sendeleistung führen. Eine Netzanalyse muss neben den Endsystemen alle anderen Netzkomponenten, wie Switches, Router, Firewalls, Gateways usw. einbeziehen. Die Messkomponenten von BS2000/OSD lassen sich durch SNMP-Agenten in eine solche Infrastruktur integrieren.

7.4.1 Phasen einer OLTP-Transaktion

Die performance-relevanten Phasen einer OLTP-Transaktion (siehe [Abschnitt „Online-Anwendung“ auf Seite 16](#)) sind:

1. Netzlaufzeit bis zum Server
2. Warten auf die Annahme der Transaktion durch BCAM
3. Bearbeitung durch BCAM und Weiterleitung an die TP-Anwendung (z.B. openUTM)
4. Warten auf Annahme durch openUTM
5. Bearbeitungszeit der Transaktion, gesteuert durch openUTM mit folgenden Detailphasen:
 - a) CPU-Bedarf durch die Anwendung / durch openUTM (mit Warten auf CPU-Zuteilung)
 - b) Ein-/Ausgabe-Bearbeitung in der Anwendung / in openUTM (mit Warten auf das Ende der Ein-/Ausgabe)
 - c) Warten auf Auftragsannahme durch Datenbank-Tasks
 - d) CPU-Bedarf durch Datenbank-Aktivitäten (mit Warten auf CPU-Zuteilung)
 - e) Bearbeitung der Ein-/Ausgabe in der Datenbank-TaskDiese Detailphasen können mehrfach auftreten
6. Senden der Antwort an BCAM
7. Warten auf die Annahme der Antwort durch BCAM
8. Bearbeitung durch BCAM

9. Übergabe der Ausgabedaten für den Netztransfer
10. Netzlaufzeit bis zum Client

Nur die Phasen 3 - 9 können mit BS2000-Mitteln (im Wesentlichen mit openSM2, siehe Handbuch „openSM2“ [18]) messtechnisch erfasst werden.

Für die Phase 2 liefert BCAM einen Indikator über mögliche zeitliche Verzögerungen. openSM2 stellt im Messprogramm BCAM-Connection die von BCAM erfassten Zeiten grafisch dar:

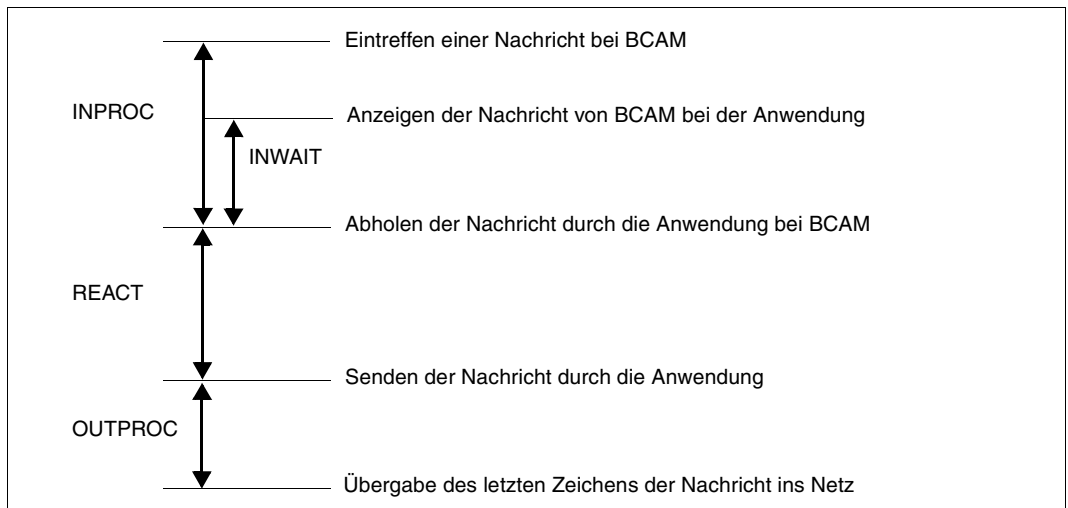


Bild 10: BCAM-Zeiten

Folgende Zeiten entsprechen sich:

INPROC	Phasen 3 und 4
INWAIT	Phase 4
REACT	Phasen 5 - 7
OUTPROC	Phasen 8 und 9

7.4.2 Optimierung der einzelnen Phasen

Dieser Abschnitt behandelt Engpässe im openUTM- und BCAM-Umfeld, die zu Antwortzeitverlängerungen in den einzelnen Phasen führen können. Er zeigt auf, wie diese Engpässe messtechnisch erkannt und auf welche Art sie behoben bzw. gemildert werden können.

Phasen 1 und 10: Netzlaufzeiten

Allgemeine Hinweise zum LAN-Anschluss finden Sie im [Abschnitt „LAN-Anschluss“ auf Seite 87](#).

Phase 2: Warten bis zur Annahme der eingehenden Nachricht durch BCAM (Inbound Flow Control):

Steuerungsmechanismen zur Kontrolle des Datenflusses schützen das Transportsystem vor Datenüberflutung durch die Partner. Dadurch wird verbindungspezifisch ein (schneller) Sender an die Verarbeitungsgeschwindigkeit eines (langsamen) Empfängers angepasst.

BCAM schützt so seinen Speicherpool, der allen Anwendungen und Verbindungen zur Verfügung steht. In Phasen, in denen der BCAM-Pool stark ausgelastet ist, hindert BCAM seine Partner daran, Nachrichten ins BS2000-System zu senden. Dadurch kann es zu steigenden Antwortzeiten kommen.

Ursache für eine länger oder dauerhaft anhaltende hohe Auslastung des BCAM-Pools können Anwendungen sein, die ihre Nachrichten langsam abholen. Auch eine, für die zu erbringende Verarbeitungsleistung, zu geringe Poolgröße kann zu einer hohen Auslastung des BCAM-Pools führen.

openSM2 stellt Messdaten für die Analyse der Auslastung des BCAM-Pools zur Verfügung, siehe Messprogramm `RESPONSE TIME` im Handbuch „openSM2“ [18].

BCAM selbst erlaubt es mit dem Kommando `/BCMON` und durch Analyse der Werte der Konsolmeldung `BCA0B21` zu ermitteln, ob ein Engpass im BCAM-Puffer vorliegt.

Kurzbeschreibung der Vorgehensweise:

- Ermittlung der eingestellten Maximalwerte (optional) mit
`/SHOW-BCAM-PARAMETERS PARAMETER=*LIMITS`
- Einschalten des BCAM-Monitoring (Ausgabe der Messwerte alle <n> Sekunden):
`/BCMON MODE=ON,RECORD=RES-MEM,SEC=<n>`

- Beschreibung der wichtigsten Ausgabe-Werte der Konsolmeldung `BCA0B21`:

<code>USED-BCAM-MIN-I</code>	minimale BCAM-Pufferbelegung für eingehende Nachrichten in KB
<code>LIMIT-ACT</code>	aktuelle obere Grenze für die Pufferbelegung in KB

- Indikator für eine hohe Auslastung des BCAM-Puffers durch eingehende Nachrichten:
(USED-BCAM-MIN-I) * 4 > LIMIT-ACT
- Alternativ oder ergänzend zu den obigen Punkten kann mit /BCSET THRESHOLD-MSG=ON eine Warnmeldung angefordert werden.

Dann wird die Konsolmeldung BCAB021 ausgegeben, wenn BCAM wegen Pool-Engpass die eingehenden Nachrichten länger als 5 Sekunden bremst (oder keine Sende-anforderungen der Anwendung annehmen kann, siehe Phase 7).
Diese Warnmeldung wird mit /BCSET THRESHOLD-MSG=OFF wieder ausgeschaltet.
- Optimierungsmaßnahme bei einem Engpass im BCAM-Pool:
Mit dem Kommando /BCMOD RESMEM=<n> sollte der maximal zulässige Schwellwert deutlich erhöht werden (ideal wäre: <n>=3*LIMIT-ACT).

Phasen 3 und 8: Bearbeitung der Ein- bzw. Ausgangs-Nachricht in BCAM

Die Zeitwerte für diese Phasen sollten im niedrigen einstelligen Bereich (in Millisekunden) liegen.

Phase 4: Warten auf openUTM

Die Zeit, die zwischen der Auftragserteilung an openUTM und der Annahme des Auftrags verstreicht, wird als INWAIT-Zeit bezeichnet. Sie wird pro Anwendung ermittelt und ist in der INPROC-Zeit enthalten.

openSM2 erfasst die INWAIT-Zeiten (wie auch die INPROC-, REACT- und OUTPROC-Zeiten) in 5 Zeitintervallen, den so genannten Buckets. Buckets können nur global, nicht aber anwendungsspezifisch eingestellt werden. Zur Vereinfachung der Überwachung sollten Intervall-Einteilung so vorgenommen werden, dass alle nicht akzeptablen Zeiten in das letzte Intervall fallen (Overrun-Wert).

Beispiel

Die normale Antwortzeit beträgt 100 - 200 ms (typisch für große S-Server). Es sollen kurzfristige Schwankungen bis zum Faktor 2 toleriert werden. Eine länger andauernde Verdopplung der Antwortzeit soll aber nicht toleriert werden.

Die Buckets in openSM2 sollten demnach so eingestellt werden, dass im Overrun-Intervall alle INWAIT-Zeiten gezählt werden, die 400 ms oder größer sind, z.B. mit
/SET-BCAM-CONNECTION-PARAMETER INWAIT-BUCKETS=(50,100,200,400)

Mit dieser Anweisung werden die Buckets so eingestellt, dass im ersten Intervall alle Wartezeiten < 50 ms, im zweiten Intervall alle Wartezeiten zwischen 50 und 100 ms, im dritten Intervall alle Wartezeiten zwischen 100 und 200 ms, im vierten Intervall alle Wartezeiten zwischen 200 und 400 ms und im Overrun-Intervall alle Wartezeiten > 400 ms gezählt werden.

Mit den Komponenten INSPECTOR oder ANALYZER von openSM2 müssen nun (für jede Anwendung) die Messwerte des Overrun-Intervalls überwacht werden. Die Werte werden in Prozent (bezogen auf die im Messintervall erfassten Zeitwerte) dieser Applikation ausgegeben. Ab einem Prozentwert von 10 oder höher deutet dies auf Engpässe bei der Auftragsannahme durch die UTM-Tasks hin.

Messmöglichkeit in openUTM

Mit dem UTM-Kommando `KDCINF STAT` werden mehrere nützliche UTM-Messwerte ausgegeben, siehe openUTM-Handbuch „Anwendungen administrieren“ [20]. Für die Analyse, ob die Zahl der UTM-Tasks einen Engpass darstellen könnte, liefert der Ausgabewert `%Load` einen wichtigen Hinweis. Dieser gibt die mittlere Auslastung aller UTM-Tasks des letzten Messintervalls an. Ein zumindest kurzzeitiger Engpass deutet sich an, wenn der Wert größer als 90 (%) wird.

Schwellwertüberwachung der UTM-Taskanzahl mit openSM2

Ein sich anbahnender Engpass an UTM-Tasks lässt sich mit dem Messreport UTM-Applikation von openSM2 erkennen. Dazu müssen von diesem Report die Werte für die Dauer einer Transaktion in Sekunden (`DT`), die Anzahl der Transaktionen pro Sekunde (`TX`) und die Anzahl der UTM-Tasks für die Applikation (`UT`) ermittelt werden.

Damit kann die mittlere Auslastung der UTM-Tasks einer Anwendung errechnet werden:

$$\text{Load (in \%)} = 100 * DT * TX / UT.$$

Im INSPECTOR von openSM2 kann dieser errechneten Wert einer Schwellwertüberwachung unterworfen werden. Bei Überschreitung eines Wertes von 90 (%) kann z.B. eine entsprechende E-Mail an die Systembetreuung erzeugt werden.

Optimierungsmaßnahme

Mit dem UTM-Kommando `KDCAPPL TASKS=<n>` kann die Anzahl der UTM-Tasks applikationsspezifisch und schrittweise solange erhöht werden, bis die INWAIT-Zeiten in einem akzeptablen Bereich liegen. Die so gefundene optimale Taskanzahl sollte in der Startparameter-Datei von openUTM eingetragen werden. Sie wirkt dann ab dem nächsten Anwendungsstart. Übersteigt `<n>` den im KDCDEF-Lauf festgelegten maximalen Wert, so muss dieser Maximal-Wert erhöht und ein neuer KDCDEF-Lauf gestartet werden.

Bei der Veränderung der Anzahl der UTM-Tasks müssen immer auch die Anzahl der TAC-Klassen und die Anzahl der Tasks in diesen TAC-Klassen berücksichtigt werden. Ebenso ist ein gewisser Puffer für Lastspitzen zu berücksichtigen.

Anmerkung

Nur wenn deutlich mehr UTM-Tasks als nötig gestartet werden, kann dies zu geringen Leistungseinbußen durch die etwas höhere Hauptspeicher- und CPU-Nutzung führen.

Phase 5 und 6: Bearbeitung der Transaktion

Die in dieser Phase (und in der Phase 7) anfallende Zeit wird im BCAM-Connection-Report als REACT-Zeit ausgewiesen.

Tuning der IO-Performance beim Zugriff auf die KDCFILE

Neben den in diesem Handbuch beschriebenen Maßnahmen bei der Hardware und in BS2000/OSD kann die Performance von Anwendungen mit hohen Transaktionsraten auch in openUTM durch optimierte Schreibzugriffe auf die KDCFILE verbessert werden. Dazu können in openUTM die Pagepools und/oder der Wiederanlaufbereich aus der KDCFILE ausgelagert werden. Diese ausgelagerten Bereiche werden nun selbst wieder auf mehrere Volumes verteilt.

Beispiel:

Der Pagepool soll auf 2 Public Volumes,
der Wiederanlaufbereich auf 4 Public Volumes verteilt werden:

```
/CREATE-FILE FILE-NAME=<filebase>.P01A, -
SUPPORT=*PUBLIC-DISK(VOLUME=<v1>,SPACE=*RELATIVE(PRIMARY-ALLOCATION=666))

/CREATE-FILE FILE-NAME=<filebase>.P02A, -
SUPPORT=*PUBLIC-DISK(VOLUME=<v2>,SPACE=*RELATIVE(PRIMARY-ALLOCATION=666))

/CREATE-FILE FILE-NAME=<filebase>.R01A, -
SUPPORT=*PUBLIC-DISK(VOLUME=<v3>,SPACE=*RELATIVE(PRIMARY-ALLOCATION=300))

/CREATE-FILE FILE-NAME=<filebase>.R02A, -
SUPPORT=*PUBLIC-DISK(VOLUME=<v4>,SPACE=*RELATIVE(PRIMARY-ALLOCATION=300))

/CREATE-FILE FILE-NAME=<filebase>.R03A, -
SUPPORT=*PUBLIC-DISK(VOLUME=<v5>,SPACE=*RELATIVE(PRIMARY-ALLOCATION=300))

/CREATE-FILE FILE-NAME=<filebase>.R04A, -
SUPPORT=*PUBLIC-DISK(VOLUME=<v6>,SPACE=*RELATIVE(PRIMARY-ALLOCATION=300))
```

Außerdem müssen in der KDCDEF die folgenden Parameter in der MAX-Anweisung geändert werden: PGPOOLSFS=2, RECBUFFS=4.

Im KDCDEF-Lauf werden dann die oben definierten Dateien verwendet, wobei das KDCDEF-Programm eventuell die Werte für PRIMARY- und SECONDARY-ALLOCATION modifiziert. Ohne obige Kommandos würde KDCDEF die Dateien selbst anlegen (ohne Volume-Zuweisung).

Nach dem Neustart von openUTM werden die neuen Dateien verwendet.

Steuerung der UTM-Aufträge durch TAC-Klassen

Ähnlich der Kategoriesteuerung mit PCS (siehe [Abschnitt „Einführung in das PCS-Konzept“ auf Seite 222](#)) können Transaktionen in openUTM in so genannte „TAC-Klassen“ aufgeteilt werden. Diese TAC-Klassen können mit zwei nicht kombinierbaren Methoden gesteuert werden:

- Prioritätensteuerung
- Prozessbeschränkung

Details zur Auftragssteuerung in openUTM finden Sie im openUTM-Handbuch „Anwendungen generieren“ [\[21\]](#).

Einsatzempfehlung:

- Bei der Aufteilung der TACs in TAC-Klassen muss darauf geachtet werden, dass höher-priore TAC-Klassen nicht durch TACs aus nieder-prioren TAC-Klassen gebremst werden (Verwendung von blockierenden Aufrufen).
- Es wird empfohlen, nicht mehr als 3 oder 4 TAC-Klassen zu definieren.
- Die Prioritätensteuerung dient vor allem dazu, dass bei (kurzfristigen) Überlasten lang laufende TACs aus nieder-prioren TAC-Klassen kurz laufende TACs aus höher-prioren TAC-Klassen nicht behindern. Somit werden hoch-priore Transaktionen unter Ausnutzung aller gestarteten Prozesse bevorzugt.
- Die Prozessbeschränkung dient dazu, dass lang laufende TACs andere, kurz laufende TACs nicht behindern können. Der Vorteil dieser Variante liegt darin, dass hier immer dafür gesorgt werden kann, dass genügend freie UTM-Tasks für neue (und wichtige) TACs zur Verfügung stehen.
- Zusätzlich können TACs durch Angabe der `RUNPRIO` in der TAC-Anweisung gesteuert werden. Dies ist aber nur für höchst-priore TACs zu empfehlen und ist mit dem Prioritätengefüge aller in BS2000/OSD ablaufenden Tasks abzustimmen.

Weitere Performance-Hinweise zu openUTM:

- Überwachung der Cache Hit-Rate
Mit dem UTM-Kommando `KDCINF STAT` (siehe [Seite 192](#)) sollte (zumindest gelegentlich) überprüft werden, ob die Trefferquote im Pagepool des UTM-Caches ausreichend hoch ist. Bei Werten unter 90 (%) sollte der UTM-Cache eventuell vergrößert werden (siehe openUTM-Handbuch „Anwendungen generieren“ [\[21\]](#)).
- Einsatz von UTM-F
Diese UTM-Variante ist hauptsächlich für Anwendungen mit „retrieval“-Zugriffen und keinen bis sehr wenigen Updates geeignet. Sie dient zur Vermeidung von Ein-/Ausgaben bei reduzierter Funktionalität (hinsichtlich Ausfallsicherheit und Wiederanlauf).

Phase 7: Warten vor BCAM

Diese Phase ist im BCAM-Connection-Report ein Teil der REACT-Zeit (siehe Phase 5).

Analog zu Phase 2 kann BCAM den Steuerungsmechanismen des Partners zur Kontrolle des Datenflusses unterliegen. Dies ist erkennbar durch ZWR-Werte > 0 (ZWR=Zero Window Receive) im openSM2-Report BCAM. BCAM nimmt zwar in solchen Situationen eine gewisse zu versendende Datenmenge der Anwendung an, wird jedoch bei Überschreiten eines Grenzwertes die Annahme von weiteren Daten verweigern.

BCAM verzögert die Annahme von Daten auch in Phasen, in denen der BCAM-Pool stark ausgelastet ist. Ursache für eine länger oder dauerhaft anhaltende hohe Auslastung des BCAM-Pools können Verbindungen sein, die der Kontrolle des Datenflusses unterliegen. Auch eine, für die zu erbringende Verarbeitungsleistung, zu geringe Poolgröße kann zu einer hohen Auslastung des BCAM-Pools führen.

Analog zu Phase 2 bietet BCAM eine Überwachungsmöglichkeit.

Zeigt die Meldung `BCA0B21` hohe Werte für `USED-BCAM-MIN-0` an, so sollte wie in Phase 2 der BCAM-Pool vergrößert werden.

Phase 8 und 9: Verzögerung bei der Übergabe ins Netz

Die Zeit in Phase 8 (reine BCAM-Verarbeitung) liegt im einstelligen Millisekunden-Bereich. Die Zeiten für Phase 8 und 9 sind gemeinsam in der OUTPROC-Zeit enthalten. Eine Überwachung der OUTPROC-Zeit analog zur INPROC-Zeit wird empfohlen (siehe [Seite 191](#)). Hohe Werte im Overrun-Intervall deuten auf mögliche Probleme der Netzperformance hin (z.B. durch Fehlfunktionen in Routern oder Switches).

8 Systemsteuerung

8.1 Job-Management

Das Job-Management (JMS) bietet eine Vielzahl von Steuermöglichkeiten für die Auswahl auf Verarbeitung wartender Aufträge (Jobs).

Zur Charakterisierung bestimmter Eigenschaften von Aufträgen dienen die Job-Klassen. Neben Parametern für die Auswahl von Aufträgen enthalten sie auch wesentliche Anweisungen für die anschließende Verarbeitung. Mit diesen Verarbeitungs-Parametern können auch andere Anwendungen, wie TP- oder Dialog-Anwendungen, hinsichtlich ihrer Ausführung gesteuert werden. Die Angabe einer Job-Klasse ist also nicht nur für die Batch-Verarbeitung wichtig.

Vom Anwender gewünschte Auswahl-Strategien für Aufträge (Job-Scheduling-Strategien) werden durch Job-Streams realisiert.

8.1.1 Job-Klassen

Der Parameter `JOB-TYPE=*BATCH` einer Job-Klasse legt fest, dass diese Klasse dem Job-Scheduling unterworfen wird. Aufträge einer Job-Klasse, für die `JOB-TYPE=*DIALOG` gilt, werden sofort gestartet, sofern der Parameter `CLASS-LIMIT` dies erlaubt.

Auswahlparameter für Batch-Aufträge

`CLASS-WEIGHT`

kennzeichnet die Dringlichkeit der in einer Job-Klasse wartenden Aufträge gegenüber einer anderen Klasse (nur wichtig nach System-Sättigungszuständen).

Wertebereich `<integer 1..9>`; 1 ist die niedrigste, 9 die höchste Dringlichkeit.

`JOB-PRIORITY`

legt die Wichtigkeit der wartenden Aufträge innerhalb einer Job-Klasse fest.

Wertebereich `<integer 1..9>`; 1 ist die höchste, 9 die niedrigste Wichtigkeit.

CPU-LIMIT

bestimmt die CPU-Zeit, die ein Auftrag in dieser Job-Klasse verbrauchen darf.
Der Parameter `NO-CPU-LIMIT=*YES` bewirkt das gleiche wie der Eintrag `PRIVILEGE=*NO-CPU-LIMIT` im Benutzerkatalog.

START

gibt die möglichen Startattribute an (`*SOON`, `*WITHIN`,...).
Die Zusatzbedingung `ALLOWED=*IMMEDIATE` hat die gleiche Wirkung wie der Eintrag `PRIVILEGE=*START-IMMEDIATE` im Benutzerkatalog. Sie bedeutet, dass ein Auftrag sofort gestartet werden darf, auch wenn das Job-Klassen-Limit bereits erreicht ist.

Wird in einem XCS-Verbund bei `/ENTER-JOB` bzw. `/ENTER-PROCEDURE` der Operand `HOST=*ANY` angegeben, so erfolgt die Verteilung der Aufträge auf die einzelnen Rechner des Verbunds automatisch (Details siehe Handbuch „HIPLEX MSCF“ [12]).

Für die Auswahl des Zielrechners werden die Kriterien in folgender Reihenfolge berücksichtigt:

1. Status der Job-Klasse

Die gewünschte Job-Klasse muss definiert und aktiv sein (sie darf z.B. nicht durch `/HOLD-JOB-CLASS` angehalten sein).

2. Kein Vorliegen eines System-Sättigungszustandes

3. Belegung der Job-Klasse

Solange das Job-Klassen-Limit (`CLASS-LIMIT`) nicht erreicht ist, wird für Aufträge die sofort ausgeführt werden sollen, die Differenz zwischen Job-Klassen-Optimum (`CLASS-OPTIMUM`) und der Anzahl bereits gestarteter Aufträge berücksichtigt.

Ist das Job-Klassen-Limit ausgeschöpft oder überschritten, wird die Differenz zwischen dem Job-Klassen-Limit und der Anzahl akzeptierter (d.h. bereits gestarteter und wartender) Aufträge herangezogen.

Ausführungsparameter für Batch-Aufträge, TP- und Dialoganwendungen

Damit Anwendungen in der gewünschten Kategorie unter Berücksichtigung des entsprechenden Task-Attributs TP, DIALOG oder BATCH (siehe auch [Abschnitt „Task-Management“ auf Seite 203](#)) sowie der gewünschten Prioritäten ausgeführt werden, sind folgende Parameter zu beachten:

TP-ALLOWED = *NO / *YES(CATEGORY=<name>)

DIALOG-ALLOWED = *NO / *YES(CATEGORY=<name>)

BATCH-ALLOWED= *NO / *YES(CATEGORY=<name>)

Diese Parameter legen das Task-Attribut für die Job-Klasse fest. Damit werden spezielle systeminterne Ablaufparameter, optimiert jeweils für die Betriebsarten TP, Dialog oder Batch, zugewiesen.

Ist kein Kategorienname angegeben, wird der zugehörige Standard-Kategorienname angenommen (z.B. bei TP-ALLOWED=*YES der Standard-Kategorienname TP).

Ein Kategorienname darf nicht mehreren Task-Attributen zugeordnet werden.

START-ATTRIBUTE = *BATCH / *DIALOG / *TP

bestimmt, mit welchem Task-Attribut Tasks dieser Job-Klasse gestartet werden.

Die Parameterwerte von START-ATTRIBUTE und TP-, DIALOG- und BATCH-ALLOWED müssen zusammenpassen.

RUN-PRIORITY

legt die Ausführungspriorität (= Task-Priorität) fest.

Mit dem Parameter RUN-PRIORITY=*PARAMETERS(DEFAULT=n, MAXIMUM=n) kann die maximale Task-Priorität, die sich ein Anwender geben darf, vorgegeben werden.

Beispiel

Eine wichtige Produktionsanwendung (Auftragsabwicklung) soll als TP-Anwendung mit hoher Priorität in einer eigenen Kategorie gefahren werden.

Mit dem Dienstprogramm JMU (siehe Handbuch „Dienstprogramme“ [6]) wird eine Job-Klasse definiert (//DEFINE-JOB-CLASS), das Zugriffsrecht festgelegt (//GRANT-JOB-CLASS-ACCESS) und die Ablage in der Datei SJMSFILE vorgenommen.

```
//DEFINE-JOB-CLASS NAME=PRI01,
  CLASS-LIMIT=5,
  CLASS-WEIGHT=2,
  JOB-PRIORITY=*PARAMETERS(DEFAULT=3),
  JOB-TYPE=*BATCH,
  TP-ALLOWED=*YES(CATEGORY=<name>),
  START-ATTRIBUTE=*TP,
  RUN-PRIORITY=*PARAMETERS(DEFAULT=150,MAXIMUM=128),
  NO-CPU-LIMIT=*YES,
  CPU-LIMIT=*PARAMETERS(DEFAULT=*NO-LIMIT,MAXIMUM=32767),
  SYSLST-LIMIT=*PARAMETERS(DEFAULT=*NO-LIMIT),
  START=*PARAMETERS(DEFAULT=*SOON,ALLOWED=...)
```

Mit //SET-MODIFICATION-MODE wird festgelegt, ob das Dienstprogramm JMU die Änderung direkt im laufenden System ausführt und/oder in der Datei SJMSFILE, die erst beim nächsten Systemstart ausgewertet wird.

Für den Start der Produktionsanwendung genügt bei /ENTER-JOB die Angabe JOB-CLASS=PRI01.

Anmerkung

Aus Kompatibilitätsgründen besteht eine logische ODER-Verknüpfung zwischen den Einträgen der Job-Klasse und der Benutzerkennung, wobei die jeweils „größeren“ Rechte gelten.

Beispiel

Job-Klasse	Benutzerkatalog	Status
TP-ALLOWED=*NO	MAX-ALLOWED-CATEGORY=TP	TP erlaubt
RUN-PRIORITY= *PARAMETERS(DEFAULT=150, MAXIMUM=128)	MAXIMUM-RUN-PRIORITY=180	maximale Priorität=128

8.1.2 Job-Streams

BS2000/OSD wird mit vordefinierten Job-Klassen geliefert. Wenn in einem BS2000-System keine Job-Klassen definiert sind, so werden alle Aufträge der Klasse \$SYSJC zugeordnet (auf die der Anwender keinen Zugriff hat) und vom fest mit dem System gekoppelten **System-Job-Scheduler \$SYSJS** verwaltet.

Der System-Job-Scheduler ist eine System-Task (siehe [Abschnitt „Erzeugen und Beenden von Tasks“ auf Seite 243](#)) und läuft privilegiert im Funktionszustand TPR. Seine auf dem LIFO-Prinzip (Last in first out) beruhende Auswahlstrategie ist nicht beeinflussbar.

Unabhängig von definierten Job-Klassen und eventuell realisierten Auswahlstrategien lässt der System-Job-Scheduler während des gesamten Systemlaufs das Starten von Aufträgen unter der Kennung TSOS zu.

Definierte Job-Klassen mit dem Parameter `JOB-TYPE=*BATCH` werden einem Job-Stream (auch Job-Scheduler genannt) zugeordnet (`STREAM=name`). Eine Job-Klasse kann nur **einem** Job-Stream angehören, aber ein Job-Stream kann mehrere Job-Klassen verwalten.

Die Bearbeitung der Jobs im Stream (Sortieren nach vorgegebener Auswahlstrategie) übernimmt der **Standard-Job-Scheduler**. Der Standard-Job-Scheduler ist ein Anwenderprogramm und läuft im Funktionszustand TU unter der Kennung TSOS. Die Auswahlstrategie wird durch den Stream-Parameter (`STREAM-PARAMETER`) bestimmt. Er wird durch das Dienstprogramm JMU (`//DEFINE-JOB-STREAM`) eingestellt.

Um verschiedene Auswahlstrategien zu realisieren, können die bei der Stream-Definition festgelegten Scheduling-Parameter dynamisch geändert werden (`//MODIFY-JOB-STREAM` des Dienstprogramms JMU).

Bis zu 16 Job-Streams können maximal gleichzeitig installiert sein.

Auswahl-Strategien (`STREAM-PARAMETER='...'`)

- Auswahl nach Ankunftszeit

`JOB-PRIORITY=NO,CPU-TIME=NO,WAIT-TIME=NO`

`WAIT-TIME` ist die Wartezeit des Auftrags nach Annahme.

Diese Strategie empfiehlt sich, wenn die CPU-Zeitforderungen der Aufträge in etwa gleich sind.

- Auswahl nach Priorität

`JOB-PRIORITY=YES,CPU-TIME=NO,WAIT-TIME=NO`

Gewährleistet die Bevorzugung wichtiger Jobs.

- Auswahl nach CPU-Zeitbedarf

JOB-PRIORITY=NO, CPU-TIME=YES, WAIT-TIME=NO

Jobs mit geringeren CPU-Zeitforderungen werden vorgezogen.

- Auswahl nach Priorität und CPU-Zeitbedarf

JOB-PRIORITY=YES, CPU-TIME=YES, WAIT-TIME=NO

Bei Aufträgen mit gleicher Job-Scheduling-Priorität entscheidet der geringere CPU-Zeitbedarf. Bei gleichem CPU-Zeitbedarf ist die höhere Priorität maßgebend.

- Auswahl nach Priorität und Wartezeit

JOB-PRIORITY=YES, CPU-TIME=NO, WAIT-TIME=YES

Durch Einbeziehung der Wartezeit nach der Job-Annahme haben auch niederpriore Aufträge die Chance, ausgewählt zu werden.

- Auswahl nach Durchsatz

JOB-PRIORITY=NO, CPU-TIME=YES, WAIT-TIME=YES

Bevorzugung von Aufträgen mit geringem CPU-Zeitbedarf unter Einbeziehung der Wartezeit nach der Annahme.

- Auswahl nach Durchsatz und Priorität

JOB-PRIORITY=YES, CPU-TIME=YES, WAIT-TIME=YES

Zusätzlich zur Auswahl nach Durchsatz wird die Job-Scheduling-Priorität berücksichtigt.

Der Job-Scheduler übergibt die von ihm ausgewählten und sortierten Jobs an den **Klassen-Scheduler**.

Der Klassen-Scheduler ist keine eigene Task, sondern gehört zum Kern des Job-Managements. Er sorgt für die Einhaltung des von der Systembetreuung gewünschten Job-Mixes (über die CLASS-LIMIT-Parameter der einzelnen Job-Klassen).

Bei der Auflösung von Stau-Situationen nach einer Systemsättigung (Sättigung des Seitenwechselbereichs) bestimmt der Parameter CLASS-WEIGHT die Dringlichkeit der einzelnen Job-Klassen.

8.2 Task-Management

In BS2000/OSD werden logisch zusammengehörende Anforderungen an das System und deren Ausführung als Task verwaltet.

Wenn es sich bei diesen Anforderungen an das System um UTM-Anwendungen handelt, dann können die unterschiedlichen Transaktionscodes innerhalb von UTM mit Hilfe so genannter TAC-Klassen gesteuert werden. Dabei stehen ähnliche Mittel zur Verfügung, wie sie für Tasks bzw. Kategorien unter PRIOR bzw. PCS eingesetzt werden.

Eine Zusammenfassung dazu befindet sich auf [Seite 194](#).

Detaillierte Informationen finden Sie in den in den Handbüchern von openUTM.

8.2.1 Einführung in das PRIOR-Konzept

Unter dem Namen PRIOR sind jene Routinen des Task-Managements zu verstehen, die es ermöglichen:

- Tasks mit Hilfe von Kategorien und Prioritäten zu steuern
- die Serverauslastung durch interne Mechanismen zu überwachen und zu steuern

Konkret bedeutet dies:

- die Verwaltung der Betriebsmittel Hauptspeicher und CPU samt Durchführung entsprechender Kontrollfunktionen
(siehe [Abschnitt „Verwalten der Betriebsmittel Hauptspeicher und CPU“](#) ab [Seite 244](#))
- die Steuerung der Tasks über Warteschlangen
(siehe [Abschnitt „Steuern der Tasks über Warteschlangen“](#) ab [Seite 261](#))

Das Konzept beruht auf einer Prioritäten-Steuerung ohne Einflussnahme auf die Größe der Betriebsmittelzuteilung. Das bedeutet, dass der Systembetreuung nur Einfluss darüber eingeräumt wird, welcher von mehreren Bewerbern ein Betriebsmittel zuerst erhält. Er hat nur geringen Einfluss darauf, welchen Anteil an der Leistung eines Betriebsmittels ein Bewerber in einer gewissen Zeitspanne erhält.

8.2.1.1 Task-Kategorien

Zur Verbesserung der individuellen Steuerungsmöglichkeiten – je nach den Anwendererfordernissen – erfolgt die Einteilung der Tasks in Kategorien. Es gibt 16 Kategorien.

Vier davon sind immer existent mit den Standard-Kategorienamen:

SYS	nur für System-Tasks
TP	Transaktions-Tasks
DIALOG	Dialog-Tasks
BATCH	Batch-Tasks

Weitere Kategorien können von der Systemverwaltung unter einem frei wählbaren Namen eingerichtet werden.

Jede Task ist einer Kategorie zugeordnet.

Zusätzlich gibt es vier Task-Attribute, deren Namen identisch sind mit den Standard-Kategorienamen SYS, TP, DIALOG und BATCH. Den Task-Attributen sind spezielle, für das Task-Scheduling wichtige Ablaufparameter zugeordnet.

Das Task-Attribut TP zeichnet sich gegenüber den anderen Task-Attributen durch eine speziell auf die Bedürfnisse des Transaktionsbetriebs optimierte Hauptspeicher-Verwaltung aus (wenige, hinsichtlich Working-Set-Bedarf meist größere User-Tasks hängen funktional zusammen und bedienen viele Terminals).

Das Task-Attribut TP kann erlangt werden durch Definition in der Job-Klasse oder durch Aufruf des TINF-Makros (in diesem Fall muss die dazu erforderliche Berechtigung im Benutzerkatalog eingetragen sein).

Wesentliches Kennzeichen einer Kategorie ist der Multiprogramming-Level (MPL). Darunter ist diejenige Anzahl Tasks pro Kategorie zu verstehen, die das Recht zur Hauptspeicher-Nutzung besitzen, sich also im aktiven Zustand befinden.

Mit den Kategorie-Attributen MIN MPL, MAX MPL und WEIGHT (/MODIFY-TASK-CATEGORIES) spezifiziert die Systembetreuung die Wichtigkeit der Kategorien untereinander für die Aktivierung (= Zuteilung der Berechtigung zur Hauptspeicher-Nutzung).

Ob es zu einer Aktivierung kommt oder ob eine Task-Verdrängung durchgeführt wird, hängt außer von den Kategorie-Attributen auch von der Systemauslastung und der Priorität der betroffenen Task ab (siehe [Abschnitt „Verwalten der Betriebsmittel Hauptspeicher und CPU“ auf Seite 244](#)).

Die Vorgabe von **MIN MPL** garantiert eine gewisse Mindestunterstützung einer Kategorie. Das System versucht vorrangig, den spezifizierten MIN MPL-Wert zu erreichen.

Die Vorgabe von **MAX MPL** bedeutet keine feste Grenze, d.h. es wird über den maximalen MPL-Wert hinaus aktiviert, solange kein Betriebsmittelengpass vorliegt.

Durch den Parameter **WEIGHT** wird die Aktivierungs- bzw. Deaktivierungs-Reihenfolge gesteuert. Zusätzlich beeinflusst der Parameter schwach die Vergabe der CPU.

Kategoriewechsel

Die Systembetreuung kann mit `/MOVE-TASK-TO-CATEGORY` die Zuordnung einer Task zu einer Kategorie ändern, wenn etwa eine andere (bessere) Bedienung dieser Task oder auch eine Entlastung einer Kategorie erreicht werden soll (mit oder ohne Einsatz von PCS).

Auswirkungen der Kategoriesteuerung

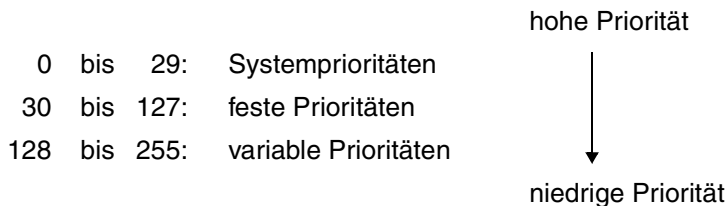
Aufgrund der genannten Eigenschaften hat die Kategoriesteuerung im Unterlast- bzw. Normallast-Bereich keinen nennenswerten steuernden Einfluss.

Im **Voll-Last-Bereich** bzw. **Überlast-Bereich** (also bei Betriebsmittelengpässen) hat die Kategorie-Steuerung eine starke Wirkung und dient daher zur **Lastbegrenzung**. Weniger wichtige Kategorien werden in den Hintergrund gedrängt.

Dieser Voll-Last-Bereich wird bei Servern mit sehr großem Hauptspeicher nur noch in Ausnahmefällen erreicht. Die Überlastung der CPU reicht alleine nicht aus, um die Lastbegrenzung wirksam werden zu lassen. Daher ist bei diesen Servern die Kategoriesteuerung weitgehend wirkungslos.

8.2.1.2 Task-Prioritäten

Durch die Vergabe der Priorität kennzeichnet der Anwender die Dringlichkeit seiner Anforderungen hinsichtlich der Ausführung. Es gibt 3 Klassen von Prioritäten:



Für den Anwender stehen die variablen und die festen Prioritäten zur Verfügung. Sie werden in dieser Form als externe Prioritäten bezeichnet.

Die Prioritätsvergabe ist möglich:

- vom Anwender auf Kommandoebene bei `/SET-LOGON-PARAMETERS`
`/ENTER-JOB`
`/CHANGE-TASK-PRIORITY`
- vom Anwender auf Programmebene mit dem Makroaufruf **TINF**
- vom Systemverwalter bzw. Operator als Standardwert in der Job-Klasse (`RUN-PRIORITY`) und mit `/CHANGE-TASK-PRIORITY`

Die maximal erlaubte Priorität muss in der Job-Klasse bzw. im Benutzerkatalog eingetragen sein.

Variable Prioritäten

Charakteristisch für variable Prioritäten ist die dynamische Einstellung der internen Priorität mit dem HRN-Algorithmus (Highest Response ratio Next), basierend auf dem Verhältnis $\text{Verweilzeit (Elapsed Time)} / \text{Verbrauchte CPU-Zeit}$ unter Berücksichtigung einer „Startpriorität“. Diese Startpriorität ergibt sich aus der vergebenen externen Priorität, den Kategorie-Gewichten, dem Leistungsvermögen des Servers und der derzeitigen Belastung.

Dies gewährleistet eine angemessene Bedienung auch für Tasks mit niedriger Startpriorität.

Die variable Priorität wird zu folgenden Zeitpunkten berechnet bzw. verbessert:

- bei der Aktivierung (siehe [Abschnitt „Hauptspeicher-Verwaltung“ auf Seite 246](#))
- bei Micro-Time-Slice-Runout (siehe [Abschnitt „Kontrollfunktion der CPU-Verwaltung“ auf Seite 260](#))
- periodisch (jede Sekunde)

Die periodische Überprüfung wird für alle aktiven Tasks sowie für alle ablaufbereiten, inaktiven Tasks durchgeführt (die also auf die Zuteilung der Berechtigung zur Hauptspeicher-Nutzung warten, sich im Zustand „inactive ready“ befinden).

Feste Prioritäten

Bei festen Prioritäten bleibt die vorgegebene externe Priorität unverändert. Sie wird starr in die interne Priorität umgerechnet.

Auswirkungen der Priorität

Da die Priorität sowohl bei der Aktivierung als auch bei der Initiierung berücksichtigt wird, zeigt die Vergabe einer von der „normalen“ externen Priorität 255 abweichende, also bessere Startpriorität bei allen Lastsituationen (bei Unter-, Voll- oder Überlast) ähnliche Auswirkungen.

Bei der Einreihung einer Ein-/Ausgabe-Anforderung in die Warteschlange vor dem Gerät wird die Priorität nicht berücksichtigt.

Die Priorität wirkt daher umso stärker, je CPU-intensiver eine Task und deren Konkurrenten sind.

Empfehlungen und Richtwerte

PRIOR steuert die Zuteilung der Betriebsmittel Hauptspeicher und CPU an User- und System-Tasks entsprechend der vorgegebenen Kategorie- und Prioritätsparameter.

Dabei spielen die Prioritäten eine besonders wichtige Rolle, da man mit ihnen einzelne Tasks bevorzugen kann, während mit der Kategoriesteuerung nur Kategorien als Ganzes priorisiert werden können. Besonders bei TP-Anwendungen sollten die zentralen Tasks in der Priorität angehoben werden. Der Einsatz von festen Prioritäten ist sehr vorsichtig zu handhaben. Sie unterliegen nicht dem HRN-Algorithmus und können daher den Server monopolisieren. Es sollten keinesfalls bessere Prioritäten als 80-100 vergeben werden.

Anmerkung

Die Parameter für die **Kategorie SYS** können nicht verändert werden und sind auf folgende Werte eingestellt:

```
MIN MPL = 30
MAX MPL = 64
WEIGHT  = 512
```

System-Tasks haben folgende Prioritäten:

Permanent vorhandene System-Tasks:	22	
Ausnahmen:		
TSN VMM:	16	
TSN PGE:	1	
TSN RMM:	128	beim Startup
	50	nach SYSTEM READY
Dynamisch erzeugte Tasks (z.B. DIAA, DSSM):	60	
Ausnahmen (z.B. BCAM):	30	

Grundsätzlich ist zu beachten, dass man eine Betriebsmittelknappheit auch durch geschickte Einstellungen nicht beseitigen kann. Während mit viel Erfahrung bei einem CPU-Engpass oder bei knappem Hauptspeicher wenigstens noch für besonders wichtige Tasks zu Lasten anderer eine Erleichterung geschaffen werden kann, ist dies bei Engpässen in der Peripherie nur schlecht möglich.

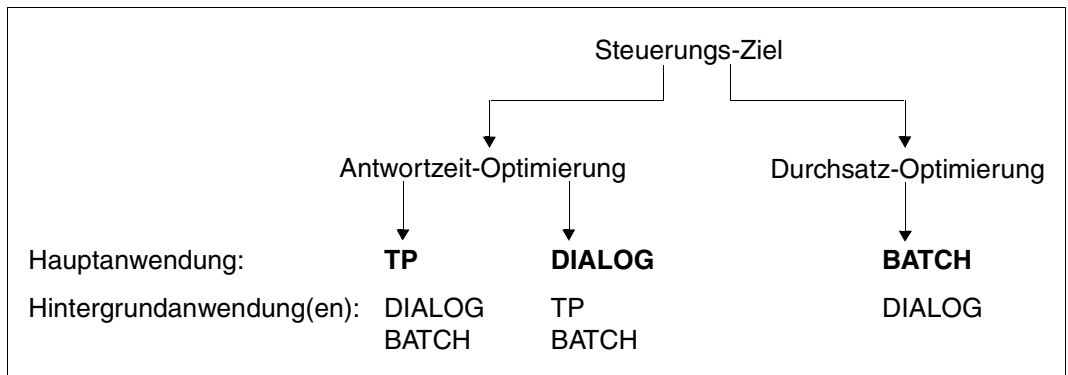
In den folgenden Abschnitten wird davon ausgegangen, dass Überlegungen hinsichtlich aufgabengerechter Dimensionierung bereits im Rahmen der Kapazitätsplanung erfolgt sind. Das Hauptaugenmerk wird auf die optimale Nutzung der installierten Betriebsmittel gelegt.

8.2.1.3 Lastanalyse

Um aus der Vielfalt der Parameter-Einstellmöglichkeiten von PRIOR die günstigste Einstellung auswählen zu können, ist die genaue Kenntnis der Last erforderlich. Folgende Fragen müssen im Rahmen der Lastanalyse beantwortet werden:

- Welche Leistungsanforderungen bestehen für welche Anwendung?
- Welche Anwendung kann im Überlastfall zugunsten wichtigerer Anwendungen in den Hintergrund gedrängt werden?
- Welche Anwendung gehört in welche Kategorie?

Zunächst ist es notwendig, das Steuerungs-Ziel zu definieren:



Die Unterscheidung in Antwortzeit- bzw. Durchsatz-Optimierung ist wichtig, weil hier gegenläufige Tendenzen vorliegen: Gutes Antwortzeitverhalten setzt kurze Wartezeiten vor den Betriebsmitteln (speziell vor Kanälen, Steuerungen und Plattenlaufwerken) voraus; daher darf deren Auslastung nicht zu hoch sein.

Die weitere Unterscheidung im Rahmen der Antwortzeit-Optimierung, ob der Schwerpunkt bei TP- oder Dialog-Anwendungen liegt, ist für die Einstellung der Kategorieparameter von Bedeutung.

- Dialog-Anwendungen stellen eine härtere Hauptspeicher-Beanspruchung hinsichtlich der Paging-Intensität dar als TP-Anwendungen (siehe auch [Abschnitt „Untersuchen eines systemorientierten Leistungsproblems“ auf Seite 291](#)).
- TP-Anwendungen reagieren andererseits empfindlicher auf Hintergrundlasten, besonders auf Paging. Diese Empfindlichkeit ist umso größer, je weniger Tasks die Abwicklung der Anwendung bewerkstelligen. Die TP-Anwendung ist als Hauptanwendung anzusehen, wenn die Anzahl der TP-Transaktionen $\geq 50\%$ der Gesamtanzahl an Transaktionen beträgt.

Die Anzahl Transaktionen pro Sekunde für die TP- und Dialog-Anwendung kann entweder nach den Formeln im [Abschnitt „Formulieren einer Leistungserwartung“ auf Seite 20](#) oder durch eine Messung mit dem Software-Monitor SM2 ermittelt werden.

Beispiel: Transaktionsrate

600 TP-Terminalbenutzer:

Durchschnittliche Denkzeit: 20 s (inklusive Netzlaufzeit)

Durchschnittliche Antwortzeit: 1 s

50 Dialog-Tasks:

Durchschnittliche Denkzeit: 20 s (inklusive Netzlaufzeit)

Durchschnittliche Antwortzeit: 2 s

Transaktionsrate für die TP-Anwendung: $600 / (1 + 20) = 28,6$ Trans/s

Transaktionsrate für die Dialog-Anwendung: $50 / (2 + 20) = 2,3$ Trans/s

Man sollte in jedem Fall eine Vermessung mit dem Software-Monitor SM2 durchführen (speziell bei komplexeren Anwendungen). Es empfiehlt sich die **Vermessung von Teillas-**
ten.

Zweckmäßigerweise beginnt man mit dem wichtigsten Teil der Last, setzt die PRIOR-Parameter für die Hauptanwendung entsprechend den Empfehlungen (siehe [„Einstellen der Kategorieparameter“ auf Seite 211](#) und [„Prioritätenvergabe“ auf Seite 219](#)) und ermittelt das Antwortzeitverhalten, die Transaktionsrate und die Betriebsmittelauslastung. Diese Werte werden für die Optimierung der Hauptanwendung im Mischlastbetrieb benötigt, um beurteilen zu können, wie stark die Hauptanwendung durch die Nebenanwendungen beeinträchtigt wird.

Durch diese Vermessung der Hauptanwendung können die in der Kapazitätsplanung getroffenen Annahmen verifiziert bzw. der tatsächliche Betriebsmittelbedarf aufgezeigt werden. Tritt bereits beim alleinigen Betrieb der Hauptanwendung ein unbefriedigendes Leistungsverhalten auf, so ist eine Engpass-Analyse durchzuführen (siehe [Abschnitt „Durchführung einer Engpass-Analyse“ auf Seite 282](#)).

Im Anschluss daran stellt man die PRIOR-Parameter für die Hintergrundanwendung entsprechend den Empfehlungen ein (siehe [„Einstellen der Kategorieparameter“ auf Seite 211](#) und [„Prioritätenvergabe“ auf Seite 219](#)) und erhöht stufenweise die Last.

Beispiel: Antwortzeit-Optimierung

Bevorzugung der TP-Anwendung vor der Dialog- und Batch-Anwendung

– Schritt 1:

Messen der reinen TP-Anwendung (modifizierte PRIOR-Parameter)

Messzeit je nach Lastfall: 15 Minuten bei kaum schwankender Last

60 Minuten bei stark schwankender Last

Ermitteln von Antwortzeitverhalten, Transaktionsrate, Betriebsmittelauslastung

– Schritt 2:

Dazuschalten der Dialog-Anwendung (modifizierte PRIOR-Parameter)

Messzeit: 60 Minuten

– Schritt 3:

Dazuschalten der Batch-Anwendung (modifizierte PRIOR-Parameter)

Messzeit: mehrere Messungen, jeweils 60 Minuten

Beobachten des Laufzeitverhaltens

Die Hauptanwendung wird durch die Nebenanwendung eine gewisse Bremsung erfahren (ca. 10 - 20%). Das ist unvermeidbar. Durch geschickte Wahl und Begrenzung der Nebenanwendungen erreicht man, dass diese Bremsung erträglich bleibt. Die Nebenanwendungen sollten möglichst nur solche Betriebsmittel belasten, die von der Hauptanwendung kaum benötigt werden. Zusätzlich ist eine hohe Empfindlichkeit der Nebenanwendungen bezüglich MPLs und Prioritäten vorteilhaft.

Anwenderrechte

Nach der Lastanalyse ist es notwendig, über `/MODIFY-USER-ATTRIBUTES` bzw. die `JOB-CLASS`-Parameter die „endgültigen“ Anwenderrechte (welche Priorität der Anwender höchstens vergeben darf) festzulegen.

Auch ist zu überprüfen, ob für DCAM-, UTM-, UDS-, SESAM-Anwendungen die TP-Berechtigung eingetragen ist, damit die Vorteile des Task-Attributs TP zur Geltung kommen können.

8.2.1.4 Einstellen der Kategorieparameter

Vorrangiges Ziel ist für Server mit knappem Speicher die effiziente Aufteilung des **verfügbaren** seitenwechselbaren Hauptspeichers NPP (Number of Pageable Pages) entsprechend der angegebenen Wichtigkeit der Kategorien.

Knapper Speicher liegt vor, wenn es bei vernünftig eingestellten MPL-Werten INACTIVE READY TASKS gibt (Report „Active and inactive tasks for category“ in der Reportgruppe CATEGORY-QUEUE von SM2).

Wird dieser Zustand im Betrieb nie erreicht, dann kann der Leser diesen Abschnitt überspringen und zum Abschnitt „[Prioritätenvergabe](#)“ auf Seite 219 gehen.

Damit PRIOR auf kleinere Lastschwankungen variabel reagieren kann, sollte der Working-Set-Bedarf der Tasks, der sich aus der Summe der MIN MPL-Werte aller Kategorien ergibt, $(0,3 \text{ bis } 0,5) * NPP$ sein. Die Wahl des Faktors (0,3 oder 0,5) hängt davon ab, wo der Schwerpunkt der Anwendung liegt (siehe unten).

Richtwerte für den durchschnittlichen Working-Set-Bedarf (4-KB-Seiten):

TP-Tasks: $WS_{TP} = 200 \text{ bis } 400$ Seiten für UTM- und DCAM-Tasks
 $= 500 \text{ bis } 1.000$ Seiten für DB-Tasks

Dialog-Tasks: $WS_{DIAL} = 50 \text{ bis } 150$ Seiten

Batch-Tasks: $WS_{BATCH} = 50 \text{ bis } 200$ Seiten

Die Verwaltung des Working-Sets für Systemroutinen und Common-Memory-Pools, die von Anwenderprogrammen aufgerufen werden, erfolgt global in der Kategorie SYS (Platzhalter ist die TSN=PGE).

Kategorie SYS: $WS_{SYS} = 2.000 \text{ bis } 4.000$ Seiten

Der NPP-Wert ist dem dem ACTIVITY-Report von SM2 zu entnehmen.



Bei den folgenden Überlegungen wird der Einfachheit halber angenommen, dass aus Anwendersicht nur die Kategorien mit den Standard-Kategorienamen TP, DIALOG und BATCH vorhanden sind.

Hauptanwendung: TP

Hintergrundanwendung: Dialog, Batch

Kategorie TP

In dieser Kategorie befinden sich die Tasks mit den höchsten Leistungsansprüchen. Auch im Überlastfall ist eine Lastbegrenzung dieser Kategorie nicht erwünscht. Daher muss der Wert für MAX MPL gleich der Gesamtanzahl der im System vorhandenen TP-Tasks sein (inklusive der Verwaltungstasks, z.B. TDADM, Datenbankadministrator, ...).

Der Wert für MIN MPL sollte der gewünschten Anzahl aktiver TP-Tasks entsprechen. Dies wird i.d.R. nahe oder gleich dem MAX MPL-Wert sein. Ein zu kleiner Wert für MIN MPL führt bei Überlast zu unerwünschter Zwangsdeaktivierung (siehe [Abschnitt „Hauptspeicher-Verwaltung“ auf Seite 246](#)) von TP-Tasks.

Empfohlene Werte:

$$\text{MIN MPL}_{\text{TP}} = 0,8 * \text{MAX MPL}_{\text{TP}}$$

$$\text{MAX MPL}_{\text{TP}} = \text{Gesamtanzahl vorhandener TP-Tasks}$$

$$\text{WEIGHT} = 500$$

Kategorie DIALOG

Diese Kategorie ist meist durch stärkere Lastschwankungen gekennzeichnet. PRIOR versucht, den von der Systembetreuung vorgegebenen MIN MPL-Wert notfalls durch Verdrängungen von Tasks anderer Kategorien zu erreichen (siehe [Abschnitt „Hauptspeicher-Verwaltung“ auf Seite 246](#)).

Da hier die Dialog-Anwendung im Hintergrund laufen soll, darf der Wert für MIN MPL nicht zu hoch sein.

Durch den Einfluss der Denkzeit sind nicht alle Dialog-Tasks gleichzeitig aktiv. Die Anzahl der aktiven Dialog-Tasks kann mit folgender Formel abgeschätzt werden:

$$\text{Anzahl aktive Dialog-Tasks} = \text{Gesamtzahl Dialog-Tasks} / N$$

$$N = (\text{durchschnitt. Denkzeit} + \text{Antwortzeit}) / \text{durchschnitt. Antwortzeit}$$

Richtwerte:

- durchschnittliche Denkzeit (inklusive Netzlaufzeit) = 16 s
- durchschnittliche Antwortzeit = 4 s

$$N = 20 \text{ s} / 4 \text{ s} = 5$$

Für die Ermittlung des MIN MPL-Wertes ist maßgebend, wieviel seitenwechselbarer Hauptspeicher (gemäß der Formel $0,5 * \text{NPP}$ bei Schwerpunkt TP-Betrieb) nach Abzug des Working-Set-Bedarfs der Hauptanwendung, der Kategorie SYS bzw. evtl. erforderlicher Batch-Tasks für **aktive** Dialog-Tasks verfügbar ist (siehe Beispiel).

Der Wert für MAX MPL ist als Lastbegrenzung für vorübergehend auftretende Überlastsituationen gedacht. Er sollte auf den doppelten Wert von MIN MPL eingestellt werden.

Empfohlene Werte:

$$\text{MIN MPL}_{\text{DIAL}} = \frac{0,5 \cdot \text{NPP} - (\text{MIN MPL}_{\text{TP}} \cdot \text{WS}_{\text{TP}} + \text{MIN MPL}_{\text{BATCH}} \cdot \text{WS}_{\text{BATCH}} + \text{WS}_{\text{SYS}})}{\text{WS}_{\text{DIAL}}}$$

WS_{TP} : durchschnittlicher Working-Set-Bedarf pro TP-Task

WS_{DIAL} : durchschnittlicher Working-Set-Bedarf pro Dialog-Task

WS_{BATCH} : durchschnittlicher Working-Set-Bedarf pro Batch-Task

WS_{SYS} : durchschnittlicher Working-Set-Bedarf für Systemroutinen und Common-Memory-Pools

Anmerkung

Die Berücksichtigung des Batch-Anteils ist an dieser Stelle nur dann erforderlich, wenn für die entsprechende IT-Installation aus betrieblichen Gründen eine Mindestanzahl parallel laufender Batch-Tasks zwingend notwendig ist.

$$\text{MAX MPL} = 2 * \text{MIN MPL}$$

$$\text{WEIGHT} = 100 \quad (50 < \text{WEIGHT} < 200)$$

Kategorie BATCH

Im Normalfall werden Batch-Tasks als Hintergrundlast eingesetzt, um die freien Betriebsmittel zu nutzen. In diesem Fall errechnet sich der Wert für MIN MPL aus der Größe des seitenwechselbaren Hauptspeichers (genauer: $0,5 * \text{NPP}$), abzüglich des Working-Set-Bedarfs für die Hauptanwendung TP, die Dialog-Anwendung und die Kategorie SYS.

Der Wert für MAX MPL richtet sich nach der Serverauslastung und sollte vom MIN MPL-Wert nicht allzuweit auseinanderliegen.

Empfohlene Werte:

$$\text{MIN MPL}_{\text{BATCH}} = \frac{0,5 \cdot \text{NPP} - (\text{MIN MPL}_{\text{TP}} \cdot \text{WS}_{\text{TP}} + \text{MIN MPL}_{\text{DIAL}} \cdot \text{WS}_{\text{DIAL}} + \text{WS}_{\text{SYS}})}{\text{WS}_{\text{BATCH}}}$$

$$\text{MAX MPL} = \text{MIN MPL} + 1$$

$$\text{WEIGHT} = 10 \quad (1 < \text{WEIGHT} < 50)$$

Beispiel zu Hauptanwendung TP

MM = 64 MB

Hauptanwendung TP: 6 UTM-Tasks, 2 DB-Tasks; insgesamt: 10 TP-Tasks

Hintergrundanwendung: 10 Dialog-Tasks, 2 Batch-Tasks (im Minimum)

Kategorie TP:

MIN MPL _{TP}	=	8
MAX MPL _{TP}	=	10
WEIGHT	=	500

Kategorie DIALOG:

Anzahl aktiver Dialog-Tasks = Gesamtzahl Dialog-Tasks / N = 2

MIN MPL _{DIAL}	=	2
MAX MPL _{DIAL}	=	4
WEIGHT	=	100

Kategorie BATCH:

MIN MPL _{BATCH}	=	2
MAX MPL _{BATCH}	=	3
WEIGHT	=	10

64 MB = 16000 Seiten, residenter MM-Bedarf = 1500 Seiten,
also NPP = 14500 Seiten.

$$\begin{aligned} \text{MIN MPL}_{TP} \cdot \text{WS}_{TP} + \text{MIN MPL}_{DIAL} \cdot \text{WS}_{DIAL} + \text{MIN MPL}_{BATCH} \cdot \text{WS}_{BATCH} + \text{WS}_{SYS} &\leq 0,5 \cdot \text{NPP} \\ (6 \cdot 300 + 2 \cdot 800) + 2 \cdot 150 + 2 \cdot 100 + 2000 &= 1800 + 1600 + 300 + 200 + 2000 = \\ 6100 &\leq 0,5 \cdot 14500 = 7250 \end{aligned}$$

Hauptanwendung: Dialog

Hintergrundanwendung: TP, Batch

Kategorie TP

Dass die TP-Anwendung als Hintergrundanwendung erscheint, bedeutet nicht zwangsläufig, dass nur geringe oder gar keine Leistungsansprüche gestellt werden, sondern dass die Anzahl der TP-Transaktionen $< 50\%$ der Anzahl der Gesamttransaktionen ist.

Dieser Fall tritt immer dann auf, wenn TP-Anwendungen neu eingeführt werden und erst eine relativ kleine Anzahl Terminalbenutzer das neue Verfahren benutzt.

Da **eine** TP-Task immer mehrere Terminalbenutzer bedient bzw. diese normalerweise auch einen wesentlich größeren Working-Set-Bedarf gegenüber Dialog-Tasks hat, ist es grundsätzlich nicht wünschenswert, dass TP-Tasks im Überlastfall zwangsdeaktiviert werden.

Es sollte daher auch hier der Wert für MAX MPL gleich der Gesamtanzahl der im System vorhandenen TP-Tasks sein.

Ebenso sollte der Wert für MIN MPL nahe oder gleich dem MAX MPL-Wert gewählt werden.

Empfohlene Werte:

$$\text{MIN MPL}_{\text{TP2}} = 0,8 * \text{MAX MPL}_{\text{TP}}$$

$$\text{MAX MPL}_{\text{TP2}} = \text{Gesamtanzahl vorhandener TP-Tasks}$$

$$\text{WEIGHT} = 500$$

Kategorie DIALOG

Liegt der Schwerpunkt der Anwendung auf der Dialog-Seite, so bedeutet das eine höhere Beanspruchung des Hauptspeichers hinsichtlich Paging-Intensität (siehe auch [Abschnitt „Untersuchen eines systemorientierten Leistungsproblems“ auf Seite 291](#)) als bei TP-Anwendungen.

Der Working-Set-Bedarf der Tasks, der sich aus der Summe der MIN MPL-Werte der Kategorien TP, DIALOG, BATCH ergibt, sollte daher $\leq 0,3 * \text{NPP}$ sein.

Für die Ermittlung des MIN MPL-Wertes ist wesentlich, wieviel seitenwechselbarer Hauptspeicher (gemäß der Formel $0,3 * \text{NPP}$ bei Schwerpunkt Dialogbetrieb) nach Abzug des Working-Set-Bedarfs der TP-Tasks, der Kategorie SYS bzw. notwendiger Batch-Tasks für aktive Dialog-Tasks verfügbar ist (siehe Beispiel).

Auch hier ist für die Zahl der aktiven Dialog-Tasks in erster Näherung einzusetzen:

$$\text{Anzahl aktive Dialog-Tasks} = \text{Gesamtzahl Dialog-Tasks} / 5$$

Empfohlene Werte:

$$\text{MIN MPL}_{\text{DIAL}} = \frac{0,3 * \text{NPP} - (\text{MIN MPL}_{\text{TP}} * \text{WS}_{\text{TP}} + \text{MIN MPL}_{\text{BATCH}} * \text{WS}_{\text{BATCH}} + \text{WS}_{\text{SYS}})}{\text{WS}_{\text{DIAL}}}$$

Anmerkung

Der Batch-Anteil ist an dieser Stelle nur dann zu berücksichtigen, wenn aus betrieblichen Gründen eine gewisse Mindestanzahl parallel laufender Batch-Tasks erforderlich ist.

$$\begin{aligned} \text{MAX MPL} &= 2 * \text{MIN MPL} \\ \text{WEIGHT} &= (300 < \text{WEIGHT} < 500) \end{aligned}$$

Kategorie BATCH

Für den Fall, dass Batch-Tasks als reine Hintergrundlast eingesetzt werden, gelten folgende Werte:

$$\text{MIN MPL}_{\text{BATCH}} = \frac{0,3 \cdot \text{NPP} - (\text{MIN MPL}_{\text{TP}} \cdot \text{WS}_{\text{TP}} + \text{MIN MPL}_{\text{DIAL}} \cdot \text{WS}_{\text{DIAL}} + \text{WS}_{\text{SYS}})}{\text{WS}_{\text{BATCH}}}$$

$$\begin{aligned} \text{MAX MPL} &= \text{MIN MPL} + 1 \\ \text{WEIGHT} &= 10 \quad (1 < \text{WEIGHT} < 50) \end{aligned}$$

Beispiel zu Hauptanwendung Dialog

MM = 64 MB

Hauptanwendung Dialog: 45 Dialog-Tasks

Batch: 2 Batch-Tasks (im Minimum)

Hintergrundanwendung: 2 UTM-Tasks, 1 DB-Task; insgesamt: 4 TP-Tasks

Kategorie TP:

MIN MPL _{TP}	=	3
MAX MPL _{TP}	=	4
WEIGHT	=	500

Kategorie DIALOG:

Anzahl aktiver Dialog-Tasks = Gesamtzahl Dialog-Tasks / N = 9

MIN MPL _{DIAL}	=	9
MAX MPL _{DIAL}	=	18
WEIGHT	=	400

Kategorie BATCH:

MIN MPL _{BATCH}	=	2
MAX MPL _{BATCH}	=	3
WEIGHT	=	10

64 MB = 16000 Seiten, residenter MM-Bedarf = 1500 Seiten,
also NPP = 14500 Seiten.

$$\begin{aligned} \text{MIN MPL}_{TP} \cdot \text{WS}_{TP} + \text{MIN MPL}_{DIAL} \cdot \text{WS}_{DIAL} + \text{MIN MPL}_{BATCH} \cdot \text{WS}_{BATCH} + \text{WS}_{SYS} &\leq 0,3 \cdot \text{NPP} \\ 2 \cdot 200 + 1 \cdot 500 + 9 \cdot 100 + 2 \cdot 100 + 2000 &= 400 + 500 + 900 + 200 + 2000 = \\ &4000 \leq 0,3 \cdot 14500 = 7250 \end{aligned}$$

Hauptanwendung: Batch

Hintergrundanwendung: Dialog

Diese Belastung ist typisch für den Nachtschichtbetrieb von Servern, deren Hauptspeicher für den Tages-Online-Betrieb ausgelegt ist.

Batch-Anwendungen stellen i.d.R. eine weitaus geringere Hauptspeicher-Beanspruchung bezüglich der Paging-Intensität dar als Online-Anwendungen, belasten aber wesentlich mehr die CPU.

Die Anzahl der verkraftbaren Batch-Tasks hängt daher weniger von der Hauptspeicher-Größe als von der CPU-Geschwindigkeit ab.

Die Einstellung der Kategorieparameter hat in diesem Fall keine Bedeutung (eine gewünschte Lastbegrenzung kann mit dem Job-Klassen-Parameter `CLASS-LIMIT` durchgeführt werden).

8.2.1.5 Prioritätenvergabe

Durch die Vergabe einer – gegenüber der Standardpriorität 255 – höheren Priorität ist die gezielte Bevorzugung einzelner Tasks bei der Aktivierung und besonders hinsichtlich der Benutzung des Betriebsmittels CPU möglich.

In der Regel ist die Vergabe einer mittleren variablen Priorität ausreichend.

Variable Priorität

Die Variation der Priorität sollte je nach Last in 5er- oder 10er-Schritten vorgenommen werden, bis das gewünschte Leistungsziel erreicht ist. Größere Prioritätsabstufungen sind notwendig, wenn:

- viele Tasks aktiv sind
- überwiegend Ein-/Ausgabe-intensive Tasks aktiv sind

Abhängig vom Steuerungsziel werden folgende Werte für die Ersteinstellung empfohlen:

Hauptanwendung: TP

Hintergrundanwendung: Dialog, Batch

Kategorie TP

Handelt es sich um eine DB/DC-Anwendung mit mehreren Tasks, so sollten Datenbanksysteme, die im Multithread-Mode arbeiten (z.B. SESAM/SQL, UDS) eine niedrigere Priorität als die zugehörigen DC-Tasks (UTM, DCAM, ...) erhalten. Die pro Datenbankaufruf erforderliche Kommunikation zwischen den DB- und DC-Tasks kann dadurch deutlich verringert werden.

Sind mehrere TP-Anwendungen unterschiedlicher Wichtigkeit vorhanden, bietet sich eine entsprechende Prioritätsabstufung an.

Empfohlener Prioritätsbereich: 130 - 180

Kategorie DIALOG

Die Dialog-Anwendung soll im Hintergrund laufen. In der Regel erfolgt in dieser Kategorie keine Bevorzugung einzelner Dialog-Tasks.

Empfohlene Priorität: 210

Kategorie BATCH

Für reine Batch-Tasks im Hintergrund genügt die Standard-Priorität. Für die Hervorhebung von Tasks innerhalb der Kategorie reicht die Prioritätsanhebung bis auf 240 aus.

Empfohlener Prioritätsbereich: 240 - 255

Hauptanwendung: Dialog

Hintergrundanwendung: TP, Batch

Kategorie Dialog

Im Allgemeinen werden die Dialog-Tasks gleichrangig behandelt.

Empfohlene Priorität: 180

Kategorie TP

Im Gegensatz zur Einstellung der Kategorieparameter, bei der danach getrachtet wird, Deaktivierungen von TP-Tasks auch im Überlastfall zu vermeiden, soll hinsichtlich der Benutzung des Betriebsmittels CPU die Hintergrundanwendung TP die kleinere Priorität erhalten.

Sollten sich bei den Dialog-Tasks Lasten befinden, die einen dem Batch-Betrieb ähnlichen Charakter haben (Übersetzungen, Programmabläufe innerhalb einer Beantwortungszeit), dann ist wie bei „Hauptanwendung TP“ zu verfahren.

Empfohlene Priorität: 210

Kategorie BATCH

Für den Betrieb als reine Hintergrundlast genügt die Standardpriorität.

Empfohlener Prioritätsbereich: 230 - 255



Im Normalfall werden Batch-Tasks als Hintergrundlast eingesetzt, um freie Betriebsmittel zu nutzen. Alle Tasks der Batch-Hintergrundlast sollten mit der niedrigsten Priorität 255 ablaufen, während alle anderen Tasks Prioritäten von 210 oder besser haben. Damit das System die Nutzung der Betriebsmittel regeln kann, müssen unabhängig davon, ob die Kategorie TP oder Dialog die Hauptanwendung darstellt, die im Hintergrund laufenden Batch-Tasks aus rechenintensiven Programmen bestehen.

Sie müssen ihre Ein-/Ausgaben über die Kanäle und Plattenlaufwerke abwickeln, die von der Vordergrundlast nicht benutzt werden.

Feste Prioritäten

Feste Prioritäten sind für Spezialanwendungen mit extremen Realzeitanforderungen vorgesehen. Dabei sind folgende Gesichtspunkte zu beachten:

Feste Prioritäten engen den Entscheidungsspielraum des Systems stark ein. Zur Aktivierung oder Initiierung anstehende Tasks führen bei voll ausgelasteten Betriebsmitteln zu einer sofortigen Verdrängung anderer Tasks.

Um eine Steigerung des Leistungsverhaltens zu gewährleisten, sollten daher feste Prioritäten nur nach genauer Analyse der Last und der Auslastung der Betriebsmittel zusammen mit lastbegrenzenden Maßnahmen (MAX MPL) eingesetzt werden.

Tasks, denen feste Priorität gegeben wird, sollten nicht abschnittsweise oder gänzlich rechenintensiv sein. Auch an ihre Fehlerfreiheit sind erhöhte Anforderungen zu stellen.

8.2.1.6 I/O-Prioritäten-Steuerung für Tasks mit IORM

Normalerweise haben alle Tasks bei der Ausführung von Platten-Ein-/Ausgaben gleiche Priorität.

Eine Ein-/Ausgabe-intensive Anwendung mit niedriger Priorität kann eine andere, höherprioritäre Anwendung behindern, wenn diese Anwendungen Ein-/Ausgaben auf das gleiche (logische) Gerät ausführen. Behinderungen können auch entstehen, wenn die Ein-/Ausgaben auf verschiedene (logische) Geräte ausgeführt werden, die jedoch auf demselben physikalischen Gerät liegen oder über dieselben Pfade angeschlossen, über dieselben Ports erreichbar oder an denselben Kanälen angeschlossen sind.

IORM kann in der Funktion IOPT (I/O Priority Handling for Tasks) solche Konfliktsituationen erkennen und steuernd in den Ein-/Ausgabe-Betrieb eingreifen. Dazu betrachtet IOPT sowohl den Auslastungsgrad der Ein-/Ausgabe-Einheiten (Geräte, Pfade, Ports und Kanäle) als auch die Ein-/Ausgabe-Prioritäten der nutzenden Tasks.

IOPT unterscheidet drei Ein-/Ausgabe-Prioritäten für Tasks:

- HIGH (hohe Ein-/Ausgabe-Priorität)
- MEDIUM (mittlere Ein-/Ausgabe-Priorität)
- LOW (niedrige Ein-/Ausgabe-Priorität)

Den Task-Kategorien (mit Ausnahme von SYS) können mit folgendem Kommando entsprechende Ein-/Ausgabe-Prioritäten (Attribut IO-PRIORITY) zugewiesen werden:

```
/MODIFY-TASK-CATEGORY IO-PRIORITY=*NONE/*HIGH/*MEDIUM/*LOW
```

Nähere Informationen zu IORM und IOPT finden Sie im Handbuch „Dienstprogramme“ [6].

8.2.2 Einführung in das PCS-Konzept

Mit PCS (Performance Control Subsystem) steht ein Lastregelungssystem zur Verfügung, das – aufsetzend auf PRIOR – folgende zusätzliche Funktionen bringt:

- verbesserte Bevorzugung von Lastanteilen durch gezielte SERVICE-Zuteilung (der SERVICE eines Servers wird ermittelt aus der Belegungszeit der CPU, Hauptspeicher (MEMORY) und der Anzahl Ein-/Ausgaben (IO))
- erhöhte Transparenz bezüglich der Aufteilung der Serverleistung auf die Kategorien
- dynamische Anpassung der SERVICE-Zuteilung an Kategorien bzw. Tasks bei Lastschwankungen
- Durchsatzkontrolle durch Überwachung der Leistungsaufnahme in den Kategorien
- Differenzierung der Last in „Kurzläufer“ und „Langläufer“ und damit verbesserte Steuerungsmöglichkeiten über automatischen Kategoriewechsel
- Verbesserung des Antwortzeitverhaltens durch Bevorzugung der Kurzläufer
- differenzierte Steuerung von mehreren TP-Anwendungen durch Aufteilung in mehrere Kategorien

Die Arbeitsweise von PCS kann über systemglobale und/oder kategoriespezifische Parameter eingestellt werden:

Kategoriebezogene Steuerparameter

Mit SERVICE-QUOTA (S-Q) wird das Leistungsvermögen des Servers prozentual auf die Kategorien verteilt. Durch SERVICE-QUOTA-MIN und SERVICE-QUOTA-MAX wird ein Rahmen vorgegeben, innerhalb dessen PCS den SERVICE für die angegebene Kategorie plant (S-Q-PLN). Schöpft eine Kategorie ihren Leistungsanteil nicht voll aus, dann steht der Rest anderen Kategorien zur Verfügung.

Über die Dehnung einer Kategorie (Parameter REQUEST-DELAY) erkennt PCS, wie stark die Tasks dieser Kategorie durch die Konkurrenz mit anderen Tasks verzögert werden. Die Dehnung ist also ein Maß für die Belastung und gibt an, um welchen Faktor sich die Laufzeit eines Auftrags verlängert, wenn er – bedingt durch Multiprogrammbetrieb – auf Betriebsmittelzuteilung warten muss.

$$\text{Dehnung} = \text{Laufzeit im Multiprogrammbetrieb} / \text{Allein-Laufzeit}$$

Bei gleichzeitigem Einsatz von SM2 und mitlaufendem SM2-Messprogramm SYSTEM für alle Platten, greift PCS periodisch die aktuelle Ein-/Ausgabezeit pro Kategorie ab, wodurch die Genauigkeit der Dehnungsberechnung entsprechend erhöht wird.

Innerhalb eines vorgegebenen Dehnbereichs (REQUEST-DELAY-MIN, REQUEST-DELAY-MAX) passt PCS, abhängig von der aktuellen Belastung (REQUEST-DELAY-ACTUAL), den S-Q-PLN-Wert der Kategorie dynamisch an.

Die SERVICE-Zuteilung auf die einzelnen Kategorien geschieht nach folgender Hierarchie:

1. Kategorien mit Dehnbereich und max. Leistungsanteil 100%

Diese Kategorien erhalten vor allen anderen Kategorien die Leistung zugewiesen („Planwerte“), die ihnen aufgrund der aktuellen Dehnung zusteht.

Fehlkapazitäten:

Gibt es mehrere derartige Kategorien und überschreitet die Summe ihrer Planwerte 100%, so werden die Planwerte im Verhältnis so reduziert, dass ihre Summe 100% beträgt.

Restkapazitäten:

Beträgt die Summe der Planwerte weniger als 100%, so wird die restliche Kapazität für Kategorien mit und ohne Dehnbereich verplant.

Sind die Tasks der Kategorien mit Dehnbereich und max. Leistungsanteil 100% nicht in der Lage, die geplante Kapazität zu verbrauchen, so wird die (aktuelle) Restkapazität an die anderen Kategorien mit und ohne Dehnbereich weitergegeben.

(Ausnahme: volle globale Antwortzeit-Orientierung)

2. Kategorien mit Dehnbereich und max. Leistungsanteil unter 100%

Kategorien mit Dehnbereich erhalten vor Kategorien ohne Dehnbereich die Leistung zugewiesen, die Kategorien mit Dehnbereich und max. Leistungsanteil 100% übrig lassen und die ihnen davon aufgrund ihrer aktuellen Dehnung zusteht.

Fehlkapazitäten:

Gibt es mehrere Kategorien mit Dehnbereich und überschreitet die Summe ihrer Planwerte die noch verfügbare Leistung, so werden die Planwerte im Verhältnis so reduziert, dass ihre Summe der verfügbaren Leistung entspricht.

Restkapazitäten:

Beträgt die Summe der Planwerte weniger als die verfügbare Leistung, so wird der Rest für Kategorien ohne Dehnbereich verplant.

Sind die Tasks der Kategorien mit Dehnbereich nicht in der Lage, die geplante Leistung zu verbrauchen, so werden die (aktuellen) Restkapazitäten den Kategorien ohne Dehnbereich verfügbar gemacht.

(Ausnahme: volle globale Antwortzeit-Orientierung)

3. Kategorien ohne Dehnbereich

Kategorien ohne Dehnbereich erhalten die Leistung zugewiesen, die Kategorien mit Dehnbereich übrig lassen. Diese Leistung wird im Verhältnis der maximal vorgesehenen Leistungsanteile auf die Kategorien ohne Dehnbereich verteilt.

Um die Regelungsfunktionen von PCS voll ausschöpfen zu können, sollte stets mindestens eine Kategorie ohne Dehnbereich vorhanden sein.

Zur besseren Differenzierung von Lastanforderungen mit hohem Betriebsmittelbedarf (Langläufern) gegenüber Lastanforderungen mit geringem Betriebsmittelbedarf (Kurzläufern) dient die Funktion „Automatischer Kategoriwechsel“. Dabei gibt man über die Größe DURATION an, wieviele SERVICE UNITS pro Bearbeitungsvorgang maximal verbraucht werden dürfen, bevor ein automatischer Wechsel in die „leistungsschwächere“ Nachfolgekategorie (NEXT CATEGORY) erfolgen soll.

SERVICE UNITS sind die gewichtete Summe der Arbeit der Betriebsmittel CPU, IO und MEMORY.

Durch den Parameter THROUGHPUT-QUOTA wird ein prozentuales Verhältnis bezüglich Antwortzeit- und Durchsatz-Optimierung innerhalb der Kategorie festgelegt. Die Auswirkungen sehen folgendermaßen aus:

THROUGHPUT-QUOTA	Antwortzeit-Optimierung	Durchsatz-Optimierung
	voll	voll
0%	50%	100%

Für Server mit TP- bzw. Dialogbetrieb ist die Antwortzeit-Optimierung wesentlich. In den entsprechenden TP-Kategorien sowie der Dialog-Startkategorie (Kurzläufer) empfiehlt sich der Einsatz von THROUGHPUT-QUOTA=0.

Bei Servern mit vorrangigem Batch-Betrieb ist ein möglichst großer Durchsatz, verbunden mit hoher Auslastung der Betriebsmittel, das wichtigste Leistungsziel. Für durchsatzorientierte Kategorien sind THROUGHPUT-QUOTA-Werte > 50% vorzusehen.

Systemglobale Steuerungsparameter

Die systemglobale Dehnung (REQUEST-DELAY-MAX) stellt einen Schwellwert für die Dehnung aller Tasks, die einer Kategorie mit Dehnbereich angehören, dar. Bei Überschreitung tritt eine Lastbegrenzung ein.

Mit THROUGHPUT-QUOTA wird das prozentuale Aufteilungsverhältnis zwischen Antwortzeit- und Durchsatzoptimierung, bezogen auf das Gesamtsystem, eingestellt. Der Parameter wirkt sich in den Grenzwerten sehr stark aus:

- THROUGHPUT-QUOTA=0
bewirkt harte Antwortzeit-Optimierung. Dies bedeutet, dass im Überlastfall (bei Überschreitung des systemglobalen REQUEST-DELAY-MAX-Wertes) Hintergrundkategorien vollständig verdrängt werden. Als Folge kann die Auslastung der Betriebsmittel und damit verbunden der Durchsatz absinken.
- THROUGHPUT-QUOTA=100
wird nur für reinen Batch-Betrieb eingesetzt.

Vorgehensweise beim Einsatz von PCS

Grundsätzlich sollte man vor dem ersten Einsatz die Hardware-Konfiguration (eventuelle Unterkonfiguration gegenüber Lastanforderungen) und die Software-Konfiguration (Lage und Größe der Systemdateien TSOSCAT, Seitenwechselbereiche und SYSEAM) überprüfen.

Je nach Lastfall sollten zunächst die vordefinierten STANDARD-Options

- STD#TP (vorrangiger TP-Betrieb),
- STD#DIA (vorrangiger Dialogbetrieb) und
- STD#BATCH (vorrangiger Batch-Betrieb)

verwendet und die damit erzielten Leistungen vermessen werden (für Einzelheiten zu den STANDARD-Options siehe Handbuch „PCS“ [23]). Gegenüber PRIOR ist dabei ein mindestens gleich gutes, i.d.R. aber besseres Leistungsverhalten zu erwarten. Ergibt die Überprüfung, dass in einzelnen Kategorien Leistungsziele nicht erreicht worden sind, dann muss für diese Kategorien die SERVICE-Planung erhöht werden. Als wesentliches Kriterium ist die aktuelle Dehnung REQUEST-DELAY-ACT zu betrachten (Information mit /SHOW-PCS-OPTION CATEGORY-NAME=*ALL bzw. Report „Request delay for category“ der Reportgruppe PCS von SM2).

Für den Transaktions- und Dialogbetrieb empfiehlt sich folgendes schrittweises Vorgehen, um die SERVICE-Vergabe an die betroffenen Kategorien zu erhöhen:

1. Liegt die aktuelle Dehnung zwischen REQUEST-DELAY-MIN und REQUEST-DELAY-MAX, so ist zunächst REQUEST-DELAY-MAX für die entsprechenden Kategorien heruntersetzen. Dadurch steigt der Wert für SERVICE-QUOTA-PLAN an. Der optimale Arbeitspunkt ist dann erreicht, wenn REQUEST-DELAY-ACT gering unter REQUEST-DELAY-MAX liegt.
2. Der SERVICE-QUOTA-MAX-Wert für die entsprechende Kategorie ist zu erhöhen. Empfohlen wird das Verhältnis $S-Q-MAX / S-Q-MIN \approx 3$
3. Der SERVICE-QUOTA-MIN-Wert wird erhöht. Er bewirkt insbesondere eine Reservierung für den Fall plötzlicher Laständerungen.

Bringen diese Maßnahmen nicht den gewünschten Erfolg in Richtung kürzerer Antwortzeiten, dann sind – abhängig von der vorrangigen Betriebsart – weitere Tuning-Maßnahmen erforderlich.

Bei Vorrang des Dialogbetriebs können Langläufer-Transaktionen über folgende Maßnahmen zurückgedrängt werden:

- a) S-Q-MAX reduzieren für die Folgekategorie (Dialog 1)
- a) S-Q-MIN reduzieren für die Folgekategorie (Dialog 1)

Bei Vorrang von TP-Betrieb kann zunächst – über die Verringerung S-Q-MAX und S-Q-MIN für die Kategorie DIALOG – die SERVICE-Planung für die Hintergrundlast Dialog reduziert werden.

Befinden sich in der Kategorie TP mehrere voneinander abhängige Tasks (z.B. UTM/UDS, UTM/SESAM), dann kann zusätzlich die Priorität für diejenigen Tasks mit dem größeren SERVICE-Bedarf erhöht werden.

Sind in der Kategorie TP mehrere Applikationen vorhanden, können die TP-Antwortzeiten generell durch Erhöhung der SERVICE-Planung verbessert werden. Für eine gezielte Verbesserung einzelner Anwendungen empfiehlt es sich, jede Anwendung in einer eigenen TP-Kategorie zu führen.

Soll gewährleistet sein, dass einer Kategorie ggf. die Gesamtleistung des Servers zugeteilt wird, so ist für sie der Wert SERVICE-QUOTA-MAX=100 vorzusehen. Mehreren Kategorien diesen Wert zu geben, ist nicht sinnvoll.

Alle hier aufgeführten Änderungen an den PCS-Standardinstellungen sind von der Systemverwaltung mit dem Dienstprogramm PCSDEFINE durchzuführen. Nachdem PCS als Subsystem gestartet worden ist, können mit Systemkommandos Informationen über PCS ausgegeben sowie dynamische Änderungen an Steuerparametern vorgenommen werden (Details hierzu siehe Handbuch „PCS“ [23]).

Ergänzungen zu PCS ab V2.7:

- Insbesondere um einzelnen Tasks „ausnahmsweise“ eine höhere Service-Zuteilung zukommen zu lassen, wurde das BS2000-Kommando `MOVE-TASK-TO-CATEGORY` eingeführt. Damit können Tasks in eine andere (in JMU definierte) Kategorie versetzt werden.
- Mit dem Kommando `MODIFY-TASK-CATEGORY`, Operand `IO-PRIORITY` können alle Tasks einer Kategorie an die Prioritätssteuerung von überlasteten Platten-Volumes durch IORM angeschlossen werden.

Beispiele für die Wirkung der PCS-Parameter

Beispiel 1

Wirkung der Standard-Option STD#DIA bei vorrangigem Dialogbetrieb zur Programm-entwicklung mit einer Batch-Last im Hintergrund.

Parameter der STANDARD-Option STD#DIA:

Kategorie	SERVICE-QUOTA		REQUEST-DELAY		DURATION	NEXT CATEGORY	THROUGH-PUT-QUOTA
	MIN	MAX	MIN	MAX			
TP	0	30	1	3	-	-	-
DIALOG	20	40	2	4	500	DIALOG1	-
DIALOG1	10	30	2	5	2000	DIALOG2	10
DIALOG2	0	50	-	-	-	-	20
BATCH	0	20	-	-	-	-	70
BATCHF	0	20	1	3	-	-	-

Alle ankommenden Dialog-Transaktionen werden zunächst von der Standard-Kategorie DIALOG aufgenommen. Kurze Dialoge, wie z.B. EDT-Anweisungen, benötigen i.d.R. weniger als 200 SERVICE UNITS und verbleiben somit bis zum Transaktionsende in der Kategorie DIALOG.

Ist der SERVICE-Bedarf größer als 500 SERVICE UNITS (z.B. bei Dateioperationen), so erfolgt automatisch der Wechsel in die „leistungsschwächere“ Kategorie DIALOG1.

Compilierungen und Programmtests mit einem SERVICE-Bedarf größer 2.000 SERVICE UNITS wechseln in die Kategorie DIALOG2 über (keine REQUEST-DELAY-Angabe = keine Antwortzeitanforderung) und laufen dort zu Ende.

Gegenüber der durchsatzorientierten Kategorie BATCH (THROUGHPUT-QUOTA=70) für die Batch-Last im Hintergrund, erfolgt eine Bevorzugung der Dialog-Langläufer in der Kategorie DIALOG2 durch den größeren SERVICE-QUOTA-MAX-Wert.

In Ausnahmefällen ist für die Durchführung besonders wichtiger Batch-Läufe die Kategorie BATCHF (BATCH FAST) einzusetzen.

Der Einsatz von PCS bringt gegenüber einem gut eingestellten PRIOR eine wesentliche Verbesserung der Antwortzeiten für kurze Dialoge. Die Antwortzeiten für Dialog-Transaktionen mit einem hohen SERVICE-Bedarf werden verlängert.

Beispiel 2

Wirkung der Standard-Option STD#TP bei vorrangigem TP-Betrieb mit einer Dialog- und Batch-Last im Hintergrund.

Parameter der STANDARD-Option STD#TP:

Kategorie	SERVICE-QUOTA		REQUEST-DELAY		DURATION	NEXT CATEGORY	THROUGH-PUT-QUOTA
	MIN	MAX	MIN	MAX			
TP	50	100	1	3	-	-	-
DIALOG	10	30	2	6	500	DIALOG1	-
DIALOG1	0	30	-	-	-	-	10
BATCH	0	10	-	-	-	-	70
BATCHF	0	20	1	3	-	-	-

Kategorien mit Angabe einer Dehnung (REQUEST-DELAY) werden bevorzugt vor Kategorien ohne Dehnung.

Übersteigt die Summe der SERVICE-QUOTA-MAX-Werte der Kategorien mit Dehnung 100%, so erfolgt eine entsprechende Normierung, d.h. die relative Konkurrenz der Kategorien zueinander bleibt erhalten.

Durch die explizite Angabe von SERVICE-QUOTA-MAX=100 für TP wird sichergestellt, dass im Bedarfsfall die Hintergrundlast vollständig verdrängt wird.

Sämtliche ankommenden Dialog-Transaktionen gelangen zunächst in die Kategorie DIALOG. Übersteigt der Service-Bedarf 500 SERVICE UNITS, so erfolgt automatisch der Wechsel in die Kategorie DIALOG1, in der keine Antwortzeitanforderungen gestellt werden.

Die Bevorzugung der aufwändigeren Dialog-Transaktionen in DIALOG1 gegenüber dem echten Batch in der Kategorie BATCH erfolgt über den höheren SERVICE-QUOTA-MAX-Wert (=30).

In Ausnahmefällen ist für die Durchführung besonders wichtiger Batch-Läufe die Kategorie BATCHF (BATCH FAST) einzusetzen.

Bei vorrangigem TP-Betrieb führt PCS im Vergleich zu PRIOR zu einer starken Verkürzung der Antwortzeiten der TP-Anwendungen. Auch die Kurzläufer der Hintergrundlast Dialog werden besser bedient, während die Langläufer stark zurückgedrängt werden. Zufriedenstellende TP-Antwortzeiten im Hochlastfall sind unter PRIOR mit variablen Prioritäten nicht erreichbar. Bei Steuerung mit PCS sind alle Leistungsverbesserungen auf einfache Weise mit einem **einzigem Steuerparametersatz** zu erreichen.

Beispiel 3

Steuerung von mehreren TP-Anwendungen

Für eine gezielte Steuerung von TP-Anwendungen wird jede Anwendung oder mehrere gleichrangige Anwendungen in einer eigenen Kategorie geführt. Mit der nachfolgenden Option werden 3 TP-Anwendungen und eine Dialog- und Batch-Last für die Restkapazität gesteuert. Die 3 TP-Anwendungen sind in ihrer Rangfolge den Kategorien TP, TP2 und TP3 zugeordnet.

Die Kategorien TP2 und TP3 müssen mit dem Task-Attribut TP als eigene Kategorien mit der Anweisung DEFINE-JOB-CLASS (siehe Handbuch „Dienstprogramme“ [6]) definiert sein.

PCS-Steuerparameter:

Kategorie	SERVICE-QUOTA		REQUEST-DELAY		DURATION	NEXT CATEGORY	THROUGH-PUT-QUOTA
	MIN	MAX	MIN	MAX			
TP	20	40	1,0	2,5	-	-	-
TP2	15	25	1,0	3,0	-	-	-
TP3	10	15	1,0	3,5	-	-	-
DIALOG	5	10	2,0	5,0	300	DIALOG1	10
DIALOG1	0	30	-	-	0	-	20
BATCH	0	20	-	-	-	-	70
BATCHF	0	5	1,0	4,0	-	-	-

Kategorien mit Angabe einer Dehnung (REQUEST-DELAY) werden bevorzugt vor Kategorien ohne Dehnung. Alle TP-Transaktionen werden aufgrund der besseren Werte für SERVICE-QUOTA und REQUEST-DELAY gegenüber Dialog-Transaktionen vorrangig bedient.

Für die TP-Kategorien wird insgesamt maximal 80% des Leistungsvermögens des Servers geplant (SERVICE-QUOTA-MAX für TP, TP2 und TP3).

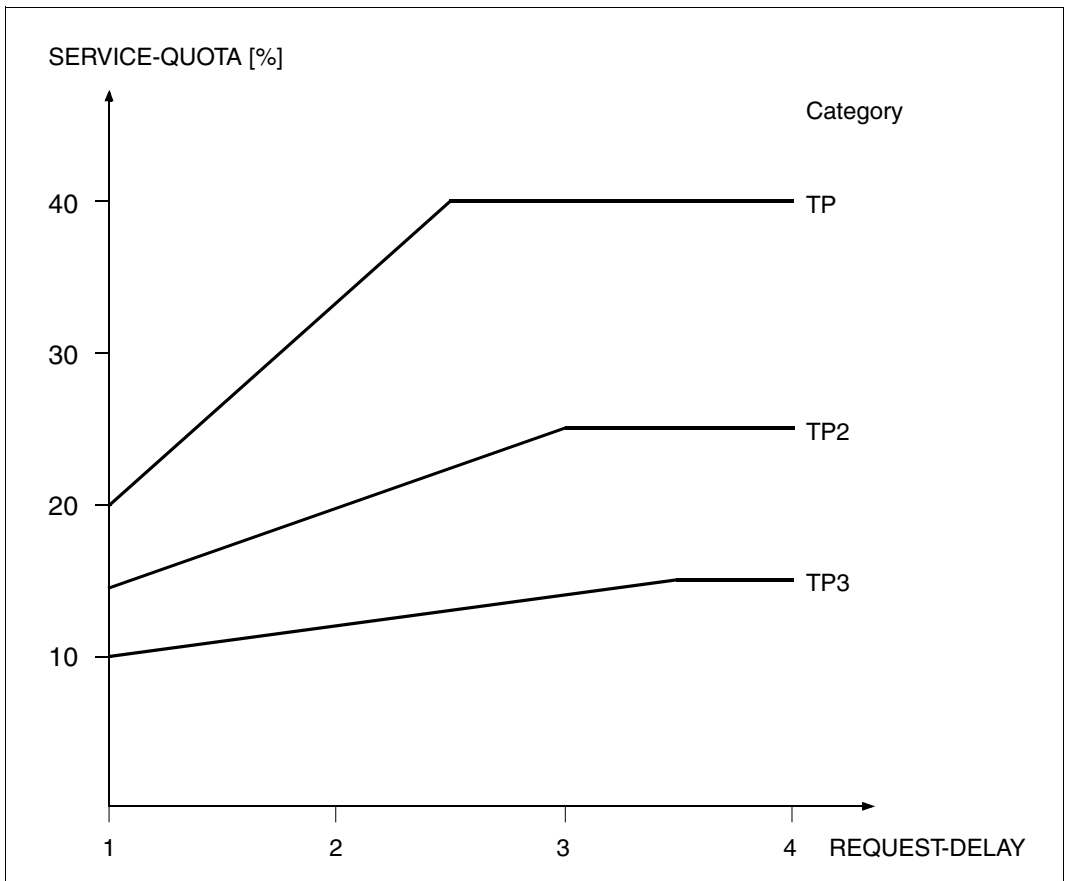


Bild 11: Anteil der TP-Anwendungen am Leistungsvermögen abhängig von der aktuellen Dehnung

Das [Bild 11](#) zeigt den Anteil am Leistungsvermögen abhängig von der aktuellen Dehnung. Unter wechselnder Belastung (ausgedrückt durch die Dehnung) wird ein wechselnder Anteil am Leistungsvermögen geplant.

Für Anwendungen der Kategorie TP wird bereits bei einer Dehnung von 2,5 (REQUEST-DELAY-MAX) der Wert von SERVICE-QUOTA-MAX (40%) geplant. Liegt der aktuelle Wert der Dehnung zwischen 1 und 2,5, dann wird ein SERVICE zwischen 20% und 40% zugeteilt.

Die Anwendungen der Kategorien TP2 und TP3 werden aufgrund der geringeren Werte für SERVICE-QUOTA und REQUEST-DELAY mit einem entsprechend niedrigeren Anteil am Leistungsvermögen bedient.

Die ankommenden Dialog-Transaktionen werden in der Kategorie DIALOG gestartet. Ist der SERVICE-Bedarf größer als 300 SERVICE UNITS, werden die Dialoge in die leistungsschwächere Kategorie DIALOG1 verdrängt. Durch die Angabe des Werts 10 bei THROUGHPUT-QUOTA für die Kategorie DIALOG wird die Aktivierungspriorität für Dialog-Transaktionen verringert.

Die aufwändigeren Dialog-Transaktionen in der Folgekategorie DIALOG1 werden gegenüber dem echten Batch in der Kategorie BATCH über den höheren SERVICE-QUOTA-MAX-Wert bevorzugt.

In Ausnahmefällen ist für die Durchführung besonders wichtiger Batch-Läufe die Kategorie BATCHF (BATCH FAST) einzusetzen.

Bei größeren Servern mit sehr vielschichtigen Arbeitslasten, rasch wechselnden Leistungsanforderungen und vielen Tasks in Konkurrenz, kann nur noch mit Einsatz von PCS das Leistungsverhalten mit den vorgegebenen Performance-Zielen in Einklang gebracht werden. Je heterogener sich die Arbeitslast zusammensetzt, je größer die Anzahl Tasks und je unterschiedlicher der Betriebsmittelbedarf der einzelnen Lastanteile ist, desto besser kann PCS seine Wirkung entfalten.

8.2.3 Einführung in das TANGRAM-Konzept

Ziel des TANGRAM-Konzepts (Task and Group Affinity Management) ist die bessere Nutzung der Hardware-Leistung von Multiprozessoren.

Zur Überbrückung des Unterschiedes zwischen CPU-Geschwindigkeit und der Zugriffszeit zum Hauptspeicher verwenden alle Server schnelle Pufferspeicher (Caches). Abhängig vom Befehlsprofil (im Wesentlichen vom Anteil der Hauptspeicher-Zugriffe) hängt die erreichbare Leistung stark von der Trefferrate im Cache ab.

Es gibt zwei Arten von Caches, die meist kombiniert werden:

- Store-Through-Caches (STC)
Die Daten werden so rasch wie möglich „durchgereicht“, d.h. nach einem Schreibvorgang werden die Daten im Hauptspeicher sofort aktualisiert.
- Store-In-Caches (SIC)
Die Daten verbleiben im Cache und werden nach einer LRU-Strategie (Least Recently Used) in den Hauptspeicher zurückgeschrieben. In diesem Fall sind die im Hauptspeicher stehenden Daten nicht aktuell.

Während bei Monoprozessoren die volle Leistung durch einen ausreichend großen Cache erreicht werden kann, ist die Lage bei Multiprozessoren wesentlich komplexer.

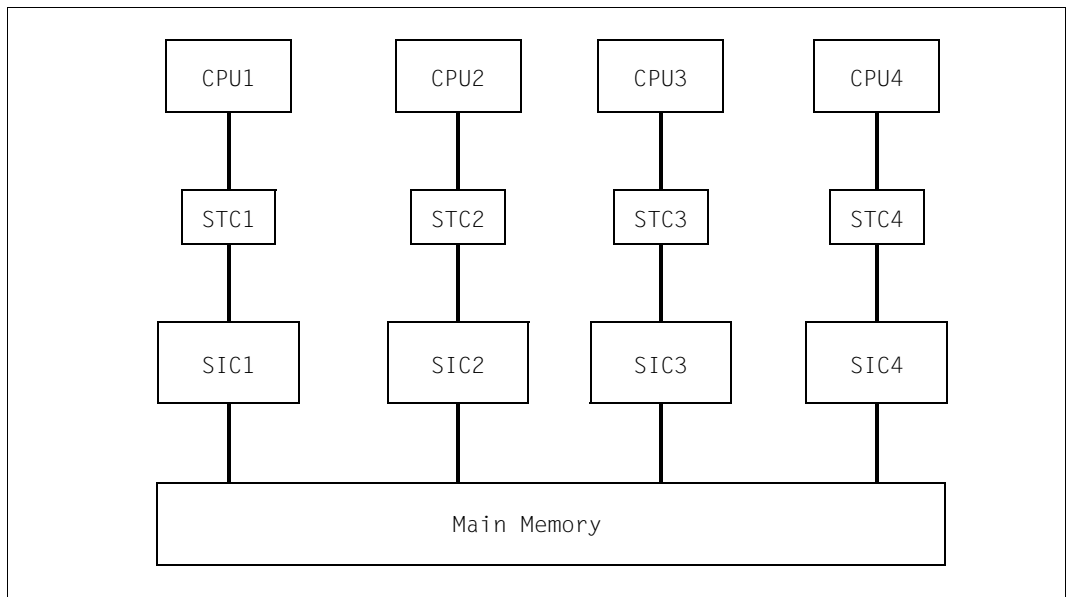


Bild 12: Cache-Hierarchie bei einem Quadroprozessor mit First-Level Store-Through-Caches und Second-Level Store-In-Caches

Je mehr Tasks, die auf verschiedenen CPUs laufen, auf die gleichen Hauptspeicher-Bereiche (z.B. Memory-Pools) zugreifen, desto größer ist die Wahrscheinlichkeit, dass die aktuelle Information sich nicht im CPU-eigenen Cache (SIC), sondern in einem „Nachbar“-Cache befindet.

Damit die CPU weiterarbeiten kann, muss die aktuelle Information erst in den eigenen Cache gebracht werden, wodurch die für den Anwender nutzbare CPU-Leistung sinkt. Die Lage verschärft sich, je größer der Schreibanteil auf die gleichen Hauptspeicher-Seiten wird.

Das TANGRAM-Konzept versucht, diesem Leistungsverlust durch die Bildung von Taskgruppen und deren Zuordnung zu bestimmten CPUs (möglichst einer Untermenge der vorhandenen physikalischen CPUs) entgegenzuwirken.

Bildung von Taskgruppen

Zu einer Taskgruppe werden sogenannte „affine Tasks“ zusammengefasst, die eine große Menge von gemeinsamen Daten schreibend referenzieren.

Die Anmeldung zu einer Taskgruppe erfolgt mit dem Makro TINF. Die Verwaltung der Taskgruppen übernimmt das bei „System Ready“ automatisch gestartete Subsystem TANGBAS.



Die Softwareprodukte UTM, UDS und SESAM/SQL führen diesen TINF-Aufruf beim Hochfahren pro Applikation bzw. DB-System automatisch aus. Die Tasks einer UTM-Applikation bilden also eine eigene Taskgruppe.

Zuordnung zu CPUs

Die Zuordnung von Taskgruppen zu CPU-Gruppen erfolgt dynamisch abhängig von der Last, wenn ein längerer Zeitraum betrachtet wird.

Für ein kurzes Zeitintervall (Parameter PERIOD, Default-Wert: 10 Sekunden) dagegen herrscht eine feste Zuordnung, um die erwähnte Cache-Problematik zu entschärfen.

Der **Zuordnungs-Algorithmus** des Subsystems TANGRAM, welches explizit gestartet werden muss, misst periodisch folgende Werte:

- CPU-Verbrauch jeder einzelnen Taskgruppe
- Auslastung der einzelnen CPUs
- CPU-Gesamtauslastung

Anschließend wird eventuell eine neue Zuordnung der Taskgruppen zu den CPU-Gruppen unter folgenden Randbedingungen vorgenommen:

- Die Taskgruppe muss eine gewisse CPU-Mindestleistung aufnehmen (Parameter THRESHOLD, Default-Wert: 10%), um bei der individuellen CPU-Zuordnung berücksichtigt zu werden.
- Lastet eine Taskgruppe einen oder mehrere CPUs weitgehend aus (Parameter CLEARANCE, Default-Wert: 20%, entspricht einer CPU-Auslastung von 80%), so erhält sie jeweils eine CPU mehr zugeordnet, um eine Laststeigerung dieser Taskgruppe im nächsten Zuteilungsintervall nicht zu unterbinden.
- Die Zuteilung der Taskgruppen zu CPUs erfolgt so, dass alle CPUs möglichst gleichmäßig ausgelastet sind.

Der **Scheduling-Algorithmus** von PRIOR überprüft bei jedem Taskwechsel, ob eine Task entsprechend der vom Zuordnungs-Algorithmus getroffenen Zuordnung auf der CPU ablaufen darf. Dabei wird die Strategie „IDLE vor Fremd“ verfolgt, d.h. es wird der Leerlauf einer CPU in Kauf genommen, bevor eine fremde Task initiiert wird (die den Cache-Inhalt komplett neu aufbauen müsste).

Nutzen von TANGRAM

Die bessere Nutzung der Hardware-Leistung bei Multiprozessoren hängt von folgenden Bedingungen ab:

- Multiprozessorgrad (höherer Gewinn bei mehr CPUs)
- Hardware-Architektur (große Store-In-Caches)
- Anwendung (Größe des Schreibanteils, Aufteilungsmöglichkeit auf Teilmenge der vorhandenen CPUs)
- Auslastung (merkbarer Gewinn erst bei höherer Auslastung)

Eine allgemeine Aussage über den Nutzen von TANGRAM ist schwierig. Messungen mit verschiedenen Lasten zeigten ein Mehr an verfügbarer Hardware-Leistung von ca. 5% bei Bi-Prozessoren und ca. 10% bei Quadro-Prozessoren.

8.3 Data-Management

Verwaltung der Datenträger und Zugriffswege

Die Geräteverwaltung NDM (Nucleus Device Management) besteht im Wesentlichen aus folgenden Komponenten:

- **Betriebsmittelbelegung**
Sie steuert und überwacht die logischen Zustände – belegt oder frei – der vom Anwender angeforderten Betriebsmittel (Geräte, Datenträger).
- **Betriebsmittelreservierung**
Sie bearbeitet die Anforderungen zur Reservierung der Betriebsmittel. Diese Anforderungen formulieren die Anwender mit `/SECURE-RESOURCE-ALLOCATION`.
- **Rekonfiguration**
Sie ermöglicht dem Operator das Zu- und Wegschalten von Hardware-Einheiten und deren logischen Verbindungen auf Kommandoebene.
- **Datenträgerüberwachung**
Sie überwacht die Zugriffsbereitschaft der Datenträger, insbesondere bei Montiervorgängen.

Bezüglich des Systemaufwands ist bei der Belegung der Betriebsmittel folgendes zu beachten:

Jede Belegung eines Volumes wird in ihrem Standard-Volume-Label (SVL) hinterlegt. Bei Private Volumes ergibt sich abhängig von der Belegungsart (`task-exclusive` oder `task-shareable`) und vom Belegungszeitpunkt (`ASSIGN-TIME=*USER/*OPERATOR`) ein unterschiedlicher Systemaufwand.

- Bei Bestimmung des Belegungszeitpunkts durch den Anwender und `task-exclusive` Belegung mit `/SET-DISK-PARAMETER VOLUME=vsn,ASSIGN-TIME=*USER, USER-ALLOCATION=*EXCLUSIVE` erfolgt bei jeder Dateioperation mit Katalogzugriff (`/COPY-FILE, /DELETE-FILE, /CREATE-FILE, /MODIFY-FILE-ATTRIBUTES, OPEN, CLOSE`) zu Beginn und am Ende eine Aktualisierung des SVLs auf dem Volume.
- Bei Bestimmung des Belegungszeitpunkts durch den Anwender und `task-shareable` Belegung mit `/SET-DISK-PARAMETER VOLUME=vsn,ASSIGN-TIME=*USER, USER-ALLOCATION=*SHARE` wird beim Zugriff des ersten Anwenders das SVL gelesen und bei der Freigabe des Volumes durch den letzten Anwender das SVL zurückgeschrieben. Zwischendurch erfolgt also keine Aktualisierung des SVLs auf dem Volume, wohl aber eine interne Tabellenverwaltung bei jeder Dateioperation mit Katalogzugriff.
- Bei Bestimmung des Belegungszeitpunkts durch das System (Einschalt-Interrupt) erfolgt die SVL-Aktualisierung einmal pro Session, unabhängig von der Belegungsart.

Empfehlungen

- Die Bestimmung des Belegungszeitpunkts sollte durch das System erfolgen.

Eingestellt wird dies mit `/SET-DISK-PARAMETER VOLUME=vsn, ASSIGN-TIME=*OPERATOR, USER-ALLOCATION=*EXCLUSIVE/*SHARE`

Die Freigabe des Volumes (Sonderfall) kann durch Übergang auf `/SET-DISK-PARAMETER VOLUME=vsn, ASSIGN-TIME=*USER` erreicht werden.

- Bei Shared-Private-Volumes sollte der Parameter `SYSTEM-ALLOCATION=*SHARE` in `/SET-DISK-PARAMETER` nur dann verwendet werden, wenn tatsächlich mehrere schreibberechtigte Systeme auf dieses Volume simultan zugreifen.

Wie bereits erwähnt (siehe [Abschnitt „Logische Betriebsarten von Platten“ auf Seite 145](#)), ist der Koordinierungsaufwand bei Dateioperationen mit Katalogzugriff sehr hoch.

Sobald im laufenden Betrieb nur noch **ein** System auf ein Shared-Private-Volume zugreift, sollte folgendes Kommando eingegeben werden:

`/SET-DISK-PARAMETER VOLUME=vsn, SYSTEM-ALLOCATION=*EXCLUSIVE`

9 Systeminterne Abläufe

Zur Erleichterung der Interpretation von Messwerten werden in den folgenden Abschnitten die internen Abläufe im Betriebssystem BS2000, soweit sie für Durchsatz- und Antwortzeitverhalten interessant sind, näher erläutert und verschiedene Richtwerte für die Auslastung der Betriebsmittel angegeben.

9.1 Unterbrechungsanalyse (Funktionszustände)

BS2000/OSD kennt vier Funktionszustände. Davon sind für den normalen Betrieb die Zustände TU, TPR und SIH wesentlich. Im Zustand MEH werden lediglich Maschinenfehler bearbeitet.

Es bedeuten:

TU Task Unprivileged
TPR Task PRivileged
SIH System Interrupt Handling
MEH Machine Error Handling

Die User-Task (das Anwenderprogramm) läuft im Funktionszustand TU.

Systemroutinen, die der User-Task direkt zugeordnet werden können (z.B. SVC: RDATA), werden im Funktionszustand TPR ausgeführt (siehe [Bild 13 auf Seite 240](#)).

Die aufgenommene CPU-Zeit im Funktionszustand TU und TPR ist als produktive Zeit im Sinne des Anwenders zu sehen.

Im Funktionszustand SIH laufen die Basissteuermechanismen des BS2000. Außer nach MEH ist der Funktionszustand SIH nicht unterbrechbar.

System-Tasks führen unterstützende organisatorische Aufgaben durch und laufen im Funktionszustand TPR (siehe [Bild 14 auf Seite 240](#)).

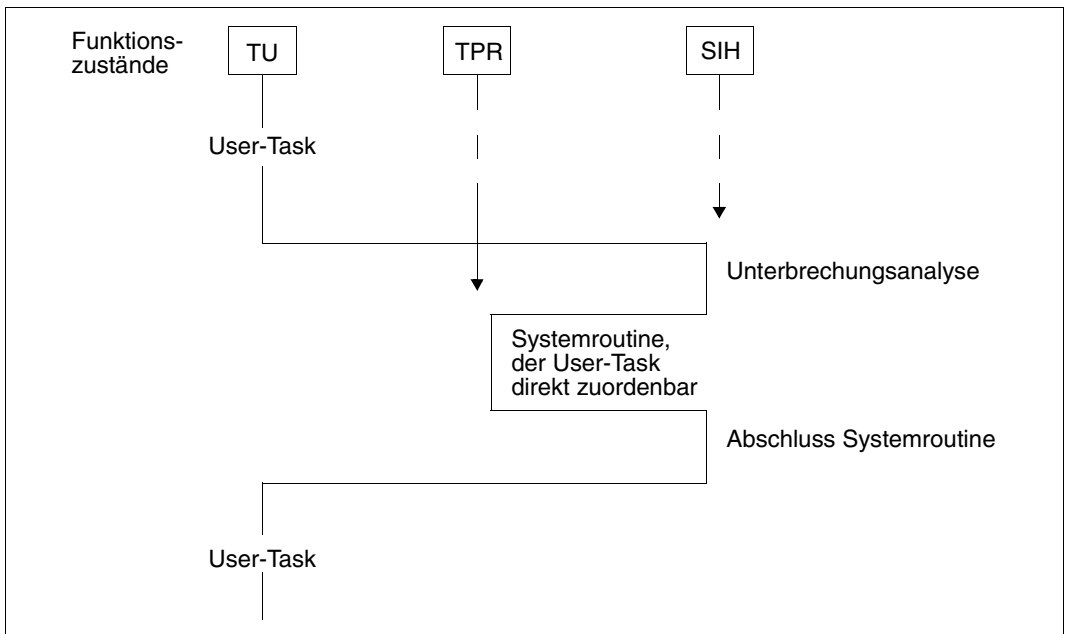


Bild 13: BS2000-Funktionszustände (User-Task)

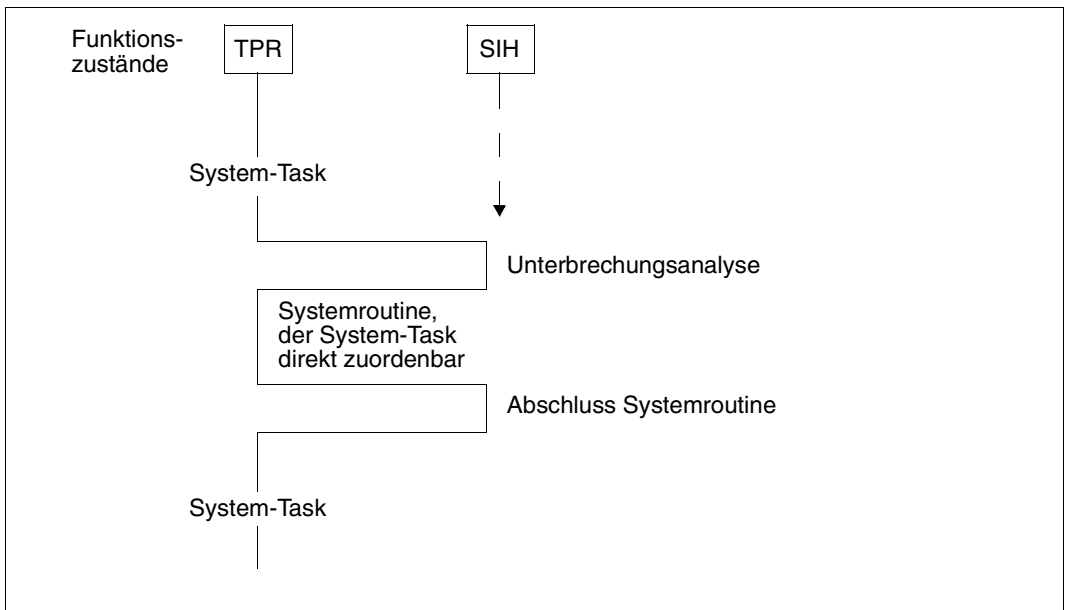


Bild 14: BS2000-Funktionszustände (System-Task)

Eine Unterbrechung tritt entweder dadurch auf, dass eine User- oder System-Task die Unterstützung des Betriebssystems braucht oder, dass ein Ereignis eintritt, das vom Betriebssystem zu behandeln ist.

Es gibt fünf Unterbrechungsarten:

- SVC-Unterbrechung (= task-bezogene Unterbrechung)

Von einer User- oder System-Task werden bestimmte Systemdienste angefordert. Neben dem Aufwand für die Systemroutine werden auch die Unterbrechungsanalyse und der Abschluss der Systemroutine der aufrufenden Task angerechnet.

- Programm-Unterbrechung (= task-bezogene Unterbrechung)

Hierzu zählen die „echten“ Fehler, die zum Programmabbruch führen (Adressenfehler, Datenfehler, ...), die bei Durchsatzüberlegungen nicht weiter betrachtet werden sollen, und die Unterbrechungsursachen:

- Seite nicht im Speicher (Event-Code: 4C)
- Adressübersetzungsfehler (Event-Code: 48)

In beiden Fällen kann zunächst der Programmlauf nicht fortgesetzt werden, da die angesprochene Seite (Page) nicht unmittelbar zur Verfügung steht.

Beim Event-Code **4C** (Seite nicht im Speicher) sind zwei Fälle zu unterscheiden:

- Die Seite ist noch im Hauptspeicher, und zwar im Free-Pool (Read Only Queue bzw. Read Write Queue):

Die über längere Zeit nicht angesprochene Seite wurde durch das Paging-Management in den Free-Pool gebracht. Nach einer einfachen Seitenumkettung kann der Programmlauf fortgesetzt werden (Page-Reclaim). Die globalen Working-Set-Strategien bewirken, dass Page-Reclaims nur noch selten auftreten (siehe [Abschnitt „Hauptspeicher-Verwaltung“ auf Seite 246](#)). In den meisten Fällen handelt es sich um den erstmaligen Zugriff auf eine Seite (First-Page-Access).

- Die Seite ist nicht im Hauptspeicher und muss mit einer Paging-Ein-/Ausgabe in den Hauptspeicher gebracht werden (Page-Read).

Beim Event-Code **48** handelt es sich um einen echten Fehler bei der Adressübersetzung, der zum Programmabbruch führt.

- Zeitgeber-/externe Unterbrechung (= asynchrone Unterbrechung)

Signal durch den Intervallzeitgeber oder ETC (Elapsed Time Clock).

- Ein-/Ausgabe-Unterbrechung (= asynchrone Unterbrechung)

Signal, dass ein Ereignis bei der Ein-/Ausgabe vorliegt. (In den meisten Fällen handelt es sich um das Eintreten einer Ende-Bedingung bei der Durchführung der Ein-/Ausgabe.)

- Maschinenfehler-Unterbrechung (= asynchrone Unterbrechung)
Signal, dass ein Maschinenfehler aufgetreten ist.

Nach erfolgter Unterbrechungsbehandlung wird mit Ausnahme der SVC-Unterbrechung durch die Dispatcher-Routine (Auswahl-Routine) des Task-Initiators diejenige User- bzw. System-Task mit der höchsten Priorität ausgewählt, die bereit ist zu arbeiten, und ihr die Kontrolle übertragen (Task in control; siehe auch [Abschnitt „CPU-Verwaltung“ auf Seite 258](#)).

9.2 Erzeugen und Beenden von Tasks

BS2000/OSD kennt User-Tasks und System-Tasks.

User-Tasks

User-Tasks werden auf Anforderung der Anwender mit Hilfe von System-Tasks erzeugt. Sie untergliedern sich vom Task-Typ her in:

- Dialog-Tasks
(Task-Typ X'40', erzeugt bei /SET-LOGON-PARAMETERS durch die System-Task DIAA)
- Batch-Tasks
(Task-Typ X'20', erzeugt bei /ENTER-JOB durch die System-Task TSC)

Vom Task-Typ zu unterscheiden ist das Task-Attribut. Dieses beeinflusst die Leistungszuweisung im laufenden Betrieb. Entsprechend dem Einsatzzweck gibt es die Task-Attribute

- TP für Transaktionsanwendungen
- DIALOG für Dialoganwendungen
- BATCH für Batch-Anwendungen

Dialog- und Batch-Tasks erhalten automatisch das Task-Attribut DIALOG bzw. BATCH.

Tasks von Transaktionsanwendungen (DCAM, UTM, UDS, SESAM/SQL, ...) laufen unter dem Task-Typ, unter dem sie gestartet werden (als Batch- bzw. Dialog-Task).

Wird eine Job-Klasse definiert mit den Parametern `TP-ALLOWED=*YES(CATEGORY=xy)`, `START-ATTRIBUTE=*TP`, so erhalten diese Tasks als zusätzliche Kennzeichnung das Task-Attribut TP (siehe [Abschnitt „Job-Klassen“ auf Seite 197](#)).

Das Task-Attribut TP kann auch mit dem Makroaufruf TINF erlangt werden bei gleichzeitigem Wechsel in die Kategorie mit dem Standardnamen TP.

System-Tasks

System-Tasks (Task-Typ X'80') erhalten automatisch das Task-Attribut SYSTEM.

Es gibt folgende permanent vorhandene (preallocated) System-Tasks und wichtige, dynamisch erzeugte System-Tasks (siehe Handbuch „Systembetreuung“ [10]).

9.3 Verwalten der Betriebsmittel Hauptspeicher und CPU

Sowohl die User- als auch die System-Tasks werden von den Routinen des Task-Managements überwacht und gesteuert.

Damit eine Task zum Ablauf kommen kann, muss das Task-Management folgende Entscheidungen treffen:

- Zuteilung der Berechtigung zur Hauptspeicher-Nutzung → Aktivierung
- Zuteilung des Betriebsmittels CPU → Initiierung

Die Initiierung kann nur für bereits aktivierte Tasks erfolgen.

Kriterien für die Verwaltung des Betriebsmittels **Hauptspeicher**:

- Multiprogramming-Level (MPL) pro Kategorie
- Priorität
- Betriebsmittel-Auslastung (CPU, Hauptspeicher, Paging Aktivität)
- Geleistete Systemdienste (CPU-Zeit, Anzahl Ein-/Ausgaben)

Kriterium für die Verwaltung des Betriebsmittels **CPU** ist ausschließlich die Priorität

Ist bei Multiprozessoren die Funktion TANGRAM eingeschaltet, so kann der Zuordnungs-Algorithmus eventuell für eine andere Initiierungsreihenfolge sorgen (siehe auch [Abschnitt „Einführung in das TANGRAM-Konzept“ auf Seite 233](#)).

Interne Kategorie-Bezeichnung

Systemintern werden sowohl die Task-Kategorien als auch die Prioritäten etwas anders dargestellt als an der Benutzerschnittstelle.

Kategorien mit Standard-Kategorienamen haben feste Bezeichnungen:

Kategorie	Kategorie-Kennzeichen
SYS	0
DIALOG	1
BATCH	2
TP	3
xy	4
...	...

Interne Priorität

Für die vergebenen variablen Prioritäten erfolgt eine Gewichtung entsprechend der Wichtigkeit der Kategorie (repräsentiert durch den Parameter WEIGHT-CODE). Die variable interne Priorität ist eine Funktion

- der externen Priorität
- des Gewichts W (WEIGHT-CODE) der Kategorie, zu der die Task gehört
- der Summe S der Gewichte aller Kategorien

$$S = W_{\text{SYS}} + W_{\text{TP}} + W_{\text{DIAL}} + W_{\text{BATCH}} + W_{\text{xy}} + \dots$$

Variable interne Priorität =

$$1 + (256 - \text{externe Priorität}) * 0,5 * (1 + W / S)$$

Vom Anwender vergebene, feste externe Prioritäten werden folgendermaßen umgerechnet:

$$\text{Feste interne Priorität} = 256 - \text{externe Priorität}$$

Die interne Priorität ist die Basis für die Berechnung der Aktivierungs- und Initiierungs-Priorität.

9.3.1 Hauptspeicher-Verwaltung

Folgende Begriffe werden erläutert:

- Aktivierung (siehe [Seite 247](#))
Zuteilung der Berechtigung zur Hauptspeicher-Nutzung an eine arbeitswillige, ablaufbereite (inactive, ready) Task
- Deaktivierung (siehe [Seite 249](#))
Entzug des Rechts auf Hauptspeicher-Nutzung bei Erwartung langer Wartezeiten oder nach Ableistung bestimmter Systemdienste
- Zwangsdeaktivierung (siehe [Seite 250](#))
Zwangswise Deaktivierung einer Task aufgrund von Betriebsmittelüberlastungen (Forced Deactivation)
- Verdrängung (siehe [Seite 251](#))
Aktivierung einer Task auf Kosten der Deaktivierung einer anderen Task (Preemption)
- Kontrollfunktionen der Hauptspeicher-Verwaltung (siehe [Seite 252](#))
 - Activation Control Function (ACF)
 - Preemption Control Function (PCF)
 - Paging Load Control Function (PLC)
 - System Service Slice Runout Control
 - Waiting Time Runout Control
- Seitenverwaltungs-Algorithmus (siehe [Seite 256](#))
 - System Working Set-Verfahren (SYS-WS)

Alle oben genannten Begriffe, bis auf den Punkt „Seitenverwaltungs-Algorithmus“, sind nur im seltenen Fall eines Hauptspeicher-Engpasses von Bedeutung.

Aktivierung

Eine Aktivierung ist nur möglich, wenn von der Kontrollfunktion ACF (Activation Control Function) aufgrund der Messung der Auslastung der Betriebsmittel CPU und Hauptspeicher sowie der Paging-Aktivität (siehe „[Kontrollfunktionen der Hauptspeicher-Verwaltung](#)“ auf Seite 252) die Aktivierung erlaubt wird.

Vorrangiges Ziel ist es, den vom Anwender spezifizierten **MIN MPL-Wert** (MINIMUM-ACTIVE-TASKS) **in jeder Kategorie zu erreichen**.

Der Parameter WEIGHT-CODE beeinflusst den Aktivierungsvorgang.

Eine Index-Entscheidung (mit NAT (Number Active Tasks) = Anzahl der aktiven Tasks der Kategorie, für die der Index berechnet wird) erfolgt, wenn eine dieser drei Aussagen zutrifft:

- noch nicht alle Kategorien haben MIN MPL erreicht
- in jeder Kategorie ist MIN MPL überschritten, aber MAX MPL noch nicht erreicht
- in jeder Kategorie ist MAX MPL bereits überschritten

Der Kategorie-Index wird wie folgt berechnet:

$$\text{Index} = (\text{NAT} + 1 - N) / \text{WEIGHT}$$

mit:

$$N = 0, \quad \text{wenn } \text{MPL} < \text{MIN MPL}$$

$$N = \text{MIN MPL}, \quad \text{wenn } \text{MIN MPL} \leq \text{MPL} < \text{MAX MPL}$$

$$N = \text{MAX MPL}, \quad \text{wenn } \text{MAX MPL} \leq \text{MPL}$$

Die Aktivierung erfolgt für die Kategorie mit dem kleinsten Index. Von dieser Kategorie wird die Task mit der höchsten Aktivierungspriorität ausgewählt.



Ausnahme

Ist der MIN MPL-Wert der ermittelten Kategorie bereits erreicht und liegt eine Aktivierungsanforderung einer Task mit fester Priorität aus einer anderen Kategorie vor, so wird diese aktiviert, sofern die Kontrollfunktion ACF dies gestattet; andernfalls kommt es zu einer Verdrängung.

Tasks mit Systemprioritäten werden immer sofort aktiviert. Auch hier entscheidet die Kontrollfunktion ACF, ob dies im Rahmen einer Aktivierung möglich ist oder ob eine Verdrängung durchgeführt werden muss.

Aktivierungspriorität

Bei variablen externen Prioritäten hängt die Aktivierungspriorität von vier Faktoren ab:

- von der variablen internen Priorität
- vom Funktionszustand (TU oder TPR)
- vom Zeitpunkt t_A der letzten Aktivierung (es wird jeweils die Differenz Δ gebildet)
- von einem Konfigurationsfaktor C

Variable Aktivierungspriorität =
(Variable interne Priorität + P) * $\Delta t_A(\text{Elapsed-Time}) / (\Delta t_A(\text{CPU-Time}) * C)$

Tasks, die auf Zuteilung der Berechtigung zur Hauptspeicher-Nutzung warten und sich im Funktionszustand TPR befinden, erhalten eine durch die Konstante P ausgedrückte höhere Aktivierungspriorität.

P = 5 falls in TPR

P = 0 falls in TU

Der Konfigurationsfaktor C hängt vom Server-Typ ab und wird dynamisch mit steigender Paging-Rate erhöht.

Feste Aktivierungspriorität = Feste interne Priorität

Deaktivierung

Die Deaktivierung von aktiven Tasks dient dazu, ablaufbereiten inaktiven Tasks das Betriebsmittel Hauptspeicher zugänglich zu machen.

Kriterien für die Deaktivierung aktiver Tasks:

- Die weitere Verarbeitung ist nicht möglich, z.B. durch Warteaufrufe im Programm (PASS, VPASS) oder Warten auf Terminal-Eingabe beim Dialogbetrieb (WRTRD, RDATA).
- Wartezeiten beim Warten auf Ereignisse werden überschritten, z.B. durch Warten auf Terminal-Eingabe beim TP-Betrieb:

DCAM: YRECEIVE

UTM: KDCWAIT

Task-Kommunikation	ITC:	REVNT
	TU-Eventing:	SOLSIG
	Forward-Eventing:	RSOFEI

(siehe „[Kontrollfunktionen der Hauptspeicher-Verwaltung](#)“, Punkt „[Waiting Time Runout Control](#)“ auf Seite 255).

- Gewisse Systemdienste in Form von CPU-Zeit und Anzahl Ein-/Ausgaben wurden geleistet (siehe „[Kontrollfunktionen der Hauptspeicher-Verwaltung](#)“, Punkt „[System Service Slice Runout Control](#)“ auf Seite 254).

Die Deaktivierung unterbleibt, wenn die Funktion „No Deactivation“ (Makro TINF, Parameter DEACT) eingeschaltet ist.

Ausnahme: Der Warteaufruf VPASS_n führt immer zur Deaktivierung.

Zwangsdeaktivierung

Die Zwangsdeaktivierung von Tasks erfolgt, wenn von der Kontrollfunktion ACF (Activation Control Function) eine zu hohe Auslastung der Betriebsmittel CPU und Hauptspeicher bzw. eine zu hohe Paging-Rate in Abhängigkeit des CPU-Typs festgestellt wird (Matrix-Entscheidung).

Innerhalb eines ACF-Messintervalls wird jeweils nur eine Task deaktiviert.

Zur Bestimmung der Kategorie, aus der eine Task zwangsdeaktiviert werden soll, sind folgende Kategorie-Attribute maßgebend:

- MIN MPL
- MAX MPL
- WEIGHT



Durch Vorgabe entsprechend hoher Werte von MIN MPL und WEIGHT für Kategorien mit wichtigen Anwendungen muss dafür gesorgt werden, dass nur Tasks aus „weniger wichtigen“ Kategorien deaktiviert werden.

Für die Zwangsdeaktivierung kommen in erster Linie Tasks mit niedriger Priorität in Frage, die erst dabei sind, ihren Working-Set aufzubauen. Dadurch hält man vergeblich geleistete Aufwände des Systems auf einem Minimum.

Von einer Zwangsdeaktivierung sind ausgeschlossen:

- Tasks mit fester Priorität
- Tasks im Funktionszustand TPR
- Tasks, die die Funktion „No deactivation“ eingeschaltet haben

Verdrängung

Zu einer Verdrängung (Preemption) kann es kommen, wenn eine Aktivierungsanforderung vorliegt, aber die Kontrollfunktion ACF aufgrund der Auslastung der Betriebsmittel CPU und Hauptspeicher sowie der Höhe der Paging-Rate keine zusätzliche Aktivierung mehr zulässt.

Preemptions werden in der Reportgruppe PRIOR-ACF von openSM2 ausgewiesen. Sie sollten unbedingt vermieden werden. Ursache kann sein, dass:

- die MIN MPL-Werte zu hoch eingestellt sind
- zu viele Tasks mit fester Priorität laufen

Zwei Möglichkeiten der Verdrängung werden unterschieden:

a) Preemption zwischen **verschiedenen** Kategorien

Die aktive Task A wird durch eine inaktive Task B verdrängt, die einer anderen Kategorie angehört. Dies tritt ein, wenn:

- für die Kategorie, der die inaktive Task B angehört, gilt: $MPL < MIN\ MPL$
- und gleichzeitig für die Kategorie der aktiven Task A: $MPL > MIN\ MPL$

Die Preemption von Tasks verschiedener Kategorien läuft folgendermaßen ab:

1. Aktivierung einer Task aus einer Kategorie mit $MPL < MIN\ MPL$ und dem kleinsten Kategorie-Index. Die höchstpriorie Task aus dieser Kategorie wird aktiviert.
2. Zwangsdeaktivierung einer Task aus einer Kategorie mit $MPL > MIN\ MPL$ (siehe „[Zwangsdeaktivierung](#)“ auf Seite 250).

b) Preemption innerhalb der **gleichen** Kategorie

Die aktive Task A und die inaktive Task B gehören der gleichen Kategorie an. Dies ist nur möglich, wenn:

- die inaktive Task B eine feste Priorität hat
- und die aktive Task A eine variable Priorität hat

Die Preemption von Tasks innerhalb der gleichen Kategorie läuft folgendermaßen ab:

1. Zwangsdeaktivierung einer Task dieser Kategorie mit variabler Priorität (siehe „[Zwangsdeaktivierung](#)“ auf Seite 250).
2. Aktivierung der Task mit der festen Priorität.

Kontrollfunktionen der Hauptspeicher-Verwaltung

- Activation Control Function (ACF)

ACF misst periodisch die Auslastung der Betriebsmittel CPU und Hauptspeicher sowie die Paging-Aktivität (→ INVOCATION LONG = INVOCL, siehe Reportgruppe PRIOR-ACF von openSM2).

In Abhängigkeit von der CPU-Auslastung liegt das Aufrufintervall zwischen 0,25 und 2 Sekunden.

CPU-UTILIZATION

Die Größe wird aus der im Intervall gemittelten CPU-Auslastung und der Wartezeit auf Zuteilung der CPU errechnet.

MEMORY-UTILIZATION

Verhältnis des vom System geschätzten Working-Set-Bedarfs für alle aktiven Tasks (als Summe der PPC-Werte) zur Anzahl der für Paging verfügbaren Seiten (NPP = Number Pageable Pages).

PAGING-UTILIZATION

Höhe der Paging-Rate.

Die gemessenen Werte werden in folgende Klassen eingeteilt:

- H (High)
- M (Medium)
- L (Low)

Die UTILIZATIONs sind in der Reportgruppe PRIOR-ACF von openSM2 enthalten. Die Festlegung der Grenzwerte für die Klassen ist abhängig vom Server (CPU-Geschwindigkeit, Hauptspeicher-Ausbau) und wird beim Startup vorgenommen.

Je nach Betriebsmittelauslastung und vergebenen Kategorie-Attributen bzw. Task-Prioritäten erfolgt anhand einer Matrix die Entscheidung, ob eine weitere Aktivierung zulässig ist oder ob eine Zwangsdeaktivierung, eine Verdrängung oder gar keine Aktion erfolgen soll.

Vor der eigentlichen Aktivierung einer Task nimmt ACF eine zusätzliche Messung vor, die aber nur die Hauptspeicher-Auslastung betrifft, um sicherzustellen, dass die Auslastung des Hauptspeichers eine weitere Aktivierung zulässt (→ INVOCATION SHORT = INVOCS, siehe Reportgruppe PRIOR-ACF von openSM2).

- Preemption Control Function (PCF)

Task-Verdrängungen auf Aktivierungsebene sind aufwändig, da ggf. der Working-Set der verdrängten Task bei der Wiederaktivierung neu aufgebaut werden muss. Um den dadurch entstehenden Overhead gering zu halten, strebt das System eine möglichst kleine Verdrängungsrate an.

PCF überwacht periodisch die geglättete Verdrängungsrate pro Intervall. Das Aufrufintervall liegt – abhängig vom Maschinentyp – zwischen 20 und 48 Sekunden. Tritt während dieses Intervalls mindestens eine Verdrängung auf, so wird die MAX MPL OPTION eingeschaltet. Dies bedeutet, dass der Wert für MAX MPL ab diesem Zeitpunkt als harte Grenze anzusehen ist, über die hinaus nicht aktiviert wird. (Die Ausschaltung der MAX MPL OPTION erfolgt, wenn über mehrere Intervalle hinweg keine Verdrängung auftritt.)

PCF will die Verdrängungsrate niedrig halten. Dazu führt PCF Verdrängungsanforderungen möglichst nicht als Verdrängungen, sondern als zusätzliche Aktivierungen durch. Dies wird erreicht, indem PCF Hauptspeicher-Bereiche für Verdrängungsanforderungen reserviert, wenn eine bestimmte Verdrängungsrate pro Intervall (Preemption-Limit) überschritten wird.

Der für Verdrängungsanforderungen reservierte Hauptspeicher wird mit steigender Verdrängungsrate stufenweise vergrößert (5 bis 15% des realen Hauptspeicher-Ausbaus, abhängig von der Hauptspeicher-Größe). Damit wird den übrigen Tasks, die keine Verdrängungsanforderung stellen, eine höhere Hauptspeicher-Auslastung vorgegaukelt, sodass deren Aktivierung unterbleibt und genügend Hauptspeicher für Tasks **mit** Verdrängungsanforderung zur Aktivierung zur Verfügung steht.

Das System kennt 4 Stufen (Preemption-Level 0 bis 3), die mit der Meldung `EXC0455 TASK PREEMPTION LEVEL=(&00)` angezeigt werden:

Level 0 Normalzustand (Verdrängungsrate < Preemption-Limit)

Level 1, 2 Kurzfristige Überlastzustände

Der Preemption-Level wird immer erhöht, wenn

– Verdrängungsrate > Preemption-Limit

– Verdrängungsrate im Intervall $n + 1$ > Verdrängungsrate im Intervall n

Level 3 Langfristiger Überlastzustand (Dieser kann durch zu hohe Einstellung der MIN MPL-Parameter hervorgerufen werden: Tasks von Kategorien, die MIN MPL noch nicht erreicht haben, stellen grundsätzlich eine Verdrängungsanforderung.)

Der Operator wird informiert, dass Preemption-Level 3 erreicht ist und erhält ab diesem Zeitpunkt Meldung über jede Änderung des Preemption-Levels. Stellt das System mit Hilfe des geglätteten Multiprogramming-Grades fest, dass die MIN MPL-Parametereinstellung zu hoch ist, so wird der Operator aufgefordert, die MIN MPL-Parametereinstellung zu überprüfen und den MIN MPL-Wert geeigneter Kategorien zu senken.

Bei Absinken der Verdrängungsrate wird der reservierte Hauptspeicher stufenweise zurückgegeben. Zu dem Zeitpunkt, an dem der Normalzustand wieder erreicht ist, erhält der Operator die Meldung:

```
EXC0456  PREEMPTION CONTROL FUNCTION TERMINATED
```

Werden Preemption-Meldungen ausgegeben, sollte in jedem Fall Abhilfe geschaffen werden (siehe „Verdrängung“ auf Seite 251).

- Paging Load Control Function (PLC)

PLC führt langfristige Messungen (Aufrufintervall 20 bis 48 Sekunden in Abhängigkeit von der CPU-Geschwindigkeit) der Paging-Aktivitäten durch und kann als Ergänzung zu ACF gesehen werden.

Während ACF die aktuelle Paging-Rate misst und mit den vorgegebenen Grenzwerten vergleicht, um daraus kurzfristige Entscheidungen abzuleiten, beeinflusst PLC die beim Startup festgelegten Grenzwerte in Abhängigkeit von der Lastzusammenstellung.

Ist die Last dialogorientiert, werden höhere Paging-Aktivitäten zugelassen gegenüber einer Last, die TP- oder TP/Batch-orientiert ist. Damit wird erreicht, dass TP-orientierte Lasten durch Paging-Aktivitäten möglichst nicht beeinträchtigt werden bzw. dass bei batch-orientierter Last der Durchsatz optimiert wird.

PLC führt eine Statistik, wie oft hinsichtlich der Paging-Aktivitäten der Zustand

- Underload
- Mediumload
- Overload

pro Aufrufintervall auftritt. Tritt einmal pro Aufrufintervall der Zustand Overload auf, so wird die MAX MPL OPTION eingeschaltet. Die MAX MPL OPTION wird ausgeschaltet, wenn über mehrere Intervalle hinweg der Zustand Paging-Overload nicht auftritt, der Preemption-Level 0 herrscht und die Preemption-Rate < 1 ist.

- System Service Slice Runout Control

Die „System Service Slice“ dient zur Aufsummierung der Systemdienste, die eine aktive Task in Form von CPU-Zeit (im Funktionszustand TU + TPR) und von durchgeführten Ein-/Ausgaben erhalten hat.

Der Systemservice wird nach der Formel $SERVICE = (A * B * C) + D$ berechnet, mit:

- A: Mittlere Befehlsausführungszeit (abhängig von der CPU-Geschwindigkeit)
- B: Mittlere Befehlsanzahl pro Ein-/Ausgabe-Anforderung
- C: Anzahl der Ein-/Ausgaben seit der letzten Aktivierung
- D: CPU-Zeit seit der letzten Aktivierung

Die Größe der „System Service Slice“ ist kategorie- und Server-abhängig.

Das Aufrufintervall für System-Service-Slice-Runout-Control beträgt 2 Sekunden.

Für die Ergreifung von Maßnahmen wird bei System-Service-Slice-Runout unterschieden, ob sich die Task im Funktionszustand TU oder TPR befindet:

– System Service Slice Runout – Funktionszustand TU

Die Deaktivierung erfolgt nur dann, wenn in der **gleichen** Kategorie eine ablaufbereite Task (Status: inactive, ready) mit höherer Priorität vorhanden ist.

– System Service Slice Runout – Funktionszustand TPR

Wenn in der gleichen Kategorie eine ablaufbereite Task (Status: inactive, ready) mit höherer Priorität vorhanden ist, erfolgt eine Deaktivierung erst beim 2. Auftreten von System-Service-Slice-Runout. Tritt die Task nach dem ersten Runout in den Zustand TU, dann wird sie sofort so wie oben beschrieben behandelt.

● Waiting Time Runout Control

Die Funktion „Waiting Time Limit“ wird zur Überwachung der Tasks verwendet, die auf ein Ereignis warten (Börsen- oder ITC-Ereignisse) und dabei Hauptspeicher belegen.

Der Aufruf von „Waiting Time Runout Control“ erfolgt nur dann, wenn mehr als 25% des seitenwechselbaren Hauptspeichers durch Tasks belegt werden, die auf Börsen-Ereignisse (z.B. DCAM: YRECEIVE, UTM: KDCWAIT, TU-Eventing: SOLSIG, Forward-Eventing: RSOFEI) oder ITC-Ereignisse (Inter-Task-Communication: REVNT) warten. Das Aufrufintervall bewegt sich in Abhängigkeit von der Hauptspeicher-Auslastung zwischen 1 und 7 Sekunden.

Wird das Waiting-Time-Limit für eine oder auch mehrere Tasks überschritten, so erfolgt die Deaktivierung in folgender Weise:

Tasks mit variabler Priorität, die einer Kategorie angehören, deren $MPL > MIN\ MPL$ ist, werden zuerst deaktiviert (sofern die Deaktivierung bei Waiting-Time-Runout erlaubt ist: Parameter DWTR im TINF-Makro).

Das Verfahren wird abgebrochen, sobald der von den wartenden Tasks gebundene seitenwechselbare Hauptspeicher die 25%-Grenze unterschreitet.

Reicht diese Maßnahme nicht aus, um die 25%-Grenze zu unterschreiten, so wird der Deaktivierungsvorgang auf wartende Tasks mit fester Priorität ausgedehnt sowie auf Kategorien, deren $MPL\text{-Wert} < MIN\ MPL$ ist (sofern die Deaktivierung erlaubt ist).

Unabhängig von der 25%-Grenze werden alle Tasks deaktiviert, die länger als 30 Sekunden warten und die genannten Bedingungen erfüllen.

Seitenverwaltungs-Algorithmus (System Working Set-Verfahren, SYS-WS)

Das Verfahren verwaltet die Hauptspeicher-Seiten global, wobei die Einteilung entsprechend ihrem „Zugriffsalter“ in vier Gruppen erfolgt:

Gruppe 1	Just paged in pages	} Aktive Gruppen: Seiten innerhalb dieser Gruppen sind ohne Hardware- Unterbrechung ansprechbar
Gruppe 2	Recently referenced pages	
Gruppe 3	Immediately replaceable pages and Not recently referenced pages	
Gruppe 4	Read Only Queue Read Write Queue Empty Queue	} Inaktive Gruppe: Free-Pool

Die Seiten jeder Gruppe sind miteinander verkettet.

Prinzipieller Ablauf

Beim Auftreten des Ereignisses „Seite nicht im Hauptspeicher“ (Page Fault) wird ein Seitenrahmen aus der Gruppe 4 mittels Paging-Ein-/Ausgabe (Page Read) mit der angeforderten Seite gefüllt und nach Gruppe 1 gebracht (Gruppen-Übergang 4 → 1; Reference-Bit=ON).

Grundsätzlich wird versucht, möglichst viele Seiten aktiv zu halten. Der Free-Pool ist daher relativ klein. Eine gewisse Minimalgröße zur Befriedigung schnell aufeinander folgender Seitenanforderungen ist jedoch notwendig.

Bei Erreichen der Minimalgröße erfolgt der Anstoß einer Routine zur Auffüllung des Free-Pools mit Seiten aus Gruppe 3 (Gruppen-Übergang 3 → 4).

Seiten mit Reference-Bit=ON werden an das Ende der Kette von Gruppe 2 gebracht und das Reference-Bit gelöscht.

Seiten, die zwei mal hintereinander mit Reference-Bit=OFF angetroffen werden, kommen in die Gruppe 4.

Die Durchsuchung der Gruppe 3 erfolgt solange, bis die benötigte Anzahl Seiten zur Auffüllung des Free-Pools gefunden ist.

Enthält Gruppe 3 nicht genügend freigebbare Seiten, so erfolgt die Auffüllung von Gruppe 3, indem sämtliche Seiten von Gruppe 2 nach Gruppe 3 gebracht werden (Gruppen-Übergang 2 → 3).

Ebenso wird die nun leere Gruppe 2 wieder aufgefüllt durch Umkettung sämtlicher Seiten von Gruppe 1 nach Gruppe 2 (Gruppen-Übergang 1 → 2).

Da die Seitenanforderungen sowohl für die automatische Auffüllung der Gruppe 1 sorgen, als auch den Seitenersetzungs-Algorithmus anstoßen, ergibt sich ein konsistenter Zustand.

Der einfache Algorithmus des SYS-WS-Verfahrens führt zu einer Aufwandsreduzierung im Paging-Management, aber auch zu einer „gröberen“ Bestimmung des Working-Set-Bedarfs der einzelnen Tasks. Die Aufwandsreduzierung macht sich besonders bei großem Hauptspeicher bemerkbar.

Wird eine Task deaktiviert, so behält sie auch im inaktiven Zustand ihren Working-Set. Wird die Task erneut aktiviert, sind keine Page-Reclaims zur Wiederbeschaffung ihres Working-Sets notwendig.

Das SYS-WS-Verfahren kennt zwar den UPG-Wert (Used-Pages) jeder Task, hat jedoch keine Möglichkeit festzustellen, in welcher Phase sich die Task befindet (auch inaktive Tasks haben einen UPG-Wert $\neq 0$).

Die Bestimmung des PPC-Wertes (Planned Page Count) erfolgt daher näherungsweise durch einen Vergleich des UPG-Wertes mit einem angenommenen, vom Kategorietyp abhängigen, mittleren Working-Set-Wert.

Da möglichst viele Seiten aktiv gehalten werden, liegt der Summen-UPG-Wert praktisch immer in der Größenordnung des verfügbaren seitenwechselbaren Hauptspeichers und hat nur begrenzte Aussagekraft. Der Working-Set-Bedarf wird ausschließlich durch den PPC-Wert ausgedrückt. Der Wert ist eine grobe Schätzung. Bei den heute üblichen großen Speicherausbauten ergeben sich Vorteile dadurch, dass die Aufwände für die ohnehin nicht benötigte genaue Abschätzung des Speicherbedarfs unterbleiben.

9.3.2 CPU-Verwaltung

Folgende Begriffe werden verwendet:

- **Initiierung**
Zuteilung des Betriebsmittels CPU an eine aktive, ablaufbereite Task (die sich im Zustand „active, ready“ befindet).
- **Deinitiierung**
Entzug des Betriebsmittels CPU bei Erwartung von Wartezeiten (Übergang in den Zustand: „active, not ready“).
- **Verdrängung**
Deinitiierung einer Task, gefolgt von der Initiierung einer höherprioren Task.
- **Kontrollfunktion der CPU-Verwaltung**
Micro Time Slice (MTS, Mikro-Zeitscheibe)

Initiierung

Die Dispatcher-Routine des Task-Initiators wählt aus der Anzahl der aktiven, ablaufbereiten Tasks die Task mit der höchsten Initiierungspriorität aus und überträgt ihr die Kontrolle (→ Task in Control).

Bei Multiprozessoren gibt es CPU-lokale Warteschlangen (siehe auch [Abschnitt „Steuern der Tasks über Warteschlangen“ auf Seite 261](#)), die bei der Aktivierung von Tasks zyklisch bestückt werden. Pro CPU wird die in der zugeordneten Warteschlange stehende Task mit der höchsten Initiierungspriorität ausgewählt. Eventuell höhere Prioritäten von Tasks in Warteschlangen, die anderen CPUs zugeordnet sind, spielen keine Rolle. Zusätzlich wird die Strategie „Fremd vor IDLE“ verfolgt, d.h. bevor die CPU mangels Last in den Wartezustand übergeht, wird eine Task aus einer „fremden“ Warteschlange initiiert. In diesem Fall erfolgt der Zugriff auf diejenige Warteschlange, die die meisten Tasks enthält („längste Q1“).

Bei Einsatz der Funktion TANGRAM erfolgt die Bestückung der CPU-lokalen Warteschlangen nicht zyklisch, sondern gemäß den TANGRAM-Zuordnungs-Algorithmen (siehe [Abschnitt „Einführung in das TANGRAM-Konzept“ auf Seite 233](#)). In diesem Zusammenhang wird die Strategie „IDLE vor Fremd“ verfolgt.

Initiierungspriorität

Die Ermittlung der Initiierungspriorität erfolgt analog der Berechnung der Aktivierungspriorität. Zusätzlich wird berücksichtigt:

- der Zeitpunkt t_A des letzten „System Service Slice Runouts“ (es wird die Differenz: Aktueller Zeitpunkt - t_A gebildet)
- die „Task in Control“ wird bei der Prioritätsberechnung bevorzugt (um die Verdrängungsrate möglichst gering zu halten, gilt: $P = P + 5$)

Deinitiierung

Im Sinne einer effizienten Nutzung der CPU entzieht das System einer Task die Kontrolle (auch wenn diese Task die höchste Priorität besitzt), wenn Wartezeiten unmittelbar absehbar sind:

- Bei kurzen Wartezeiten erfolgt der Übergang in den Zustand „active, not ready“, z. B. bei Durchführung der Ein-/Ausgabe (DVS-, Paging-Ein-/Ausgabe), Warten auf Börsen- oder ITC-Ereignisse, Warteaufruf VPASS 0 im Programm.
- Bei langen Wartezeiten kommt es zusätzlich zu einer Deaktivierung, z.B. bei Terminal-Eingabe im Dialogbetrieb, Warteaufruf PASS, VPASS im Programm.

Verdrängung

Auf Initiierungsebene erfolgt eine Verdrängung ausschließlich in Abhängigkeit von der Initiierungspriorität.

Damit die „Task in Control“ verdrängt werden kann, muss die Priorität der neu zu initiierenden Task:

- mindestens um den Wert 5 höher sein
- mindestens um den Wert 10 höher sein, wenn sich die „Task in Control“ im Funktionszustand TPR befindet

Nach dem Aufruf eines SVC, der nicht zur Deinitiierung der Task geführt hat, findet keine Verdrängung statt. Nur wenn während der Bearbeitung dieses SVC eine andere Task auf die erste Position der CPU-Warteschlange gelangte und die oben genannten Bedingungen erfüllt sind, verliert die SVC-rufende Task die CPU.

Kontrollfunktion der CPU-Verwaltung

- Micro Time Slice (MTS, Mikro-Zeitscheibe)

Die „Micro Time Slice“ ist diejenige Zeit, die eine Task maximal die CPU ohne Unterbrechung belegen darf.

Die verbrauchte CPU-Zeit im Funktionszustand TU und TPR wird aufsummiert, um die CPU-Intensität der einzelnen Task festzustellen. Bei „Micro Time Slice Runout“ erfolgt die Überprüfung bzw. Neufestsetzung der Priorität. Bei Multiprozessoren wird nach einem „Micro Time Slice Runout“ aus allen Warteschlangen (Q1) diejenige Task ausgewählt, die die höchste Priorität hat.

Die Größe der MTS ist innerhalb zweier Grenzwerte variabel und hängt von der CPU-Intensität ab. Nach jeder mit einem Wartezustand verbundenen Ein-/Ausgabe wird die MTS neu festgesetzt:

$MTS = \text{Aktuelle CPU-Zeit} - \text{CPU-Zeit der vorletzten Ein-/Ausgabe}$

Die Grenzwerte sind abhängig von der CPU-Geschwindigkeit und sind umso kleiner, je schneller die CPU ist.

Je Ein-/Ausgabe-intensiver die Charakteristik einer Task ist, desto kleiner ist die zugehörige MTS, wobei aber ein Minimum-Wert nicht unterschritten werden kann. Aufgrund der geringen CPU-Intensität einer Ein-/Ausgabe-intensiven Task tritt das Ereignis „Micro Time Slice Runout“ nicht auf.

Ändert sich die Task-Charakteristik in Richtung größerer CPU-Intensität, so ergibt sich relativ rasch ein „Micro Time Slice Runout“ mit anschließender Prioritätsberechnung. Damit ist ein schnelles Reagieren des Systems (es braucht nicht die periodische Prioritätsüberprüfung abgewartet werden) auf Änderungen der Lastcharakteristik von Ein-/Ausgabe-intensiv in Richtung CPU-intensiv gewährleistet. Dieser Fall der Laständerung ist besonders kritisch, da das Dominieren CPU-intensiver Tasks eine entsprechende Verschlechterung des Simultanarbeitsgrades und damit des Durchsatzes zur Folge hat.

9.4 Steuern der Tasks über Warteschlangen

Im Betriebssystem vorhandene Tasks können sich grundsätzlich in fünf verschiedenen Zuständen befinden.

- TASK IN CONTROL
- ACTIVE READY
- ACTIVE NOT READY
- INACTIVE READY
- INACTIVE NOT READY

Es können nur so viele Tasks den Zustand TASK IN CONTROL einnehmen, wie der Server CPUs hat.

Die übrigen vier Zustände sind Wartezustände, die sich in zwei Gruppen einteilen lassen, je nachdem, ob die Berechtigung zur Hauptspeicher-Nutzung bereits erteilt wurde (ACTIVE) oder nicht (INACTIVE).

Die Zuordnung der Tasks zu den einzelnen Wartezuständen erfolgt mit Hilfe von Warteschlangen (Queues), wobei aus Übersichtlichkeitsgründen bei den Zuständen NOT READY eine weitere Unterteilung nach Ereignissen stattfindet, warum diese Task nicht ablaufbereit ist.

Zur Verbesserung des MP-Faktors bei Multiprozessoren sind die Warteschlangen des Aktivraumes für jede CPU getrennt vorhanden (CPU-lokal), während die restlichen Warteschlangen für alle CPUs gemeinsam sind (siehe [Bild 15 auf Seite 262](#)).

Die Zustandswechsel (charakterisiert durch Warteschlangenübergänge) werden durch die PEND/UNPEND-Routinen realisiert:

- UNPEND-Routinen bringen die Task aus der jeweiligen Warteposition näher in Richtung Benutzung der CPU.
- PEND-Routinen bringen die Task aufgrund eines eintretenden Ereignisses in einen Wartezustand.

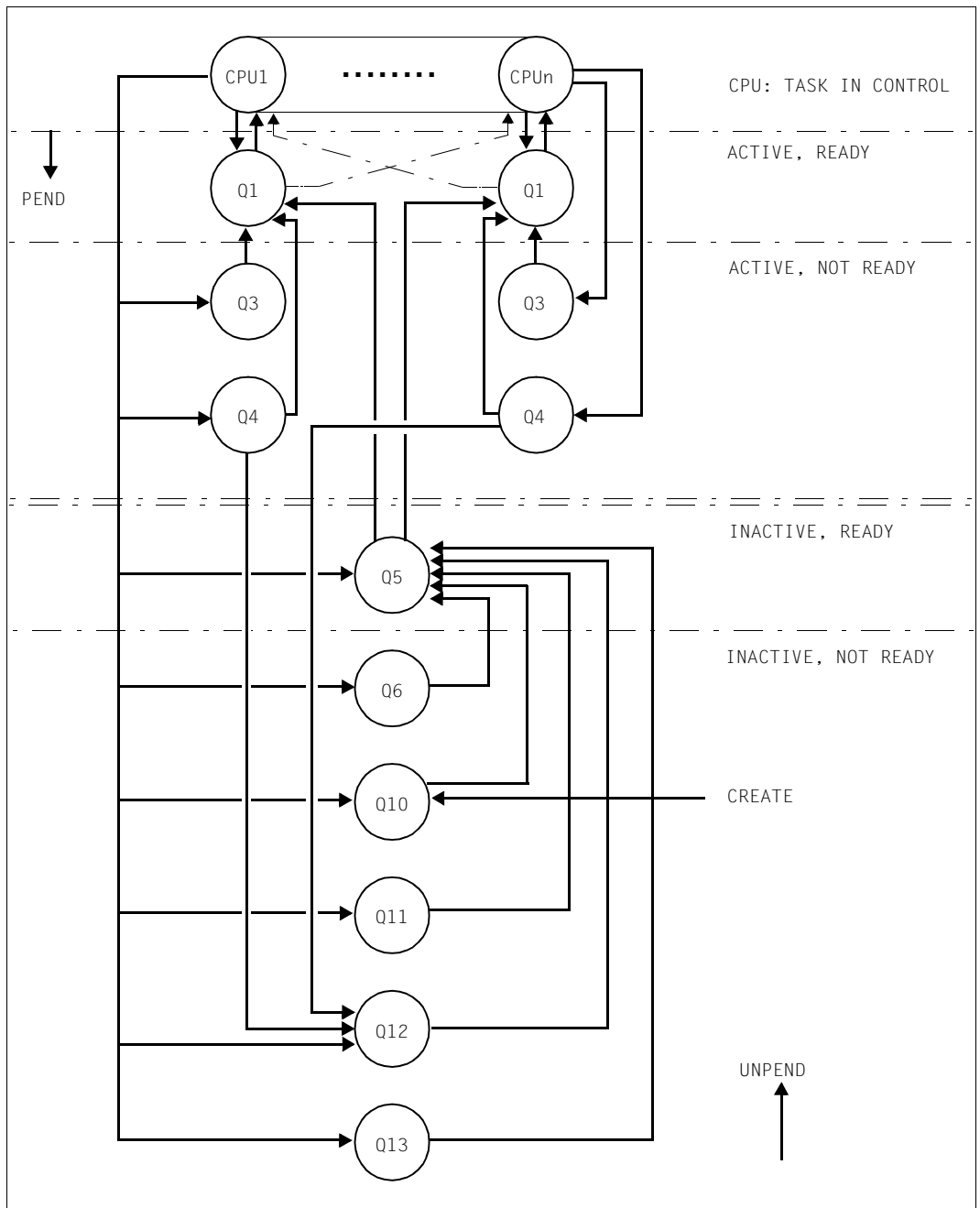


Bild 15: Warteschlangenübergänge

Übersicht über die einzelnen Warteschlangen

Die Warteschlangen Q0, Q1, Q2, Q3 und Q4 sind einmal pro CPU vorhanden.

- **TASK IN CONTROL**
 - Q0 Task belegt CPU

- **ACTIVE, READY**
 - Q1 CPU-Queue
 Sie enthält aktive ablaufbereite Tasks, die auf Benutzung der CPU warten.

- **ACTIVE, NOT READY**
 - Q2 Sonderfall (nicht eingezeichnet): Mess-System-Tasks
 - Q3 Warten auf Beendigung der Paging-Ein-/Ausgabe
 - Q4 Warten auf Beendigung der DVS-Ein-/Ausgabe (Platte, Band)
 Warten auf Börsen-Ereignisse (Short Wait)
 Warten auf Beendigung VPASS 0, VPASS MSEC=Y

- **INACTIVE, READY**
 - Q5 Memory-Queue
 Sie enthält inaktive ablaufbereite Tasks, die auf Berechtigung zur Hauptspeicher-Nutzung warten.
 Besteht aus 4-16 Sub-Queues entsprechend den vordefinierten Kategorien:
 SYS
 TP
 DIALOG
 BATCH
 und bis zu 12 weiteren von der Systemverwaltung definierbaren Kategorien.
 - Q6 PCS-Not-Admitted-Queue
 Diese Warteschlange enthält inaktive ablaufbereite Tasks, die vom PCS (Performance Control Subsystem) nicht zur Aktivierung zugelassen werden.

- INACTIVE, NOT READY
 - Q10 HOLD-Queue
 - Enthält neu erzeugte Tasks (Aufruf: \$CREA)
 - Warten auf periphere Geräte (/SECURE-RESOURCE-ALLOCATION)
 - Anhalten durch Systembetreuung (/HOLD-TASK)
 - Nichtbeendbare Tasks (schwere Fehler-Situationen)
 - Q11 Inaktive System-Tasks
 - Q12 Warten auf Terminal-Eingabe (WRTRD, RDATA)
 - Warten auf Börsen-Ereignisse (Long Wait)
 - Warten auf Beantwortung einer Bedienplatzmeldung
 - Q13 Warten auf Beendigung der Aufrufe PASS, VPASS
 - Warten auf ITC-Ereignis

9.5 Ablauf einer Ein-/Ausgabe

Die Abwicklung von Ein-/Ausgaben erfolgt in BS2000/OSD über das Datenverwaltungssystem (DVS). Im Vergleich zu den Befehlsausführungen in der CPU (Mikrosekunden oder Bruchteile davon) dauert die physikalische Durchführung der Ein-/Ausgabe um eine Größenordnung länger (Millisekunden) und hat daher einen maßgeblichen Einfluss auf das Zeitverhalten.

Bei interaktiven Anwendungen stehen die Ein-/Ausgaben auf Platten (genauer: logische Volumes) im Vordergrund. Im folgenden Abschnitt werden die wesentlichen Zeitanteile bei der Abwicklung einer Ein-/Ausgabe näher erläutert.

9.5.1 Ein-/Ausgaben auf Platten

Der Anwender formuliert seine Ein-/Ausgabe-Anforderungen im Anwenderprogramm mit Makroaufrufen (GET, PUT, GETKY, STORE, PAM, ...) auf **logischer** Ebene.

Bei Verwendung höherer Programmiersprachen werden die READ- bzw. WRITE-Anweisungen durch das jeweilige Laufzeitsystem entsprechend umgesetzt.

Die Makro-Auflösung enthält einen äquivalenten SVC-Aufruf, der aber nicht bei jedem Makroaufruf durchlaufen wird.

Bei sequenziellen Eingabe-Operationen (GET) stellt die Zugriffsmethode (SAM, ISAM) dem Anwender den nächsten logischen Satz aus dem Eingabe-Puffer zur Verfügung, in den der Datenblock physikalisch eingelesen wurde. Nur wenn sich kein Satz mehr im Eingabe-Puffer befindet, wird der SVC-Aufruf durchlaufen.

Bei sequenziellen Ausgabe-Operationen (PUT) erfolgt der SVC-Aufruf durch die Zugriffsmethode (SAM, ISAM) nur, wenn im Ausgabe-Puffer kein Platz mehr ist.

Bevor eine „Update“-Operation PUTX (ISAM) durchgeführt werden kann, muss der entsprechende Satz gelesen worden sein (GET, GETKY). Nur wenn sich beim Update-Vorgang die Satzlänge vergrößert, wird der SVC-Aufruf durchlaufen.

Stellt die aufgerufene DVS-Zugriffsmethode fest, dass eine physikalische Ein-/Ausgabe durchzuführen ist, so wird mit dem SVC-Aufruf \$EXCP (Execute Channel Program) oder \$XCPW (Execute Channel Program with Wait) die Ein-/Ausgabe angestoßen.

Diese Aufrufe stehen dem Anwender nicht unmittelbar zur Verfügung.

Die zeitlichen Verhältnisse auf der Kanal- bzw. Platten-Seite sind für beide Aufrufe gleich; nur die Reaktion der aufrufenden Programmroutine ist unterschiedlich.

Beim Aufruf \$XCPW wird der Programmablauf unterbrochen (Versetzung in den Wartezustand durch die systeminterne PEND-Routine von Q0 → Q4) und erst nach Beendigung der Ein-/Ausgabe fortgesetzt.



Die Warteschlange Q4 ist nur das „Sammelbecken“ für Tasks, die auf Durchführung bzw. Beendigung der Ein-/Ausgabe (und Börsen-Ereignisse) warten. Aus der Länge der Q4 kann i.A. noch kein Rückschluss auf einen evtl. Engpass bei der Ein-/Ausgabe getroffen werden. Wesentlich sind in diesem Zusammenhang die Warteschlangen vor den einzelnen Ein-/Ausgabe-Geräten (Device-Queues; siehe auch Reportgruppe DEVICE von openSM2).

Beim Aufruf \$EXCP läuft das Programm asynchron zur Abwicklung der Ein-/Ausgabe weiter (führt andere, von der Ein-/Ausgabe unabhängige Funktionen durch) und gibt zu gegebener Zeit einen expliziten Warte-Aufruf. Der Einfachheit halber wird im Folgenden nur der Aufruf \$XCPW weiter dargestellt.

Ist das logische Volume zum Aufrufzeitpunkt des \$XCPW belegt (mit der Durchführung von Ein-/Ausgaben für andere Anwender beschäftigt), so tritt eine Wartezeit vor dem Volume auf. Eine höhere Volume-Auslastung bewirkt eine größere Wartezeit (siehe [Abschnitt „Software-Bedienzeit“ auf Seite 268](#)).

Bei Einsatz der Funktion PAV stehen pro logischem Volume (Basis-Gerät) mehrere Geräteadressen (Alias-Geräte) zur Verfügung, über die gleichzeitig Ein-/Ausgabe-Aufträge abgewickelt werden können. Die Entstehung größerer Wartezeiten wird dadurch vermieden.

Sobald das Volume frei ist (oder zumindest ein freies Alias-Gerät verfügbar ist), wird vom Betriebssystem der Befehl START SUBCHANNEL (SSCH) zusammen mit dem „Operating Request Block“ (ORB), der in der „Logical Path Mask“ eine Beschreibung der möglichen Zugriffspfade enthält, an das DCS übergeben.

Sind sämtliche Zugriffspfade belegt (was bei mehreren Zugriffspfaden sehr selten vorkommt), tritt eine entsprechende Wartezeit auf (Function pending time: siehe Reportgruppe SERVICETIME von openSM2), bevor die Ein-/Ausgabe mit dem Befehl START IO (SIO) eingeleitet werden kann. Die Wartezeit auf einen freien Zugriffspfad ist i.d.R. so klein, dass sie vernachlässigt werden kann.



Auf SQ-Servern wird die eigentliche Abwicklung der Ein-/Ausgabe (ausgelöst durch den zu SSCH äquivalenten Befehl) durch die I/O-Treiber von X2000 auf einer separaten CPU durchgeführt.

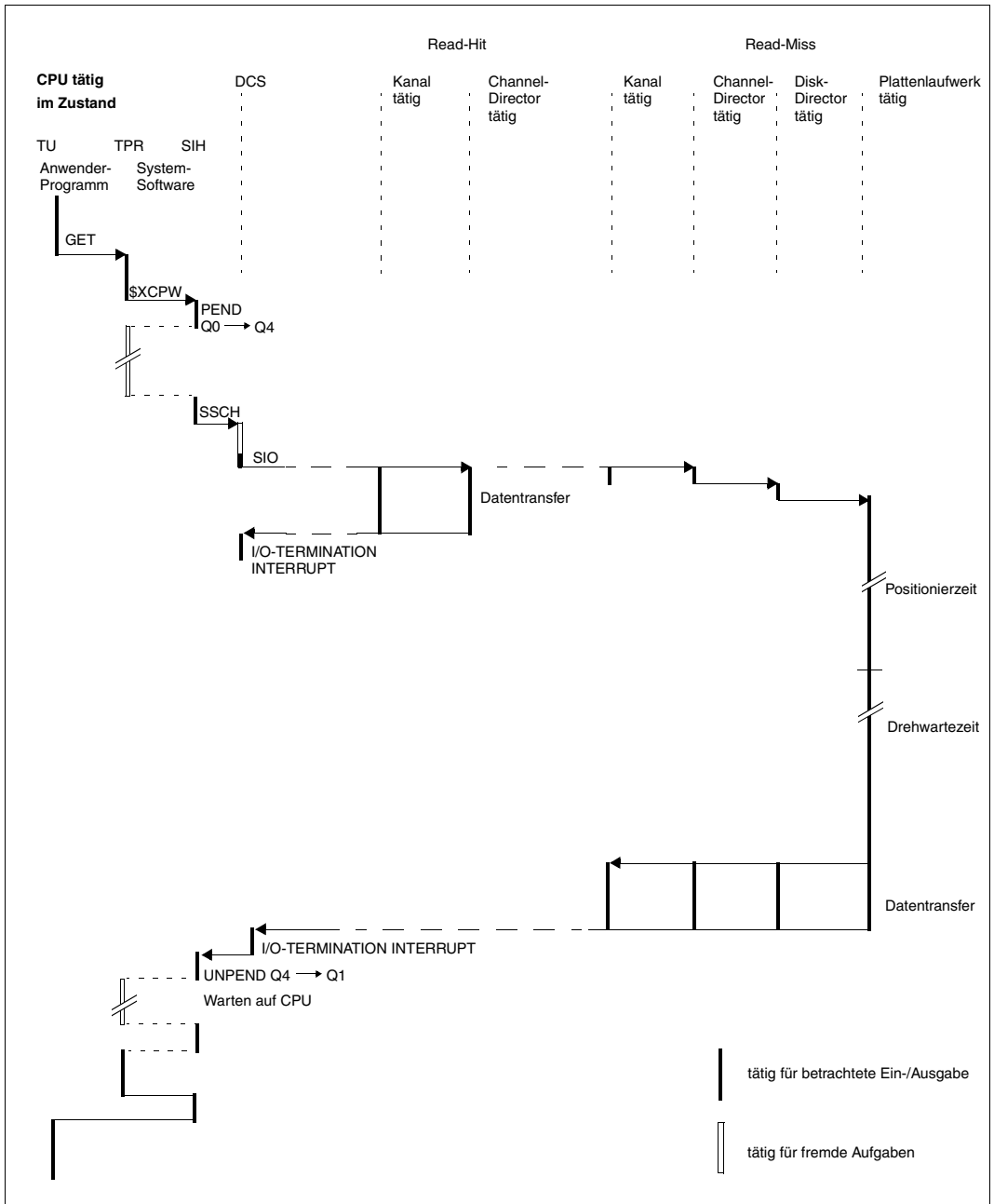


Bild 16: Zeitlicher Ablauf einer Platten-Ein-/Ausgabe Read-Hit/Read-Miss

9.5.2 Hardware-Bedienzeit

Als Hardware-Bedienzeit wird die Zeit vom Befehl START SUBCHANNEL (SSCH) bis zum IO-Termination-Interrupt bezeichnet.

Bei Plattenspeichersystemen ergibt sich eine stark unterschiedliche Hardware-Bedienzeit, je nachdem ob es sich um einen Cache-Hit oder Cache-Miss handelt.

Ein typischer Wert für einen Cache-Hit liegt bei Ein-/Ausgaben mit einer Blockgröße von 4 KB bei ca. 1 ms. Der Wert für einen Cache-Miss liegt in diesem Fall bei ca. 10 ms (siehe [Abschnitt „Eigenschaften externer Plattenspeichersysteme“ auf Seite 45](#)).

9.5.3 Software-Bedienzeit

Die Software-Bedienzeit je Ein-/Ausgabe, die der Anwender sieht, besteht im Wesentlichen aus:

- der Wartezeit vor dem jeweiligen Volume und
- der Hardware-Bedienzeit bei der Durchführung der physikalischen Ein-/Ausgabe (siehe Reportgruppe DISK von openSM2: SOFTWARE-DURATION und HARDWARE-DURATION).

Die Wartezeit W vor dem Volume ergibt sich nach den Gesetzen über die Warteschlangenbildung in Abhängigkeit von:

- der Verteilung der Zeiten zwischen den Ein-/Ausgabe-Anforderungen (Zwischenankunftszeiten),
- der Verteilung der Hardware-Bedienzeiten und
- der Auslastung des Volumes.

Ohne tieferes Eindringen in die Warteschlangen-Theorie können für die Abschätzung der durchschnittlichen Wartezeit W die zwei folgenden Formeln herangezogen werden:

1. Bei Annahme der Voraussetzung „M/M/1“ (ungünstigster Fall):

M exponentielle Verteilung der Zwischenankunftszeiten

M exponentielle Verteilung der Hardware-Bedienzeiten

1 eine Bedienstation

$$W = S * U / (1-U)$$

S durchschnittliche Hardware-Bedienzeit (Service time)

U Auslastung des Volumes (Utilization)

Beispiel

$$S = 6 \text{ ms}; U = 30\%$$

$$W = 6 \text{ ms} * 0,3 / (1 - 0,3) = 2,6 \text{ ms}$$

Bei 30% Auslastung des Volumens müssen die Ein-/Ausgabe-Anforderungen durchschnittlich 2,6 ms warten, bis das Volume frei wird.

2. Bei Annahme der Voraussetzung „M/D/1“ (günstigster Fall):

M exponentielle Verteilung der Zwischenankunftszeiten

D konstante Hardware-Bedienzeit

1 1 Bedienstation

$$W = S * U / (2 * (1-U))$$

Ein Vergleich der beiden Formeln zeigt, dass die durchschnittliche Wartezeit W bei konstanter Hardware-Bedienzeit (M/D/1) nur halb so groß ist wie bei einer Exponentialverteilung der Hardware-Bedienzeit (M/M/1).

In der Praxis kommt weder eine streng exponentielle Verteilung der Hardware-Bedienzeiten, noch eine konstante Bedienzeit vor. Die aufgrund der Auslastung von Volumens auftretenden Wartezeiten liegen etwa in der Mitte der Ergebnisse der beiden Formeln.

Die folgende Tabelle verdeutlicht, um welchen Faktor die Hardware-Bedienzeit abhängig von der Volume-Auslastung gedehnt wird.

Volume-Auslastung (%)	Dehnfaktor	
	M/M/1	M/D/1
10	1,11	1,06
20	1,25	1,13
30	1,43	1,21
40	1,67	1,33
50	2,00	1,50
60	2,50	1,75
70	3,33	2,17
80	5,00	3,00
90	10,00	5,50

Die Wartezeit vor dem Volume bildet einen wesentlichen Bestandteil der Ein-/Ausgabezeit. Volumens, die von mehreren Anwendern gemeinschaftlich benutzt werden, sollten nicht stärker als 30% ausgelastet werden.

Bei Einsatz der Funktion PAV stehen pro logischem Volume (Basis-Gerät) mehrere Geräteadressen (Alias-Geräte) zur Verfügung, wodurch die nach außen sichtbare Volume-Auslastung deutlich absinkt. (SM2 bildet den Mittelwert über das Basis-Gerät und die zugehörigen Alias-Geräte.)

10 Einsatz des Mess-Systems openSM2

openSM2 (BS2000/OSD) ist ein Softwareprodukt zur Überwachung und Leistungsvermessung von BS2000/OSD-Servern. Es ist zur Beobachtung des Lastverhaltens und der Betriebsmittelauslastung, zur Abschätzung von Trends und zur Problemanalyse sehr gut geeignet. Einführende Informationen zu openSM2 finden Sie im Handbuch „openSM2“ [18].

openSM2 (BS2000/OSD) bildet zusammen mit openSM2 (Open Systems) die Produktlinie openSM2, eine leistungsfähige Lösung zur Überwachung heterogener IT-Umgebungen.

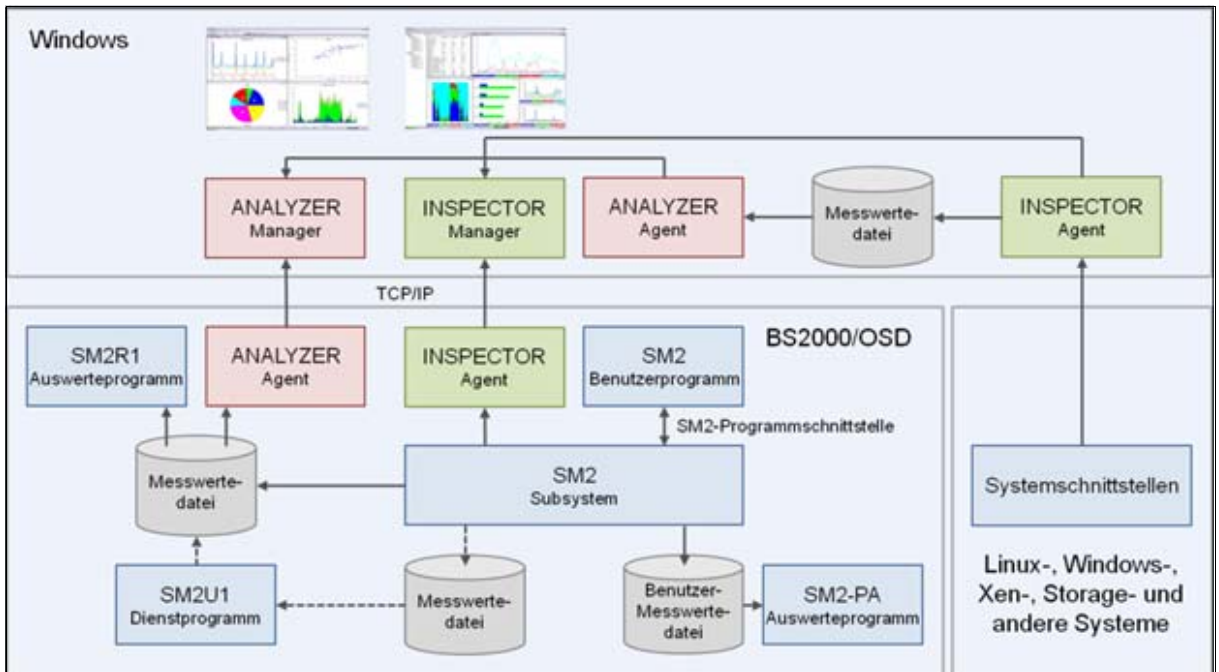


Bild 17: Komponenten der Produktlinie openSM2

10.1 Messmonitor SM2

Die Berechnung der Messwerte durch den Messmonitor SM2 erfolgt in gleichmäßigen zeitlichen Abständen, den Messintervallen (10 Sekunden bis 1 Stunde). Am Ende jedes Messintervalls werden die Messwerte in einen Puffer und in die Messwertedatei geschrieben.

Die Erfassung der Messwerte erfolgt

- einmal am Ende des Messintervalls (für aufsteigende Zähler durch Differenzbildung zu den Werten des vorhergehenden Messintervalls)
- während des ganzen Messintervalls (für variierende Messgrößen)
 - stichprobengesteuert (Sampling)
durch Erfassung einer Stichprobe mehrmals im Messintervall (Stichprobenintervall) und Bildung des Mittelwerts über alle Stichprobenwerte am Ende des Messintervalls
 - ereignisgesteuert (Eventing)
durch Aktivierung einer Messroutine beim Eintreten bestimmter Ereignisse, z.B. Start einer Ein-/Ausgabe

Art	Genauigkeit	Overhead	Messzeitraum	Datenvolumen
Sampling	abhängig von der Häufigkeit der Stichproben	abhängig von der Häufigkeit der Stichproben	beliebig lang	überschaubar
Eventing	hoch	abhängig von der Anzahl der Ereignisse	kurz (15-30 Min.)	u.U. umfangreich

Für folgende Anlässe wird der Einsatz von SM2 empfohlen:

- ständig (System-begleitend)
- bei auftretenden Lastproblemen
- bevor eine neue Anwendung auf den Server zu übernehmen ist
- vor und nach Umstellung auf neue Versionen (z.B. Betriebssystem, Anwendersoftware, Datenbanken)
- bei Einsatz neuer Funktionen (z.B. NK-ISAM, PCS, DAB)
- bei Änderung der Hardware

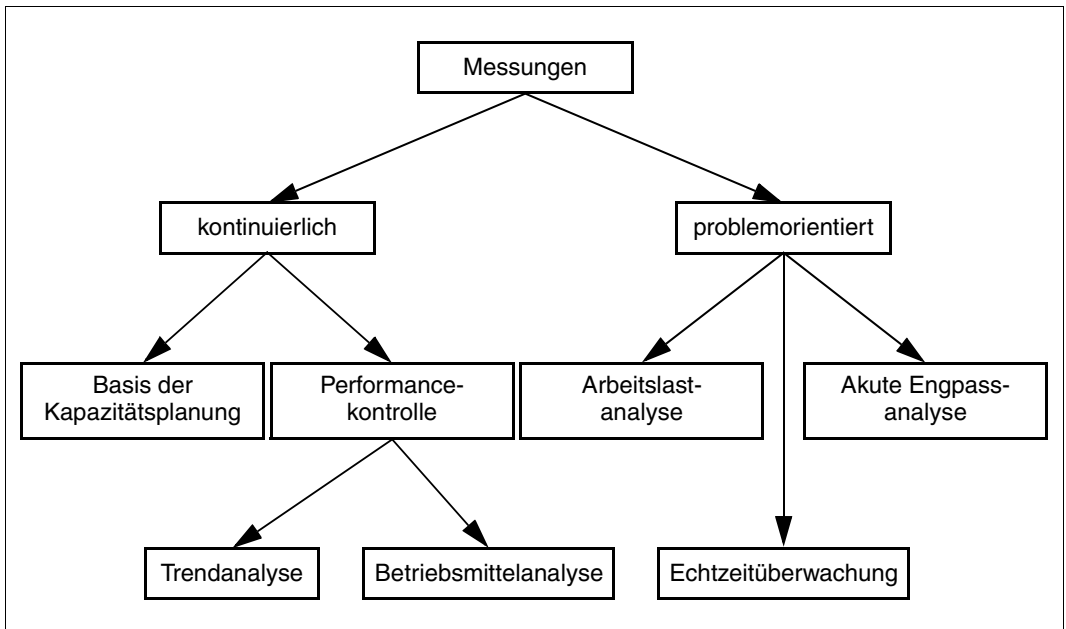


Bild 18: Einsatz des Messmonitors SM2

10.2 Routinemäßige Systemüberwachung

10.2.1 Parameter für die Messung und Auswertung

SM2 sollte in jeder Session mitlaufen und seine Ergebnisse in eine Datei protokollieren. Bei Langzeitmessungen genügt zur Gewährleistung der Vertrauenswürdigkeit der Messdaten (diese ist abhängig von der Zahl der Stichproben) die Entnahme einer Stichprobe pro Sekunde bezogen auf ein Auswerte-Teilintervall von einer Stunde:

Stichprobenintervall (SAMPLING-PERIOD)	1000 Millisekunden
Messintervall (OFFLINE-PERIOD)	5 Minuten
Auswerte-Teilintervall	1 Stunde
Messzeitraum	von System Ready bis Shutdown

Es ist nicht notwendig, jede Session auszuwerten. Um den Überblick zu behalten, muss man aber zumindest häufig Stichproben in Form von Tagesauswertungen machen. Auswertungen über längere Zeiträume sind sehr gut dazu geeignet, Trends aufzuweisen und Entscheidungen über Investitionen besser zu fundieren. Bei Jahresauswertungen sollte man sich auf wenige Reports beschränken.

Bei ungewöhnlichen Ereignissen wie Beschwerden, Anfragen oder Konfigurationsänderungen genügt es nicht, nur den Tag des Interesses auszuwerten. Man benötigt zusätzlich die Auswertung eines anderen Tages zum Vergleich. Legt man die Grafiken beider Auswertungen nebeneinander und vergleicht die Bilder, dann gelangt man anhand der Unterschiede viel schneller zu relevanten Aussagen. Zu beachten ist, dass entsprechende Grafiken in den beiden Auswertungen in verschiedenen Maßstäben gedruckt sein können, was den Vorher-Nachher-Vergleich etwas erschwert.

SM2 wird auch eingesetzt, um Systemmessungen zu unterstützen. Deshalb liefert SM2 einige Reports, die über das Interesse an Routineüberwachungen des Systems nach Standard-Gesichtspunkten hinausgehen.

10.2.2 Bewertung charakteristischer Reports

Für Routine-Überwachungen haben sich folgende Reports des Auswerteprogramms ANALYZER als besonders aussagekräftig erwiesen:

Reportgruppe	Report	Bedeutung
CATEGORY-CPU	CPU utilization (TU + TPR) for category	CPU-Auslastung pro Kategorie ¹
CATEGORY-IO	IOs for category	DVS-Ein-/Ausgaberaten pro Kategorie ¹
	Duration of non paging IOs for category	HW-/SW-Bedienzeit pro Kategorie ¹
CATEGORY-QUEUE	Active and inactive task for category	Anzahl Tasks pro Kategorie
CHANNEL	Utilization	Kanalauslastung ²
CPU	Utilization real	CPU-Auslastung (echter Wert; auch bei VM2000)
	Sum SVC calls	SVC-Aufrufe
DISK	Utilization	Plattenauslastung (logische Volumes)
IO	IOs for device classes	DVS-Ein-/Ausgaberaten
	Paging IOs	Paging-Ein-/Ausgaberaten
MEMORY	Paging area utilization	Auslastung Pagingarea
	Page frames	Größe des System-Working-Set
PERIODIC-TASK-JOBNAME		Verbrauchsdaten pro Jobname:
	CPU time	CPU-Anteil
	IO	Ein-/Ausgaberaten
PERIODIC-TASK-TSN		Verbrauchsdaten pro TSN:
	CPU time	CPU-Anteil
	IO	Ein-/Ausgaberaten
PERIODIC-TASK-USERID		Verbrauchsdaten pro Benutzerkennung:
	CPU time	CPU-Anteil
	IO	Ein-/Ausgaberaten
VM2000	VM2000 utilization	Lastanteile der VM2000-Gastsysteme ³
	VM2000 Hypervisor	VM2000-Hypervisor-Anteil ⁴
WORKING-SET	Main memory utilization	verfügbare Hauptspeicherseiten

¹ mit Messprogramm SYSTEM

² bei Kanal Typ FC nur mit Messprogramm CHANNEL-IO

³ mit Messprogramm VM (die Werte für alle Gastsysteme werden nur auf dem Monitorsystem geliefert)

⁴ mit Messprogramm VM (nur auf S-Servern)

10.2.3 Anmerkung zu verschiedenen Reports

Reports der Reportgruppe BCAM-CONNECTION

Diese Reports liefern dann sinnvolle Ergebnisse, wenn es um den Anteil der Antwortzeiten geht, der im Server verursacht wird. Die Übertragungszeiten der Nachrichten und ggf. Staus im Netz werden nicht erfasst. Daher sind diese Reports nur dazu geeignet, die Lastsituation im Zentralrechner zu beobachten. Aussagen über die vom Anwender erlebten Antwortzeiten (siehe [Kapitel „Leistungserwartungen aus Anwendersicht“ auf Seite 15](#)) sind nicht möglich.

Report „CPU utilization (TU+TPR) for category“ der Reportgruppe CATEGORY-CPU

Dieser Report und der nachfolgend beschriebene Report der Reportgruppe CATEGORY-IO sind nur bei Einschaltung des SM2-Messprogramms SYSTEM zu erhalten. Der Report dient dazu, die Verursacher-Kategorie für eine zu hohe CPU-Auslastung zu finden oder die Aufteilung der CPU auf die verschiedenen Kategorien zu überwachen. Letzteres ist ein Hilfsmittel zur Einstellung von PRIOR bzw. PCS.

Report „IOs for category“ der Reportgruppe CATEGORY-IO

Dieser Report dient der Überwachung der Anzahl der Ein-/Ausgaben für die Kategorien. Zur Systemsteuerung ist er kaum einzusetzen, da die Anzahl der Ein-/Ausgaben der Kategorie bei der Steuerung der Kategorien nicht berücksichtigt wird. Er ist also der Einstieg für die Lösung eines Leistungsproblems. Für eine tieferegehende Engpass-Analyse kann der Betriebsmittelbedarf (CPU-Zeit, Anzahl Ein-/Ausgaben) mit dem Messprogramm TASK bis auf Taskebene verfolgt werden.

Report „Utilization“ der Reportgruppe CHANNEL

Bei Kanälen mit angeschlossener Plattenperipherie sollte eine Auslastung von 60% nicht überschritten werden.

Beim Kanal Typ FC muss zur Erfassung der Auslastung das Messprogramm CHANNEL-IO eingeschaltet sein. Die Auslastung kann beim Kanal Typ FC nicht direkt gemessen werden, sondern wird indirekt bestimmt, indem die aktuelle Anzahl Ein-/Ausgabe-Operationen unter Berücksichtigung der Blockgröße mit einem Durchsatzwert verglichen wird, der bei Labor-messungen maximal erreicht werden konnte.

Kanalauslastung bei VM2000-Betrieb:

- Kanal Typ S: Da nur die gesamte Auslastung eines Kanal gemessen werden kann, ist es nicht möglich die Auslastungsanteile der einzelnen Gastsysteme zu ermitteln. In jedem Gastsystem wird also immer die Gesamtauslastung eines Kanals durch alle Gast-systeme ausgegeben.

- Kanal Typ FC: Nur bei eingeschaltetem Messprogramm CHANNEL-IO wird die Auslastung ermittelt und zwar über die gastsystem-spezifische Ein-/Ausgaberate. Die Ausgabe der Auslastung erfolgt daher gastsystem-bezogen. Der Summenwert über alle Gast-systeme sollte 60% nicht übersteigen.

Kanäle für Magnetbandkassetten dürfen bis 100% belastet werden.

Zu hohe Kanalbelastungen können anzeigen, dass zuviele Geräte an einem Kanal hängen. Aber auch die Verwendung der Seitenkettung treibt die Kanalbelastung hoch. Deshalb sollte diese Funktion nur benutzt werden, wenn sie erkennbare Vorteile bringt. Das ist immer dann der Fall, wenn alle mit einer Ein-/Ausgabe übertragenen Blöcke benötigt werden und damit weitere Ein-/Ausgaben eingespart werden.

Reports der Reportgruppe CPU

Im Report „Utilization real“ wird die gemessene CPU-Auslastung wiedergegeben (auch bei VM2000-Betrieb). Der Report „Utilization normed“ sollte nicht verwendet werden, insbesondere nicht bei VM2000-Betrieb!

Die CPU-Auslastung ist je nach Anwendungsart und Lastart unterschiedlich zu beurteilen. Bei reinen Batch-Anwendungen ist eine Auslastung von 100% machbar und wünschenswert. Eine ausgewogene Auslastung der übrigen Betriebsmittel soll angestrebt werden.

Bei Mischlastbetrieb mit TP-, Dialog- und Batch-Anwendungen sollte die Hauptanwendung maximal 70% der CPU-Leistung des Servers aufnehmen (bei Multiprozessor-Servern sind, abhängig von der Anzahl CPUs, bis zu 90% für die Hauptanwendung vertretbar). Die ideale Hintergrundlast macht keine Ein-/Ausgaben und kein Paging. Sie ist extrem rechenintensiv und dadurch ein guter Verbraucher für Restkapazitäten. Solche Eigenschaften gestatten eine vollständige Nutzung der CPU, sind jedoch in der Praxis selten anzutreffen. Jede Ein-/Ausgabe und jeder Paging-Transfer können die TP- oder Dialog-Anwendungen stören. Ein-/Ausgabe-intensive Lasten und Programme mit großem Speicherbedarf sollten nicht als Hintergrundlast zugelassen werden. Die Hintergrundlast muss über Prioritäten bzw. PCS-Einstellungen gegenüber der Hauptanwendung deutlich zurückgesetzt werden. Ihre Dateien sollten idealerweise auf anderen Plattenlaufwerken (zumindest aber auf anderen Volumes) liegen und über andere Pfade erreichbar sein.

Der SIH-Anteil hängt u.a. ab von der Anzahl der Ein-/Ausgaben jeder Art und der Intensität der Task-Kommunikation. Der SIH-Anteil sollte 20% (S-Server) bzw. 10% (SQ-Server) nicht überschreiten. Hohe TPR-Anteile sind durch SVC-Aufrufe bedingt, die aufwendige Systemdienste ausführen. Die Last kann als normal angesehen werden, wenn gilt:

$$TU + TPR > 3 * SIH \text{ (S-Server) bzw. } TU + TPR > 7 * SIH \text{ (SQ-Server)}$$

Es gibt immer wieder Einzelfälle, in denen die Last die o.g. Regeln verletzt und trotzdem ein zufriedenstellendes Verhalten hat. Solange die Antwortzeiten gut sind, besteht in solchen Fällen kein Grund zum Eingreifen.

In der Regel laufen die Anwendungen auf einem SQ-Server im /390-Modus. Da Programme, die in diesem Modus laufen, einen höheren CPU-Bedarf als auf S-Servern haben, ist ein hoher Anteil im Funktionszustand TU normal (etwa 60 %, normiert auf 100 % Auslastung). Wird ein höherer normierter TU-Anteil ausgewiesen, so liegt die Leistung des SQ-Servers unter dem Nominalwert (dem offiziellen RPF-Wert). Wenn der (normierte) TU-Anteil aber kleiner ist, dann liegt die Leistung über dem Nominalwert.

Report „Time“ der Reportgruppe DISK

Der Report „Time“ ist nur verfügbar, wenn das Messprogramm SAMPLING-DEVICE mit Erfassung von Hardware- und Software-Bedienzeiten eingeschaltet ist. Aus diesem Report kann in Verbindung mit dem Report „Utilization“ ermittelt werden, ob die Plattenperipherie stark belastet ist. Treten Software-Bedienzeiten (Software time) auf, die um mehr als 10% über der Hardware-Bedienzeiten (Hardware time) liegen, dann müssen gelegentlich Ein-/Ausgaben vor einem belegten Volume warten. Mit dem SM2-Messprogramm DISK-FILE und dem SM2-Online-Report DISK-FILE können dann die betroffenen Dateien ermittelt werden.



Wenn die Anwendung mit asynchronen Ein-/Ausgaben arbeitet, so kann es vermehrt zu langen Software-Wartezeiten kommen ohne dass eine Tuning-Maßnahme nötig bzw. möglich ist.

Report „IOs for device classes“ der Reportgruppe IO

Dieser Report ist im Zusammenhang mit dem Report „Utilization real“ der Reportgruppe CPU zu beurteilen. Es gibt Lasten mit hoher Anzahl von Ein-/Ausgaben und sehr rechenintensive Lasten. Das wichtigste Kriterium ist daher, dass Kanäle und Volumes nicht überlastet sind. Die Richtwerte für die max. Anzahl DVS-Ein-/Ausgaben pro Sekunde sind nach der Regel ermittelt, dass (je nach Servergeschwindigkeit) nur ein Anteil von 15 - 25% (S-Server) bzw. 8 - 12% (SQ-Server) der CPU-Leistung im Funktionszustand SIH+TPR für die Behandlung der Ein-/Ausgaben verbraucht werden sollte. Es gibt Situationen, in denen diese Grenze sinnvoll überschritten werden kann.

Siehe auch [Abschnitt „Richtwerte für die BS2000/OSD-Server“ auf Seite 331](#).

Report „Paging IOs“ der Reportgruppe IO

Paging erlaubt die effiziente Nutzung eines relativ kleinen Hauptspeichers im Verhältnis zu den virtuellen Adressraumanforderungen.

Durch die verfügbaren kostengünstigen Hauptspeicher wird der Hauptspeicher-Ausbau i.d.R. so groß gewählt, dass dynamisches Paging kein Problem darstellt. Der folgende Text ist daher nur in Sonderfällen zu beachten.

Paging ist die Folge von Speichermangel. Paging erzeugt Last auf der Peripherie und auf dem Prozessor. Deshalb sollten die im [Abschnitt „Richtwerte für die BS2000/OSD-Server“ auf Seite 331](#) angeführten Paging-Raten (Page-Reads pro Sekunde) nicht überschritten werden. Eine Überschreitung der empfohlenen Maximalwerte ist nur zulässig, wenn die Antwortzeiten gut sind **und** alle übrigen Auslastungen (CPU, Kanäle, Volumes) niedrig liegen.

Bei Engpässen am Kanal oder den Paging-Volumes kann es sein, dass die genannten Paging-Raten nicht erreicht werden und dennoch die Antwortzeiten unbefriedigend sind. Verursacht wird dies durch zu lange Wartezeiten bei PAGE READ. Eine Fehlkonfiguration der Paging-Peripherie hat schwere Auswirkungen und muss unbedingt vermieden werden. Sie ist leicht an den Reports „Utilization“ der Reportgruppen CHANNEL und DISK zu erkennen.

TP-Betrieb ist sehr empfindlich gegenüber Paging – dies umso mehr, je mehr wichtige Arbeiten von wenigen zentralen Tasks gemacht werden. Solche Drehscheiben dürfen keine Paging-Ein-/Ausgaben erleiden, da sich die dadurch entstehende Verzögerung auf viele andere Tasks überträgt. Paging neigt dazu, sich schlagartig zu verstärken. Es ist deshalb nicht empfehlenswert, den Hauptspeicher eines Servers voll auszureizen (siehe auch die Anmerkungen zum Report „Main memory utilization“ der Reportgruppe WORKING-SET sowie die Abschnitte [„Überwachen der Hauptspeichernutzung“ auf Seite 31](#) und [„Paging-Aktivitäten“ auf Seite 299](#)).

Report „Paging area utilization“ der Reportgruppe MEMORY

Der maximale Füllstand der Paging Area sollte 50% betragen.

Wenn der mittlere Füllstand des Seitenwechselbereichs nicht größer als 50% ist, dann können Paging-Spitzenlasten mit akzeptablen Paging-Ein/Ausgabezeiten abgefangen werden. Es stehen dann genügend große zusammenhängende Bereiche für das performante Auslagern von geänderten Speicherseiten zur Verfügung.

Spätestens ab einem mittleren Füllgrad von 70% tritt bei Paging-Spitzenlasten eine erhebliche Antwortzeit-Verlängerung ein.

Report „Page frames“ der Reportgruppe MEMORY und „Main memory utilization“ der Reportgruppe WORKING-SET

Der Wert „Available pages (NPP)“ gibt an, wieviel vom realen Hauptspeicher-Ausbau für den Seitenwechsel noch zur Verfügung steht

Seitenwechselvorgänge beginnen aufzutreten, wenn der Wert „Number of page frames for system global wset“ (SWS) etwa 80 - 90% von NPP erreicht.

Steigt der Wert von SWS auf Werte größer 90% von NPP, so sollte an eine Hochrüstung des Hauptspeichers gedacht werden. Bei VM2000-Betrieb reicht hier oft (zumindest als Übergangsregelung) den Hauptspeicher zwischen den Gastsystemen besser aufzuteilen. Bei diesen hohen SWS-Werten wird davor gewarnt neue Anwendungen aufzunehmen oder Versionswechsel häufig verwendeter Software vorzunehmen. Es droht sonst die Gefahr einer deutlichen Steigerung der Paging-Rate!

Reports der Reportgruppen PERIODIC-TASK-...

Diese Reports sind sehr nützlich für die Analyse möglicher Verursacher von Ressourcen-Engpässen (insbesondere CPU und IO). Je nach Anwendung ist es sinnvoller die Analyse auf Basis der Benutzererkennung, des Job-Namens oder der TSN vorzunehmen.

Reports der Reportgruppe VM2000

S-Server

Aus Messungen auf dem Monitorsystem kann man mit dem Report „VM2000 Utilization“ die CPU-Auslastung aller Gastsysteme auswerten. Addiert man dann den Wert aus dem Report „VM2000 Hypervisor“ dazu, so ergibt sich die Gesamtauslastung des Servers.

SQ-Server

Die Gesamtauslastung des Servers kann über die Auslastung der einzelnen CPU-Pools bestimmt werden. Diese werden im Monitorsystem im Report „CPU pool utilization“ geliefert. Die dort dargestellten Werte der CPU-Pool-Auslastung beziehen sich auf die CPU-Auslastung aller CPUs des jeweiligen Pools. Sie zeigen also nicht die CPU-Auslastung bezogen auf alle verfügbaren CPUs des Servers. Sie können daher nicht einfach addiert werden, sondern sind nach folgender Formel zu „normieren“:

```
Utilization [%] for CPU pool * Number of attached real CPUs for CPU pool  
/ Total number active CPUs
```

Die Gesamtauslastung des Servers ist die Summe aller normierten Werte der verfügbaren CPU-Pools.

Wenn neben BS2000/OSD-Gastsystemen auch Linux- und Windows-Gastsysteme vorhanden sind, dann muss die durch diese Gastsysteme erzeugte Auslastung zusätzlich berücksichtigt werden. Dazu werden die Reports der Reportgruppe „Pool“ des Systemtyps „Xen“ benötigt. Diese Reports erhalten Sie über das Softwareprodukt openSM2 (Open Systems).

Unter VM2000 kann (je nach Anzahl und Last der Gastsysteme) die BS2000-CPU um 5-15% höher ausgelastet werden als im Native-Betrieb.

10.3 Durchführung einer Engpass-Analyse

10.3.1 Prinzipielle Vorgehensweise

Vor dem Einstieg in die Einzelheiten der Durchführung soll zunächst nochmals auf die grundlegende Problematik bezüglich der Erfüllung der Leistungserwartungen hingewiesen werden:

Leistungserwartungen des einzelnen Anwenders

Die Leistungserwartungen des einzelnen Anwenders sind zeitlicher Natur (Antwortzeit, Verweilzeit). Die Zeitdauer für die Bearbeitung seiner Anforderungen an das IT-System wirkt sich direkt auf den Arbeitsfortgang und damit auf die „Produktivität“ des Anwenders aus.

Die Bearbeitungszeitdauer setzt sich aus der Bedienzeit und der Wartezeit auf die entsprechenden Betriebsmittel zusammen. Während die Bedienzeit den unmittelbaren Betriebsmittelbedarf widerspiegelt, hängt die Wartezeit von der Auslastung der Betriebsmittel ab. Die Leistungserwartungen des Anwenders lassen sich daher umso leichter erfüllen, je geringer die Auslastung der Betriebsmittel ist.

Leistungserwartungen des Server-Betreibers

Die Leistungserwartungen des Server-Betreibers sind aus wirtschaftlichen Gründen auf die Erzielung eines maximalen „Systemdurchsatzes“ (hohe Transaktions- bzw. Job-Durchsatzrate) ausgerichtet, worunter sich implizit die Forderung nach höchstmöglicher Auslastung der Betriebsmittel verbirgt.

Zur Erfüllung der Leistungserwartungen sowohl des einzelnen Anwenders als auch der Systembetreuung ist also immer ein Kompromiss erforderlich.

Trotz der verständlichen Forderung der Investitionsplaner nach höchstmöglicher Auslastung der Betriebsmittel haben sich in der Praxis verschiedene Richtwerte für die Auslastung der Betriebsmittel herauskristallisiert, die nach Möglichkeit nicht überschritten werden sollten.

Diese Richtwerte sind in den meisten Fällen keine technischen Grenzwerte. Ihre Überschreitung führt zu einer ineffizienten Nutzung des IT-Systems und erschwert die Erfüllung der Leistungserwartungen der einzelnen Anwender.

Maßgebend für die Vorgehensweise ist daher vor jeder Messung die Klärung, welche Leistungserwartungen nicht erfüllt werden:

- Systemorientierte Leistungserwartungen (der Systemdurchsatz lässt zu wünschen übrig → die Leistung des **gesamten IT-Systems** ist unbefriedigend)

Kennzeichen dafür sind:

- zu niedrige Transaktionsraten (als unmittelbares Ergebnis eines generell unbefriedigenden Antwortzeitverhaltens)
- zu niedrige Durchsatzraten (allgemein zu lange Verweilzeiten der Aufträge)

- Anwenderorientierte Leistungserwartungen (die Leistungserwartungen **einzelner Anwender** werden nicht erfüllt)

Kennzeichen dafür sind:

- zu hohe Antwortzeiten für bestimmte Typen von Transaktionen
- zu lange Verweilzeiten für bestimmte Aufträge

Liegt ein **systemorientiertes** Leistungsproblem vor, so sind die Ursachen dafür mit größter Wahrscheinlichkeit in der Überlastung eines oder mehrerer Betriebsmittel zu finden.

Beim Auftreten eines **anwenderorientierten** Leistungsproblems gilt es, die Gründe für zeitliche Verzögerungen bei der Behandlung spezieller Lastanforderungen herauszufinden.



Da der Einsatz zusätzlicher Messprogramme zu einem Ressourcen-Mehrbedarf führt (i.W. CPU-Zeit) sollte darauf geachtet werden, dass die Messprogramme nach erfolgreicher Messung wieder abgeschaltet bzw. aus den Start-Prozeduren entfernt werden.

10.3.1.1 Parameter für die Messung und Auswertung

Engpass-Analyse-Messungen werden i.A. zur Zeit der Spitzenbelastung durchgeführt. Zur Erleichterung der Analyse wird die Erfassung möglichst vieler Messdaten angestrebt. Der Messzeitraum sollte eine halbe Stunde bis maximal 1 Stunde betragen.

Als Bezugsgrundlage für die Beurteilung der Messwerte ist ein Zeitraum von 15 Minuten zugrunde zu legen (Mittelwert über 3 Messwerte-Teilintervalle von 5 Minuten).

Die Messdatenerfassung durch SM2 erfolgt teils ereignisbezogen, teils im Stichprobenverfahren. Um die Vertrauenswürdigkeit der Stichproben sicherzustellen, ist eine gewisse Anzahl Messwerte pro Auswerte-Teilintervall erforderlich (mindestens 1500). Es wird ein Stichprobenintervall von 400 Millisekunden empfohlen.

Pro Messintervall erfolgt durch SM2 die Mittelwertbildung über die erfassten Messwerte. Um die Schwankungsbreiten pro Auswerte-Teilintervall transparent zu machen, ist es günstig, dass sich jedes Auswerte-Teilintervall aus mehreren Messintervallen zusammensetzt.

Empfohlene Messparameter

Stichprobenintervall (SAMPLING-PERIOD)	400 Millisekunden
Messintervall (OFFLINE-PERIOD)	60 Sekunden
Auswerte-Teilintervall	5 Minuten
Messzeitraum	0,5 - 1 Stunde

Bei Engpass-Analyse-Messungen steht die **Ausgabe der Messwerte in die Messwertedatei** gegenüber der Ausgabe auf Bildschirm im Vordergrund. Es wird also die Datei-Auswertung mit ANALYZER der Online-Auswertung mit INSPECTOR vorgezogen.

10.3.1.2 Messwerte für die Untersuchung von systemorientierten Leistungsproblemen

Für das Auffinden der überlasteten Betriebsmittel sind folgende Daten erforderlich:

- Auslastungswerte: CPU, Kanäle, Geräte (Platten), Hauptspeicher
Bei Platten sieht SM2 ausschließlich logische Volumes. Ist das Softwareprodukt SHC-OSD im Einsatz, liefert das Messprogramm STORAGE-SYSTEM zusätzlich Daten über die physikalischen Laufwerke von Symmetrix-Systemen.
- Anzahl Ein-/Ausgaben pro Gerät
- Anzahl Tasks pro Geräte-Warteschlange
- Working-Set-Bedarf pro Task-Kategorie

Um die Abhängigkeit des Antwortzeitverhaltens von der Betriebsmittelauslastung zu verdeutlichen, ist die parallele Führung folgender Statistiken günstig:

- verbindungsspezifische Antwortzeit-Statistik (Messprogramm BCAM-CONNECTION)

Zusätzlich wird die Einschaltung folgender Messprogramme empfohlen:

- erweiterte System-Statistik (Messprogramm SYSTEM)
Dieses Messprogramm liefert kategoriespezifisch u.a. Dehnungen, Verweilzeiten in System-Warteschlangen, Aussagen zur Hardware- und zur Software-Bedienzeit (HARDWARE und SOFTWARE DURATION) von DVS- und Paging-Ein-/Ausgaben.
- erweiterte Geräte-Statistik (Messprogramm SAMPLING-DEVICE)
Das Messprogramm wird beim Aufruf von SM2 automatisch gestartet, jedoch ohne die Erfassung von Hardware- und Software-Bedienzeit pro Volume. Sollen diese Messwerte ebenfalls in die Messung einfließen, muss das Messprogramm mit den Anweisungen `//SET-SAMPLING-DEVICE-PARAMETER DISK-SERVICETIME=*ON` und `//CHANGE-MEASUREMENT-PROGRAM TYPE=*SAMPLING-DEVICE` neu gestartet werden.
- Kanal-Statistik (Messprogramm CHANNEL-IO)
Mit diesem Messprogramm erhält man die Anzahl Ein-/Ausgaben sowie die Anzahl übertragener PAM-Blöcke pro Kanal.
- Katalog-Statistik (Messprogramm CMS)
Dieses Messprogramm gibt Aufschluss über die Anzahl, Art und Zeitdauer der Zugriffe auf Katalogeinträge pro Pubset.
- VM2000-Statistik (Messprogramm VM)
Dieses Messprogramm sollte bei VM2000-Betrieb nur im Monitor-System gestartet werden (dort aber standardmäßig). Es liefert im Monitorsystem den CPU-Anteil aller Gastsysteme sowie auf S-Servern den Anteil an Hypervisor-Active und -Idle.

10.3.1.3 Messwerte für die Untersuchung von anwenderorientierten Leistungsproblemen

Für die Aufdeckung von zeitlichen Verzögerungen sind neben den oben erwähnten globalen Systemauslastungswerten zusätzliche Daten notwendig. Als Einstieg kann mit der periodischen Task-Statistik (Messprogramm PERIODIC-TASK) die entsprechende Task-Auswahl vorgenommen werden.

Für die weitere Analyse sind folgende Messgrößen erforderlich:

- taskspezifische Ermittlung des Betriebsmittelverbrauchs (Messprogramm TASK)
Das Messprogramm liefert:
 - CPU-Zeitbedarf
 - Anzahl SVC-Aufrufe
 - Anzahl Ein-/Ausgaben und Ein-/Ausgabezeiten, aufgeschlüsselt auf die im Messprogramm definierten Volumes
 - Dauer der Wartezustände
 - Paging-Aktivitäten
 - Nachrichtenlängen pro Terminal-Ein-/Ausgaben

Für die weitergehende Analyse des CPU-Zeitbedarfs einzelner Tasks in Form einer PC-Statistik und SVC-Statistik wird eine zusätzliche Benutzer-Task-Messung mit anschließender Auswertung durch SM2-PA (Programm-Analysator) empfohlen.

- SVC-Statistik (Messprogramm SVC)
Das Messprogramm zeigt die Aufrufhäufigkeit jedes SVC getrennt nach TU und TPR.
- DAB-Trefferrate (Messprogramm DAB)
Das Messprogramm liefert die Anzahl der Zugriffe pro spezifiziertem DAB-Teilbereich.
- Wartezeiten vor einzelnen Geräten (Messprogramm SERVICETIME)
- ISAM-Pool-Trefferrate (bei Einsatz von NK-ISAM; Messprogramm ISAM)
Das Messprogramm liefert die Anzahl der Zugriffe pro spezifiziertem ISAM-Pool.
- Anzahl Zugriffe auf bestimmte Dateien (Messprogramm FILE)
- UTM-Statistik (Messprogramm UTM)
Das Messprogramm liefert applikationsspezifische Daten, u.a. Verweilzeit in UTM, Aufteilung der Verweilzeit auf Datenbankanteil, TAC-Klassen-Wartezeit und Zeit für verteilte Verarbeitung. Zusätzlich werden durchschnittliche Verbrauchswerte für CPU und Ein-/Ausgaben in UTM und im DB-System (nur SESAM/SQL und UDS) sowie die Anzahl der DB-Calls pro Dialogschritt bzw. Asynchronvorgang geliefert.

Unter den folgenden Voraussetzungen werden UTM-Messwerte ausgegeben:

- Das Subsystem UTM-SM2 muss gestartet sein.
- Der Start von UTM-SM2 erfolgt automatisch, wenn beim KDCDEF-Lauf der Operand `MAX SM2=ON` angegeben ist.
Nachträglich wird UTM-SM2 über die UTM-Administrationsschnittstelle mit `KDCAPPL` und `SM2=ON` gestartet, zusammen mit der Bereitschaft zur Messwertübergabe.
- Das SM2-Messprogramm UTM muss eingeschaltet sein.
- Die UTM-Applikation muss bereit sein, Daten an SM2 zu liefern.
Beim KDCDEF-Lauf, Operand `MAX SM2=ON/OFF/NO`, kann die Messwertübergabe wie folgt beeinflusst werden:
 - Bei `SM2=OFF` (Standardeinstellung) kann über die UTM-Administrationsschnittstelle mit `KDCAPPL` und `SM2=ON` die Messwertübergabe pro Applikation nachträglich eingeschaltet werden.
 - Bei `SM2=ON` ist keine zusätzliche Administration erforderlich.
 - Bei `SM2=NO` wird die Messwertübergabe generell verhindert und kann auch nachträglich nicht eingeschaltet werden.
- DB-spezifische Verbrauchswerte (Ein-/Ausgaben, CPU-Zeit) werden nur unter folgenden Bedingungen geliefert:
 - Das BS2000-Accounting ist aktiv.
 - Der UTM-Abrechnungssatz UTMA ist eingeschaltet (mit `/MODIFY-ACCOUNTING-PARAMETERS, ADD-RECORD-TYPE=UTMA`).
 - Das UTM-Accounting ist eingeschaltet (`KDCAPPL, Parameter ACC=ON`).
 - Bei SESAM/SQL wird zusätzlich mit der Anweisung `ACC, TP=ON, CPU` die Messwert-Erfassung aktiviert.

Weitere Informationen sind im Handbuch „openUTM – Konzepte und Funktionen“ [19] und im Handbuch „Einsatz von openUTM-Anwendungen unter BS2000/OSD“ [22] zu finden.

- **Datenbank-Statistik**
Die Messprogramme SESAM-SQL und UDS-SQL liefern Messdaten zu den Datenbanksystemen SESAM/SQL und UDS/SQL.

Voraussetzung in SESAM/SQL:

Zur Übergabe der Statistikdaten von SESAM/SQL an openSM2 starten Sie SESMON im Batch-Betrieb:

```
/START-SESAM-PERF-MONITOR
//SET-MONITOR-OPTIONS . . . ,OUTPUT=*SM2
```

Bei `OUTPUT=*SM2` kann pro SESMON-Instanz nur ein DBH angegeben werden. Zur Ausgabe von Daten weiterer DBHs muss jeweils eine neue SESMON-Instanz gestartet werden.

Das Zeitintervall, mit dem SESMON die Daten an openSM2 übergibt, wird automatisch auf ca. 30% des SM2-Messintervalls eingestellt. Eine manuelle Einstellung ist nicht möglich.

Voraussetzung in UDS/SQL:

Die Übergabe der Statistikdaten von UDS/SQL an openSM2 wird entweder beim Start des UDS-Monitors mit der Anweisung `MEDIUM=S,n` oder im laufenden Betrieb mit dem Kommando `/INFORM-PROGRAM MSG='ADD MEDIUM=S,n'` eingeleitet. Sie kann mit dem Kommando `/INFORM-PROGRAM MSG='FINISH MEDIUM=S'` wieder beendet werden.

Mit `n` wird das Zeitintervall in Sekunden ($5 \leq n \leq 999$) definiert, mit dem der UDS-Monitor die Daten an SM2 übergibt. Es sollte deutlich kleiner als das in SM2 eingestellte Messintervall gewählt werden, damit innerhalb eines SM2-Messintervalls mehrmals Daten übergeben werden.

Die Messwerte werden von den Datenbanksystemen asynchron an openSM2 geliefert und gelten für ein oder mehrere vom Datenbanksystem festgelegte Intervalle, die nicht exakt mit dem SM2-Intervall übereinstimmen müssen. Hierbei kann es sowohl Unterschiede bei der Dauer der Intervalle als auch zeitliche Verschiebungen zwischen den Datenbanksystem- und den SM2-Intervallen geben.

Für die Normierung der Messwerte auf eine Sekunde wird die Dauer des oder der Datenbanksystem-Intervalle herangezogen. Die Werte sind also exakt, aber sie passen nur bedingt zum SM2-Intervall.

- SYMMETRIX-Statistik (Messprogramm STORAGE-SYSTEM)
Wenn das Softwareprodukt SHC-OSD im Einsatz ist, dann werden die Hit-Raten von Symmetrix-Systemen erfasst. Zusätzlich zu den Daten über logische Volumes, werden auch Informationen über die physikalischen Laufwerke geliefert.



Zur Überwachung der Plattenspeichersysteme ETERNUS DX benötigen Sie das Softwareprodukt openSM2 (Open Systems).

- Information über TCP/IP-Verbindungen (Messprogramm TCP-IP)
Pro Verbindung wird die IP- und PORT-Nummer samt Anzahl übertragener „Transport Servie Data Units“ geliefert.
- Anzahl Zugriffe und übertragene Daten von Netzgeräten (Messprogramm SAMPLING-DEVICE)

- Kommunikation im Rechnerverbund (Messprogramm MSCF)
Es werden Daten über die Kommunikation des lokalen Rechners mit anderen Rechnern geliefert.
- Daten zu Basisfunktionen im HIPLEX-Verbund (Messprogramm NSM)
- Statistik über Lockanforderungen aus TU, TPR und NSM
(Messprogramm DLM)
- POSIX-Statistik (Messprogramm POSIX)
Es werden Messwerte über die Nutzung verschiedener POSIX-Funktionen geliefert.
- Anzahl Zugriffe zum Globalspeicher (Messprogramm GS)
Es werden Lese- und Schreibzugriffe pro GS-Partition angezeigt.
- detaillierte Platten-Statistik (Messprogramm DISK)
Dieses Messprogramm verliert mit dem fast ausschließlichen Einsatz von Plattenspeichersystemen mit Cache seine ursprüngliche Bedeutung. Ursache ist die Entkopplung des logischen Volumens vom physikalischen Gerät (siehe [Abschnitt „Konfigurierung in BS2000/OSD“ auf Seite 52](#)).

10.3.1.4 Auswertung einer SM2-Messung

Bei der Auswertung mit ANALYZER (bzw. mit SM2R1) empfiehlt sich ein schrittweises Vorgehen:

1. Grobauswertung

Engpass-Suche durch zeitliche Eingrenzung (Auswerte-Teilintervall = 15 min.)

Auffinden jener Auswerte-Teilintervalle, in denen die empfohlenen Richtwerte überschritten werden.

Eine wertvolle Hilfe für das Aufspüren von Engpässen stellt die im ANALYZER (bzw. in SM2R1) vorhandene **automatische Leistungsanalyse** dar. Diese Funktion vergleicht die in der SM2-Datei gesammelten Messwerte mit den im folgenden Abschnitt empfohlenen Richtwerten und gibt eine entsprechende Meldung an den Anwender aus.

Wichtig ist, dass ausschließlich kritische Zeiträume des Produktivbetriebes für die automatische Leistungsanalyse herangezogen werden.

2. Feinauswertung

Untersuchen der kritischen Auswerte-Teilintervalle (15 min) durch Herabsetzen des Auswerte-Teilintervalls (TIME-STEPS) auf die Größe des Messintervalls (OFFLINE-PERIOD), um die Schwankungsbreite der Messwerte zu verdeutlichen.

Zur leichteren Beurteilung der Messergebnisse sollte darauf geachtet werden, dass sowohl die Messwerte über Zeitperioden vorliegen, in denen die Leistungserwartungen nicht erfüllt werden, als auch über Zeiträume, in denen die Leistungserwartungen erfüllt werden. Durch den Vergleich beider Fälle wird die Zuordnung der Probleme zu den Ursachen wesentlich erleichtert.

Im folgenden Abschnitt wird versucht, durch die Angabe von Richtwerten für die Auslastung von Betriebsmitteln die Interpretation von Messwerten zu erleichtern.

10.3.2 Untersuchen eines systemorientierten Leistungsproblems

Zum besseren Verständnis der angegebenen Richtwerte für die Betriebsmittelauslastung soll zunächst eine Quantifizierung der Arbeit von BS2000/OSD vorgenommen werden.

Systemzeit (SIH-Zeit) wird i.W. für die Bearbeitung folgender Ereignisse benötigt:

- Steuerung des Multiprogramming-Betriebs
- Durchführung DVS-Ein-/Ausgaben
- Durchführung Paging

Bei voll ausgelasteter CPU gelten folgende Richtwerte:

Der SIH-Anteil sollte insgesamt 20% (S-Server) bzw. 10% (SQ-Server) nicht übersteigen – auch bei starker DVS-Ein-/Ausgabe-Tätigkeit.

Der Aufwand für die Steuerung des Multiprogramming-Betriebs bewegt sich bei normalen Lasten zwischen 5 - 10% (S-Server) bzw. 3 - 5% (SQ-Server) der jeweiligen CPU-Auslastung.

Bei üblicher Last und ausgewogener Konfiguration beträgt der SIH-Anteil für die

- Durchführung der DVS-Ein-/Ausgaben: 3 - 6% (S-Server) bzw. 2 - 4% (SQ-Server)
- Durchführung des Paging: max. 2% (S-Server) bzw. max. 1% (SQ-Server)

Bei der Interpretation der mit SM2 gemessenen Werte wird empfohlen, in der Reihenfolge der anschließend beschriebenen Abschnitte vorzugehen.

10.3.2.1 Auslastung der Ein-/Ausgabe-Peripherie

Richtwerte für die DVS-Ein-/Ausgaberate

Im [Abschnitt „Richtwerte für die BS2000/OSD-Server“ auf Seite 331](#) sind Anhaltspunkte für die maximal sinnvolle Ein-/Ausgaberate in Abhängigkeit vom CPU-Typ zu finden. Diese Richtwerte sind bestimmt nach der Regel, dass nicht mehr als ein gewisser Anteil der CPU-Leistung (bei S-Servern 15 - 25% bzw. bei SQ-Servern 12 - 16% im Funktionszustand SIH+TPR) für die Einleitung und Behandlung von Ein-/Ausgaben verbraucht werden soll. Es ist möglich, dass die Richtwerte in einzelnen Fällen deutlich überschritten werden. Dies kann durchaus eine sinnvolle Anwendung sein. Bei Überschreiten der empfohlenen Werte sollte stets durch Einzelprüfung entschieden werden, ob ein Leistungsproblem vorliegt.

ANALYZER-Reports zur Beurteilung der DVS-Ein-/Ausgaberate

Reportgruppe	Report	Bedeutung
IO	IOs for device classes	Anzahl DVS-Ein-/Ausgaben/s
CATEGORY-IO	IOs for category	Anzahl DVS-Ein-/Ausgaben/s je Task-Kategorie
STORAGE-SYSTEM-SYMMETRIX	Reads	Anzahl Lesezugriffe/s pro Symmetrix-System
	Writes	Anzahl Schreibzugriffe/s pro Symmetrix-System
	Reads + writes	Gesamtanzahl Zugriffe/s pro Symmetrix-System
	Sequential reads	Anzahl sequenzielle Lesezugriffe/s pro Symmetrix-System



Mit SHC-OSD werden die Daten für Symmetrix-Systeme mit dem Messprogramm STORAGE-SYSTEM erfasst (Reports der Reportgruppe STORAGE-SYSTEM-SYMMETRIX).

Zur Überwachung der Plattenspeichersysteme ETERNUS DX benötigen Sie das Softwareprodukt openSM2 (Open Systems).

*Tuning-Ansätze zur Reduzierung der DVS-Ein-/Ausgaberate
(nicht nach allgemeiner Wichtigkeit sortiert!)*

- Reorganisation der ISAM-Dateien
Verringerung der Anzahl Indexstufen durch organisatorische Maßnahmen (Aufteilung der Datenbestände, Überprüfung auf Aktualität der Daten).

- Umstellung auf NK-ISAM-Dateien und damit Einsparung von Ein-/Ausgaben durch den Einsatz selbst konfigurierender bzw. benutzerspezifisch einstellbarer ISAM-Pools
Mit Hilfe der Reports der Reportgruppe ISAM-POOL bzw. ISAM-FILE kann die Nutzung der ISAM-Pools überwacht werden:

Report	Messgröße	Bedeutung
Fix operations	Number of fix-hit operations	Anzahl ISAM-Zugriffe/s, die aus dem ISAM-Pool befriedigt wurden
	Number of fix-IO operations	Anzahl ISAM-Zugriffe/s, die zu einer Ein-/Ausgabe auf Platte (Lesezugriff) geführt haben
	Number of fix-wait operations	Anzahl ISAM-Zugriffe/s, bei denen auf das Freiwerden von Pool-Seiten gewartet werden musste
Slot operations	Number of reserve slot wait operations	Anzahl der Wartezustände/s aufgrund von Engpässen bei der internen Pool-Verwaltung infolge eines zu kleinen Pools
Used pages	Number of total pages	Anzahl PAM-Blöcke des ISAM-Pools, die nach Abzug des Platzbedarfs für Verwaltungsdaten zur Pufferung zur Verfügung stehen
Index operations	Number index operations	Anzahl der Index-Zugriffe insgesamt
	Number index hit operations	Anteil der Index-Zugriffe, die aus dem ISAM-Pool befriedigt werden

Die Größe des ISAM-Pools ist richtig dimensioniert (genaue Berechnung siehe Handbuch „DVS-Makros“ [8]) bei einem Verhältnis:

$$\frac{\#FIX-HIT OPERATIONS}{\#FIX-HIT OPERATIONS + \#FIX-IO OPERATIONS} \geq 0,85$$

Gleichzeitig muss erfüllt sein:

Number of fix-wait operations = 0

Number of reserve slot wait operations = 0

Voraussetzung für das Anlegen entsprechend großer ISAM-Pools ist eine ausreichende Hauptspeicher-Kapazität, damit nicht der Effekt der Einsparung von DVS-Ein-/Ausgaben durch eine erhöhte Anzahl von Paging-Ein-/Ausgaben kompensiert wird.

- Erhöhung der Blockungsfaktoren (bei Platten- und Bandverarbeitung)
Die Erhöhung der Blockgröße ist immer dann ratsam, wenn sequenzielle Verarbeitung vorliegt. Bei wahlfreier Verarbeitung (PAM, ISAM) muss diese Bedingung im Einzelfall geprüft werden. Die Erhöhung der Blocklänge vermindert die Anzahl der Ein-/Ausgaben und entlastet damit das Gerät und die CPU, nicht jedoch den Kanal.
- Vergrößerung der internen Datenpuffer bei Datenbankanwendungen (siehe jeweilige Datenbank-Benutzerhandbücher)
Immer wenn durch größere interne Puffer der Benutzer-Adressraum entsprechend größer wird, muss auch genügend Hauptspeicher bereitgestellt werden.
- Verringerung der Ein-/Ausgaberate auf die KDCFILE bei UTM-Anwendungen durch Vergrößerung des UTM-Cache-Bereichs (siehe Handbuch „openUTM Anwendungen administrieren“ [20]).
- Einsatz von DAB (ADM-PFA)
Durch das Lesen von bis zu 16 PAM-Blöcken (32 PAM-Blöcken bei AutoDAB) ergibt sich bei überwiegend sequenzieller Verarbeitung eine deutliche Einsparung an Ein-/Ausgaben. Voraussetzung ist ein genügend großer Hauptspeicher und eine entsprechende CPU-Reservekapazität (durch Reduzierung der zeitlichen Verzögerungen bei der physikalischen Ein-/Ausgabe erhöht sich die CPU-Intensität; zur Handhabung siehe Handbuch „DAB“ [5]).
Mit Hilfe der Reports „Reads for internal area“ und „Writes for internal area“ der Reportgruppe DAB kann die Trefferrate auf die spezifizierten DAB-Teilbereiche überprüft werden. Bei Einsatz des AutoDAB gibt /SHOW-DAB-CACHING einen Überblick über die Dateien, die aktuell dem Caching unterliegen. Siehe auch den [Abschnitt „ADM-PFA-Caching“ auf Seite 180](#).
- Einsatz von HIPERFILEs unter Nutzung der Cache-Medien Haupt- und Globalspeicher (siehe [Abschnitt „Arbeiten mit HIPERFILEs“ auf Seite 176](#))

Liegt der gemeinsame Wert für die DVS-Ein-/Ausgaberate (siehe den [Abschnitt „Richtwerte für die BS2000/OSD-Server“ auf Seite 331](#)) unter dem angegebenen Richtwert, so ist als nächster Schritt die Überprüfung der Auslastung der Kanäle und der angeschlossenen Peripheriegeräte erforderlich.

Richtwerte für die Kanalauslastung

Bei Kanälen mit angeschlossenen Plattenspeichersystemen ETERNUS DX oder Symmetrix sollte die Auslastung 60% nicht überschreiten. Dies gilt sowohl für den Native-Betrieb als auch den Betrieb unter VM2000 (siehe auch Hinweise zum „Report „Utilization“ der Reportgruppe CHANNEL“ auf Seite 276).

Reports der Reportgruppe CHANNEL zur Beurteilung der Kanalauslastung

Report	Bedeutung
Utilization	Kanalauslastung (channel busy state)
Data transfer	Anzahl übertragener PAM-Blöcke
IOs	Anzahl Ein-/Ausgaben pro Kanal

Tuning-Ansatz bei Kanalüberlast: Rekonfigurierung

Bei Symmetrix-Systemen sind i.d.R. alle Kanäle, die zu **einer** Symmetrix-Box gehören, relativ gleichmäßig ausgelastet. Kurzfristige Überlastungen (15 Minuten) aus anwendungstechnischen Gründen sind tolerierbar.

Bei länger andauernden Überlastungen müssen mehr Kanäle bereitgestellt werden. Sind Kanäle vom Typ S im Einsatz, ist der Austausch gegen Kanäle vom Typ FC zu empfehlen, siehe [Seite 35](#).

Richtwerte für die Geräteauslastung

Platten (genauer: logische Volumes)

Werden Volumes von mehreren Anwendern gemeinsam benutzt, so ist die Höhe der Auslastung ein wesentliches Kriterium für die Wartezeit vor dem jeweiligen Volume. Bei Online-Anwendungen sollte die Wartezeit nicht höher als ein Drittel der Hardware-Bedienzeit sein.

Wesentlich ist daher die Unterscheidung, ob es sich um ein Volume handelt, das gemeinsam von mehreren Anwendern (bzw. Tasks) benutzt wird, oder ob es einem Anwender fest zugeordnet ist.

Bei gemeinsam von mehreren Anwendern (bzw. Tasks) simultan benutzten Volumes (Public Volumes, Private Volumes mit z.B. einer UTM-Anwendung mit mehreren Tasks, Shared-Private-Volumes, Shared-Public-Volumes) sollte die Auslastung 30% (bezogen auf 15 min) nicht übersteigen.

Bei Einsatz der Funktion PAV stehen pro Volume (Basis-Gerät) mehrere Geräteadressen (Alias-Geräte) zur Verfügung, über die gleichzeitig Ein-/Ausgabe-Aufträge abgewickelt werden können (siehe auch [Seite 55](#)). Eine höhere Auslastung des Basis-Gerätes (nicht

sichtbar, da SM2 den Mittelwert über Basis-Gerät und Alias-Geräten bildet) führt daher nicht zu einer größeren Wartezeit vor dem Volume. Die nach außen sichtbare Volume-Auslastung ($\hat{=}$ dem Mittelwert über Basis- und Alias-Geräten) sollte 30% nicht übersteigen.

Bei Volumes, die nur einem Anwender (bzw. einer Task) zugeordnet sind, ist eine Auslastung bis 100% vertretbar.

Magnetbandkassettengeräte

MBK-Geräte sind gewöhnlich einer Task fest zugeordnet. Daher ist eine Auslastung bis 100% vertretbar.

Hier ist darauf zu achten, dass die Auslastung nicht durch die „Verarbeitung“ allzu vieler Blocklücken in die Höhe getrieben wird.

ANALYZER-Reports zur Beurteilung der Geräteauslastung (Volumes)

Reportgruppe	Report	Bedeutung
CATEGORY-IO	Duration of non paging IOs for category	Hardware- und Software-Bedienzeit für DVS-Ein-/Ausgaben pro Task-Kategorie
	Duration of paging IOs for category	Hardware- und Software-Bedienzeit für Paging-Ein-/Ausgaben pro Task-Kategorie
DISK	Utilization	Auslastung pro Volume; Die Auslastung wird jeweils aufgeschlüsselt in die Anteile für DVS- und Paging-Ein-/Ausgaben.
	IOs	Anzahl Ein-/Ausgaben pro Sekunde und Volume
	Queue length	Durchschnittliche Warteschlangenlänge vor den einzelnen Volumes
	Data per IO	Anzahl PAM-Blöcke pro Ein-/Ausgabe und Volume
	Time	Hardware- und Software-Bedienzeit für DVS-Ein-/Ausgaben pro Volume
	Number alias devices	Anzahl Alias-Geräte pro Basis-Gerät
SERVICETIME	Duration of IOs for device	Hardware-Bedienzeit für Ein-/Ausgaben pro Volume mit Detail-Aufschlüsselung sowie der Wartezeit vor dem Volume

Reportgruppe	Report	Bedeutung
STORAGE-SYSTEM-SYMMETRIX	Pubset reads	Anzahl Lesezugriffe/s und Hit-Rate pro Symmetrix-Pubset
	Pubset writes	Anzahl Schreibzugriffe/s und Hit-Rate pro Symmetrix-Pubset
	Pubset reads + writes	Gesamtanzahl Zugriffe/s und Hit-Rate pro Symmetrix-Pubset
	Pubset sequential reads	Anzahl sequenzielle Lesezugriffe/s und Hit-Rate pro Symmetrix-Pubset
	Device reads	Anzahl Lesezugriffe/s und Hit-Rate pro Symmetrix-Device
	Device writes	Anzahl Schreibzugriffe/s und Hit-Rate pro Symmetrix-Device
	Device reads + writes	Gesamtanzahl Zugriffe/s und Hit-Rate pro Symmetrix-Device
	Device sequential reads	Anzahl sequenzielle Lesezugriffe/s und Hit-Rate pro Symmetrix-Device
	Device sequential writes	Anzahl sequenzielle Schreibzugriffe/s und Hit-Rate pro Symmetrix-Device
	Physical disk IOs	Gesamtanzahl Zugriffe/s und Schreibzugriffe/s pro physikalischem Plattenlaufwerk
	Director IOs	Gesamtanzahl Zugriffe/s und Hit-Rate pro Channel-Director



Für die Erfassung der Daten von Symmetrix-Systemen mit dem Messprogramm STORAGE-SYSTEM wird das Softwareprodukt SHC-OSD benötigt. Für die Erfassung der Daten von ETERNUS DX-Systemen wird das Softwareprodukt openSM2 (Open Systems) benötigt.

Tuning-Ansätze bei Geräteüberlast (Volumes)

Grundsätzlich ist zunächst die Überprüfung der Hardware-Bedienzeit (Report „Time“ der Reportgruppe DISK bei eingeschaltetem Messprogramm SAMPLING-DEVICE), unter Berücksichtigung der Anzahl übertragener PAM-Blöcke pro Ein-/Ausgabe, zweckmäßig.

In sehr seltenen Fällen muss die Analyse wie folgt fortgesetzt werden:

- a) Ist die Hardware-Bedienzeit unwesentlich größer als DEVICE CONNECT TIME (Messprogramm bzw. Report „Duration of IOs for device“ der Reportgruppe SERVICE-TIME), so handelt es sich um Read-Hits bzw. Fast-Write-Hits.

Die Überlastung des logischen Volumes entsteht durch die Häufigkeit der Zugriffe und führt zu Wartezeiten der Tasks vor dem Volume. In diesem Fall ist der Einsatz der Funktion PAV sinnvoll. Andernfalls ist die Auslagerung besonders häufig benutzter Dateien auf andere, weniger ausgelastete Volumes (möglichst auch auf ein anderes physikalisches Laufwerk) erforderlich.

- b) Enthält die Hardware-Bedienzeit einen deutlichen Anteil DEVICE DISCONNECT TIME (Report „Duration of IOs for device“ der Reportgruppe SERVICETIME), können folgende Ursachen vorliegen:
- Unzureichende Unterstützung des Read-Cachings wegen zu kleinem Cache (Read-Hit-Rate zu klein, Report „Reads for device“ der Reportgruppe SYMMETRIX-DEVICE bzw. Report „Device reads“ der Reportgruppe STORAGE-SYSTEM-SYMMETRIX).
 - Durch fehlende „Erholungsphasen“ bei Dauerschreiblasten können Cache-Inhalte nicht zwischenzeitlich auf Platte gesichert werden (hoher Anteil Delayed-Fast-Writes, erkennbar durch eine Write-Hit-Rate < 100%, Report „Writes for device“ der Reportgruppe SYMMETRIX-DEVICE bzw. Report „Device writes“ der Reportgruppe STORAGE-SYSTEM-SYMMETRIX).
 - Konkurrenzsituation zwischen logischen Volumes, die auf dem gleichen Plattenlaufwerk liegen (Überprüfung der Volumes pro Plattenlaufwerk mit dem Kommando /SHOW-SYMMETRIX-DEVICE-CONFIG UNIT=*BY-VOLUME (...), INF=*PHYSICAL des Softwareprodukts SHC-OSD; Kontrolle mit Report „Physical Disk IOs“ der Reportgruppe STORAGE-SYSTEM-SYMMETRIX).

- c) Ein Anteil REMAINING SERVICE TIME (Report „Duration of IOs for device“ der Reportgruppe SERVICETIME) in der Hardware-Bedienzeit weist auf folgende Ursachen hin:
- Dehnung der Hardware-Bedienzeit durch andere Gastsysteme unter VM2000. In diesem Fall ist die Überprüfung der CPU-Quoten-Einstellung zweckmäßig.
 - Einsatz von REC / SRDF (siehe [Seite 50](#)).
Bei CKD-Formatierung liegt der Anteil REMAINING SERVICE TIME in der gleichen Größenordnung wie DEVICE CONNECT TIME. Ist der Anteil REMAINING SERVICE TIME deutlich höher als DEVICE CONNECT TIME, so sind die „Remote Data Links“ nicht leistungsfähig genug, d.h. ihre Anzahl muss erhöht werden.
Bei FBA-Formatierung ist die DEVICE CONNECT TIME höher als es der zu übertragenden Datenmenge entspricht (dies ist eine Eigenschaft des FBA-Protokolls: die Diskonnektierung erfolgt später als bei CKD-Formatierung). Ein Anteil REMAINING SERVICE TIME der gleich groß ist wie DEVICE CONNECT TIME weist auf eine zu geringe Anzahl „Remote Data Links“ hin.

10.3.2.2 Paging-Aktivitäten

Das Charakteristikum der Hauptspeicher-Verwaltung ist der Paging-Mechanismus, der die effiziente Nutzung eines – im Verhältnis zu den virtuellen Adressraumanforderungen relativ kleinen – Hauptspeichers erlaubt.

Da Paging sowohl zu zeitlichen Verzögerungen führt als auch Last auf dem Prozessor und der Plattenperipherie erzeugt, sollten die im [Abschnitt „Richtwerte für die BS2000/OSD-Server“ auf Seite 331](#) angeführten Paging-Raten nicht überschritten werden.

Weiter ist zu beachten, dass eine ausreichende Anzahl Volumes zur Durchführung der Paging-Ein-/Ausgaben verfügbar ist. Pro Volume können ca. 30 Paging-Ein-/Ausgaben pro Sekunde durchgesetzt werden (bei 10% Auslastung und einer Read-Hit-Rate von 70%).

Hinweise über die Ursache der Paging-Rate ergeben sich aus dem Verhältnis des vom System genutzten Speichers (SWS) zur Anzahl der für Paging verfügbaren Hauptspeicher-Seiten (NPP).

Siehe auch die Hinweise zu „[Report „Page frames“ der Reportgruppe MEMORY und „Main memory utilization“ der Reportgruppe WORKING-SET“ auf Seite 280](#).



Die Tabellen im [Abschnitt „Richtwerte für die BS2000/OSD-Server“ auf Seite 331](#) enthalten typische Hauptspeicher-Ausbauten für verschiedene Server. Der tatsächlich benötigte Hauptspeicher-Bedarf hängt von der gefahrenen Last ab. TP- und Dialog-Lasten haben üblicherweise einen deutlich größeren Hauptspeicher-Bedarf als Batch-Lasten.

10.3.2.3 CPU-Auslastung

Das Betriebsmittel CPU wird i.d.R. von vielen Tasks simultan benutzt. Auch hier ist die Auslastung ein wesentliches Kriterium für die Wartezeit einer Task vor dem Betriebsmittel CPU.

Die durchschnittliche Hardware-Bedienzeit des Betriebsmittels CPU pro Anwenderanforderung (z.B. zwischen zwei DVS-Ein-/Ausgaben) ist vergleichbar klein gegenüber der Durchführung der Ein-/Ausgabe.

Die Wartezeit vor dem Betriebsmittel CPU hat daher auf die Antwort- bzw. Verweilzeit wesentlich geringeren Einfluss als die Wartezeit vor Platten (Volumes). Im Vergleich zu gemeinsam benutzten Volumes ist deshalb eine höhere Auslastung vertretbar, doch sollte für die **Hauptanwendung** eine Auslastung von **70%** nicht überschritten werden (bei Multi-processor-Servern ist - je nach Anzahl CPUs - eine max. Auslastung bis 90% tolerierbar). Wenn mehrere Gastsysteme genutzt werden, so ist i.d.R. eine um 5 - 15% höhere Gesamtauslastung als im Native-Betrieb möglich.

Zusätzlich erlaubt die Systemsteuerung (PRIOR) die Bevorzugung von Tasks mit Hilfe der Prioritätsvergabe. Damit ist es möglich, auch einen Teil der restlichen **30%** (bzw. 10%) durch **niederpriorie Tasks** (mit entsprechend langen Wartezeiten vor dem Betriebsmittel CPU) zu nutzen. Eine CPU-Auslastung von 100% ohne jegliche Beeinträchtigung der Hauptanwendung wird nur in Ausnahmefällen möglich sein.

ANALYZER-Reports zur Beurteilung der CPU-Auslastung

Reportgruppe	Report	Bedeutung
CATEGORY-CPU	CPU utilization (TU+TPR) for category	CPU-Auslastung in den Funktionszuständen TU und TPR pro Task-Kategorie
CPU	Utilization real	CPU-Auslastung in den Funktionszuständen TU, TPR und SIH (echte Werte). Dieser Report ist bei VM2000-Einsatz wichtig, um den CPU-Anteil des gemessenen Gastsystems am Gesamtsystem zu ermitteln.
	Active logical machines	Anzahl aktiver logischer Maschinen (Anzahl der von BS2000/OSD genutzten CPUs)
	Sum SVC calls	Summe aller SVC-Aufrufe in den Funktionszuständen TU und TPR

Hohe CPU-Auslastung

Von einer hohen CPU-Auslastung wird gesprochen, wenn eine Auslastung > 90% vorliegt.

Eine hohe CPU-Auslastung bedeutet nicht zwangsläufig, dass das Betriebsmittel CPU der Engpass ist.

Wird eine hohe CPU-Auslastung festgestellt (Voraussetzung: die Anzahl der von BS2000/OSD genutzten CPUs stimmt mit der Anzahl der tatsächlich verfügbaren CPUs überein), muss zunächst überprüft werden, ob auch eine effiziente Nutzung des Betriebsmittels CPU vorliegt.

Dazu dient das Verhältnis der Auslastung in den Funktionszuständen TU und TPR zur Auslastung im Funktionszustand SIH

(siehe dazu Anmerkungen zu Report „Utilization normed“, [Seite 277](#)).

Anzustreben ist das Verhältnis:

$TU + TPR > 3 * SIH$ (S-Server) bzw. $TU + TPR > 7 * SIH$ (SQ-Server)

Bei Vorliegen eines ungünstigeren Verhältnisses muss die Untersuchung in Richtung Reduzierung der für den Anwender im Wesentlichen nicht produktiven SIH-Zeit erfolgen:

- Überprüfen der Höhe der DVS-Ein-/Ausgaberate (siehe [Abschnitt „Auslastung der Ein-/Ausgabe-Peripherie“ auf Seite 292](#)).
- Überprüfen der Höhe der Paging-Rate (siehe [Abschnitt „Paging-Aktivitäten“ auf Seite 299](#)).
- Aufrufe an BS2000/OSD werden grundsätzlich über SVCs abgewickelt. Die Bearbeitung der SVC-Unterbrechung bedeutet einen entsprechenden Aufwand im System, welcher der aufrufenden Task verrechnet wird.



Werden die in den Tabellen im [Abschnitt „Richtwerte für die BS2000/OSD-Server“ auf Seite 331](#) dargestellten SVC-Aufrufe pro Sekunde erreicht, so beträgt der Aufwand für die Unterbrechungsanalyse und Abschlussbehandlung der Systemroutine (also nicht der Aufwand für die Systemroutine selbst) ca. 5% der CPU-Zeit.

Die Gesamtanzahl der SVCs kann über den Report „Sum SVC calls“ ermittelt werden. Mit dem Messprogramm TASK des SM2 kann die SVC-Anzahl taskspezifisch gemessen werden. Die weitere Aufschlüsselung in Form einer SVC-Statistik ist durch eine Benutzer-Task-Messung für ausgewählte Tasks mit anschließender Auswertung durch SM2-PA möglich.

Zur Beurteilung der Frage, inwieweit das Betriebsmittel CPU bei hoher CPU-Auslastung **und** effizienter Nutzung den Engpass bildet, dient das Verhältnis:

Wartezeit auf Benutzung der CPU / Hardware-Bedienzeit der CPU

Richtwert

Das Betriebsmittel CPU kann als überlastet angesehen werden, wenn sich für die Hauptanwendung folgendes Verhältnis bzw. für die Hintergrundanwendung ein entsprechend größeres Verhältnis ergibt.

Wartezeit auf Benutzung der CPU / Hardware-Bedienzeit der CPU > 3

Treten für die Hauptanwendung größere Wartezeiten bezüglich Benutzung des Betriebsmittels CPU auf, so können zwei Gründe vorliegen:

- Die Einstellung der Systemsteuerparameter (insbesondere die Prioritätsvergabe) entspricht nicht den Erfordernissen.
- Der CPU-Zeit-Gesamtbedarf aller Tasks ist zu hoch (kritischer Fall).

Nähere Hinweise dazu geben die prozentualen Verweilzeiten in den System-Warteschlangen des SM2-Messprogramms SYSTEM (Auswertung mit SM2R1-Anweisung //PRINT-QUEUE-TRANSITION).

Beispiel 1

CPU-Auslastung: 96 %

TU-Anteil: 36 %

TPR-Anteil: 45 %

SIH-Anteil: 15 %

Hauptanwendung: Task-Kategorie TP

Verweilzeit in O1 (%) / Verweilzeit CPU (%) = 5,4% / 1,2% = 4,5

Hintergrundanwendung: Task-Kategorie Batch

Verweilzeit in O1 (%) / Verweilzeit CPU (%) = 2,1% / 1,4% = 1,5

Durch ungünstige Einstellung der Systemsteuerparameter läuft die geplante Hintergrundanwendung Batch nicht im Hintergrund, wodurch die Tasks der Hauptanwendung TP zu lange auf Benutzung der CPU warten müssen.

Abhilfe: Vergabe einer höheren Priorität für die Hauptanwendung TP oder die Nutzung von PCS (siehe [Abschnitt „Einführung in das PCS-Konzept“](#) auf Seite 222).



Aus den prozentualen Verweilzeiten in den System-Warteschlangen kann **nicht** auf die CPU-Auslastung geschlossen werden.

(Die Summe der Verweilzeiten in allen System-Warteschlangen ergibt 100%.)

Beispiel 2

CPU-Auslastung: 96 %

TU-Anteil: 36 %

TPR-Anteil: 45 %

SIH-Anteil: 15 %

Hauptanwendung: Task-Kategorie TP
$$\text{Verweilzeit in O1 (\%)} / \text{Verweilzeit CPU (\%)} = 5,4\% / 1,2\% = 4,5$$
Hintergrundanwendung: Task-Kategorie Batch
$$\text{Verweilzeit in O1 (\%)} / \text{Verweilzeit CPU (\%)} = 16,8\% / 1,4\% = 12$$

Der CPU-Zeit-Gesamtbedarf aller Tasks ist zu hoch. Das Betriebsmittel CPU ist überlastet.

Werden die Leistungsanforderungen der einzelnen Anwender bei hoher CPU-Auslastung trotz effizienter Nutzung und kurzer Wartezeiten auf die Benutzung des Betriebsmittels CPU nicht erfüllt, so ist der Betriebsmittelbedarf pro Anwenderanforderung (z.B. CPU-Zeitbedarf pro Transaktion) zu hoch (siehe [Abschnitt „Untersuchen eines anwenderorientierten Leistungsproblems“ auf Seite 306](#)).

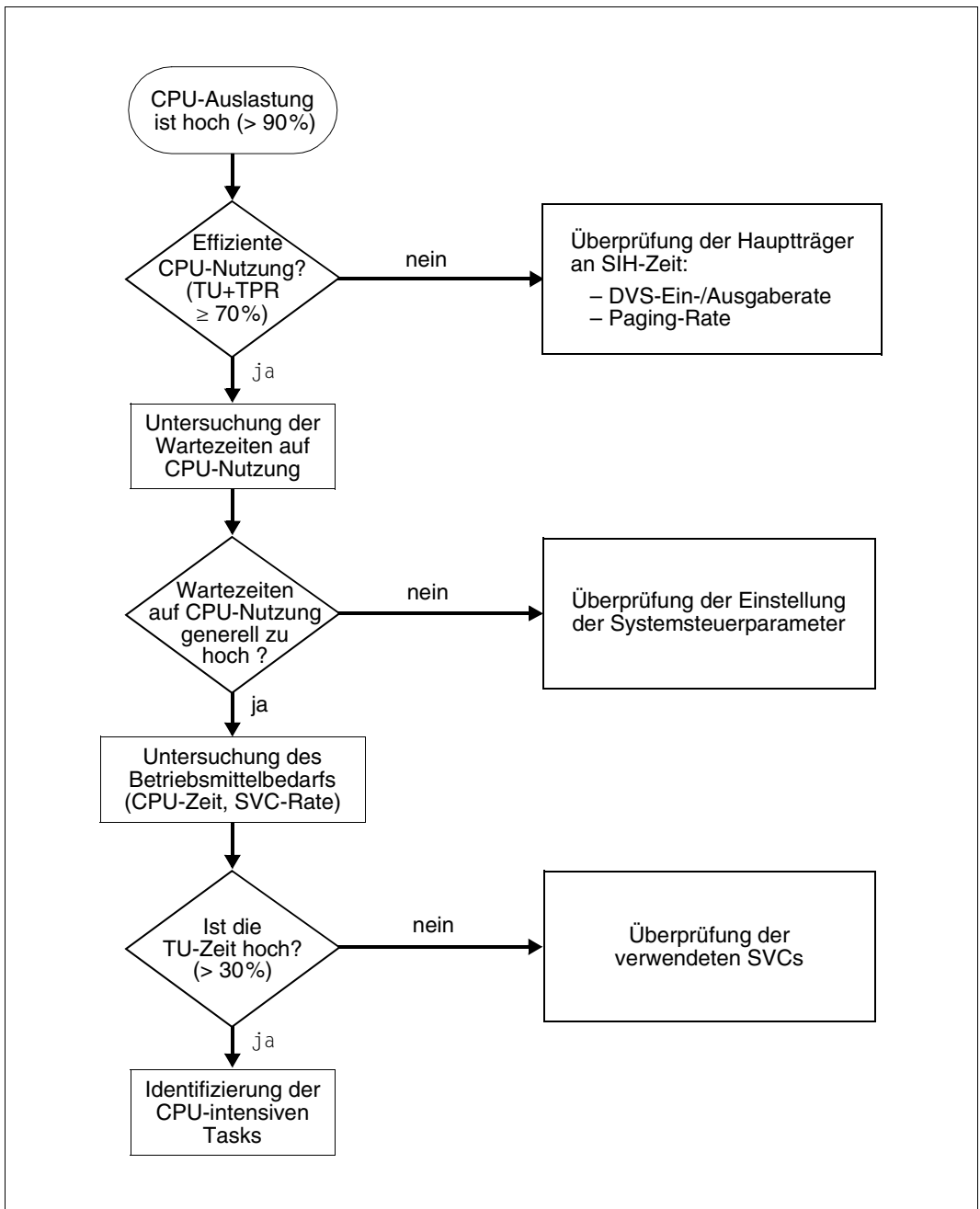


Bild 19: Tuning-Ansätze bei hoher CPU-Auslastung

Niedrige CPU-Auslastung

Treten Leistungsprobleme auf, ohne dass das Betriebsmittel CPU entsprechend ausgelastet ist (< 90%), so können folgende Gründe vorliegen:

- Konfliktsituation auf der DVS-Ein-/Ausgabe-Seite
(siehe [Abschnitt „Auslastung der Ein-/Ausgabe-Peripherie“ auf Seite 292](#))

Besonders unangenehm wirkt sich die Überlastung von Volumes aus, auf denen sich Dateien befinden, die von vielen Anwendern gleichermaßen benötigt werden (z.B. bestimmte Anwender-Steuerdateien bzw. die Systemdateien TSOSCAT oder SYSEAM).

Mit dem SM2-Messprogramm FILE kann die Anzahl der Ein-/Ausgaben auf diese Dateien ermittelt werden.

- Zu hohe Paging-Rate (siehe [Abschnitt „Paging-Aktivitäten“ auf Seite 299](#))

Bei gravierendem Hauptspeicher-Mangel (dadurch gekennzeichnet, dass die angegebenen Richtwerte für die Paging-Rate weit überschritten werden) ergeben sich Engpässe auf den Public Volumes bezüglich durchzuführender Paging-Ein-/Ausgaben pro Sekunde, die zu einem erzwungenen CPU-Wartezustand (IDLE) führen.

- Verzögerungen durch PASS/VPASS-Aufrufe

Bei allzu häufigen Aufrufen der SVCs PASS/VPASS werden Tasks zu oft in den Wartezustand versetzt und können die CPU nicht belegen.

- Engpässe durch Server-Tasks

Es gibt Anwendungen, die große Teile der Arbeit von spezialisierten Server-Tasks abwickeln lassen. Wird das Fortkommen dieser Tasks durch Paging, Lock-Situationen, Betriebsmittelüberlastung, ungünstige Prioritäten o.ä. behindert, pflanzt sich diese Behinderung auf die gesamte Anwendung fort. Die Server-Task wirkt selbst wie ein Betriebsmittel.

- Ungenügende Parallelverarbeitung durch eine zu geringe Task-Anzahl

Bei Leistungsbetrachtungen spielen nicht nur die Bedienzeiten der jeweiligen Hardware-Betriebsmittel eine Rolle, sondern auch die Zeit, in der eine Task zur Erfüllung einer gewünschten Anwenderfunktion durch CPU- und Ein-/Ausgabe-Tätigkeit belegt ist.

Beispiel

Beträgt die durchschnittliche Verarbeitungszeit pro Transaktion 0,5 s (0,2 s CPU-Zeit + 0,3 s Ein-/Ausgabezeit), so kann mit einer Task folgende maximale Transaktionsrate (Transaktionen/s) erzielt werden (die Wartezeit auf die entsprechenden Hardware-Betriebsmittel ist nicht berücksichtigt):

$$1 / 0,5 \text{ sek. pro Transaktion} = 2 \text{ Transaktionen pro Sekunde}$$

Dabei ergibt sich eine CPU-Auslastung von $0,2 * 2 = 0,4$, d.h. 40%.

Eine höhere Transaktionsrate bzw. eine bessere Auslastung des Betriebsmittels CPU lässt sich nur durch eine Erhöhung der Task-Anzahl erreichen (Details siehe [Abschnitt „Task-Belegungszeit“ auf Seite 307](#)).

10.3.3 Untersuchen eines anwenderorientierten Leistungsproblems

Das Auffinden der Ursachen von anwenderorientierten Leistungsproblemen ist im Vergleich zur Untersuchung von systemorientierten Leistungsproblemen mit SM2 wesentlich schwieriger.

Bei unbefriedigenden Antwort- bzw. Verweilzeiten bestimmter Anwender ist neben der Beurteilung der globalen Auslastungswerte die Interpretation der taskspezifischen Messwerte erforderlich.

Zusätzlich wird die Durchführung einer SM2-Benutzer-Task-Messung empfohlen:

Über /START-TASK-MEASUREMENT wird dem Anwender bzw. der Systembetreuung die Möglichkeit geboten, seine Tasks zur Überwachung anzumelden.

Neben den taskspezifischen Messwerten kann eine PC-Statistik und eine SVC-Statistik der eigenen Task angefordert werden. Die Auswertung der in eine anwenderspezifische Datei geschriebenen Messwerte erfolgt durch das Softwareprodukt SM2-PA (siehe Handbuch „SM2-PA“ [29]).

10.3.3.1 Task-Belegungszeit

Die Task-Belegungszeit besteht aus der CPU-Zeit und der Ein-/Ausgabezeit einer gewünschten Anwenderfunktion sowie der Wartezeit auf die entsprechenden Hardware-Betriebsmittel aufgrund deren Auslastung.

Je größer die Task-Belegungszeit ist, desto geringer ist die Anzahl der Anwenderanforderungen, die durch diese Task pro Zeiteinheit abgefertigt werden können.

Beispiel

Task-Belegungszeit für eine durchschnittliche Transaktion im TP-Betrieb:

Hardware-Bedienzeit der CPU	200 ms
Warten auf CPU (bei 60% CPU-Auslastung, Voraussetzung „M/M/1“)	300 ms
Hardware-Bedienzeit aller Ein-/Ausgaben	400 ms
Warten auf Ein-/Ausgabe (bei 30% Auslastung, Voraussetzung „M/M/1“)	170 ms
	1070 ms

Wird für diese Anwendung während der Zeit der Spitzenbelastung eine größere Transaktionsrate als $1 / 1,07 \text{ Sek. pro Transaktion} = 0,93 \text{ Transaktionen pro Sek.}$ gefordert und steht nur **eine** Task zur Verfügung, so wird diese Task zum Engpass.

Die entstehenden Wartezeiten vor der Task (die eintreffenden Nachrichten stauen sich im BCAM-Puffer) können ein Vielfaches der Task-Belegungszeit erreichen (siehe auch Report „Inwait time distribution“ der Reportgruppe BCAM-CONNECTION, [Seite 276](#)).

Hinweise auf die Task-Belegungszeit liefert das SM2-Messprogramm TASK.

Aus den Duration-Angaben für die verschiedenen Wartezustände und die CPU-Zeitaufnahme kann man die Task-Belegungszeit abschätzen. Schwierigkeiten entstehen, wenn man nicht mit Sicherheit sagen kann, ob bestimmte Wartezustände freiwillig oder unfreiwillig sind, d.h. ob ein Task den Service, der für ihn erbracht wird, selbst angefordert hat oder nicht.

Das folgende Verhältnis weist auf eine zu lange Task-Belegungszeit hin ($D = \text{Duration}$):

$$\frac{(\text{CPU-Zeit} + \text{CPU-WAIT}(D) + \text{DISK-IO-WAITS}(D) + \text{NON-DISK-IO-WAITS}(D))}{\text{Verweilzeit}} > 0,7$$

In der Verweilzeit ist die Wartezeit auf andere Tasks **nicht** enthalten (diese ist also unter Umständen noch dazu zu rechnen).

Die ausgewiesene CPU-Zeit bezieht sich auf die Funktionszustände TU und TPR. Die Zeiten für die Ein-Ausgabe betreffen die Software-Bedienzeit.

10.3.3.2 Behinderungen durch andere Tasks

Konfliktsituationen beim Zugriff auf gemeinsame Hardware-Betriebsmittel (i.W. Public/Private Volumes)

Deutliches Anzeichen für die Behinderung durch andere Tasks ist ein SOFTWARE DURATION-Wert (Volume belegt), der mehr als 10% größer ist als der entsprechende Wert HARDWARE DURATION (Ausnahme: asynchrone Ein-/Ausgabe, siehe Erläuterung zu „Report „Time“ der Reportgruppe DISK“ auf Seite 278).

In diesem Fall muss eine Task vor der Einleitung der Ein-/Ausgabe auf das Volume warten, weil dieses tätig ist. Welche anderen Tasks dieses Volume ebenfalls mit Ein-/Ausgaben belegen, kann dem SM2-Messprogramm TASK (mit INFO=HIGH) entnommen werden.

Zur Lösung des Problems ist der Einsatz der Funktion PAV oder eine Verlagerung von Dateien notwendig.

Konfliktsituationen bei Veränderung (Update) gemeinsamer Daten

Konfliktsituationen können beim Update gemeinsamer Daten (z.B. Tabellen) entstehen, deren Inhalt für den Betriebsablauf in einem konsistenten Zustand erhalten werden muss.

Konflikte dieser Art können als Zwangs-Serialisierung bezeichnet werden und sind in den taskspezifischen Kenndaten gekennzeichnet durch einen relativ hohen Anteil von:

- „active waits“ bei Serialisierung über den Börsen-Mechanismus bzw. Angabe von Warteaufrufen VPASS 0
- „inactive waits“ bei Verwendung des Warteaufrufs PASS

Es gibt jedoch viele Fälle, in denen sich ein solcher Zustand mit SM2 durch keinerlei Anzeichen vom normalen Zustand unterscheiden lässt. Anwendersysteme (wie z.B. UTM, UDS, SESAM/SQL, ORACLE) bieten heute eigene Monitore, mit denen man solche Konflikte lokalisieren kann.

Probleme bei mehrstufigen Task-Konzepten (Server-Tasks oder Handler-Tasks)

Sehr häufig werden Funktionen, die allen Tasks gemeinsam sind, herausgezogen und in **einer** Task konzentriert (z.B. Durchführung der Ein-/Ausgabe zu speziellen Datenbeständen oder Datenbankaufrufe).

Wird die Task-Belegungszeit für diese zentrale Task zu lang, so treten für die darauf aufsetzenden Tasks Wartezustände auf.

Ein Engpass vor einer UTM-Anwendung infolge einer zu geringen Anzahl von (UTM-)Server-Tasks kann direkt gemessen werden (siehe die Erklärungen im [Abschnitt „Transaktionsbetrieb und BCAM“ auf Seite 188](#)).

Erkennbar sind Wartezustände häufig durch ein Ansteigen der im SM2-Messprogramm TASK ausgewiesenen Anzahl bzw. Zeitdauer der „active waits“ für die aufrufenden Tasks. In der Anzahl „active waits“ ist bei TP-Tasks neben „Warten auf Ein-/Ausgabe“ auch das „Warten auf Terminal-Eingabe“ enthalten. Dieser Anteil ist bei hoher Task-Belegungszeit der zentralen Task vernachlässigbar.

Der Server-Task muss so programmiert sein, dass er keine Sperren benötigt, auf die er gegebenenfalls wartet. Wird er gegenüber anderen Tasks nicht vorrangig behandelt (Priorität), so besteht ebenfalls die Gefahr von plötzlichen Durchsatzeinbrüchen. Durch das Auftragsverfahren zwischen Auftraggeber- und Server-Task kann es im Falle von nicht fehlerfreier Programmierung zu Verklemmungen kommen, die den Server-Task stilllegen. Alle diese Probleme zeigen, dass es sich bei Server-Tasks um ein Coding von besonders hoher Qualität handeln muss.

Beispiel (vereinfacht)

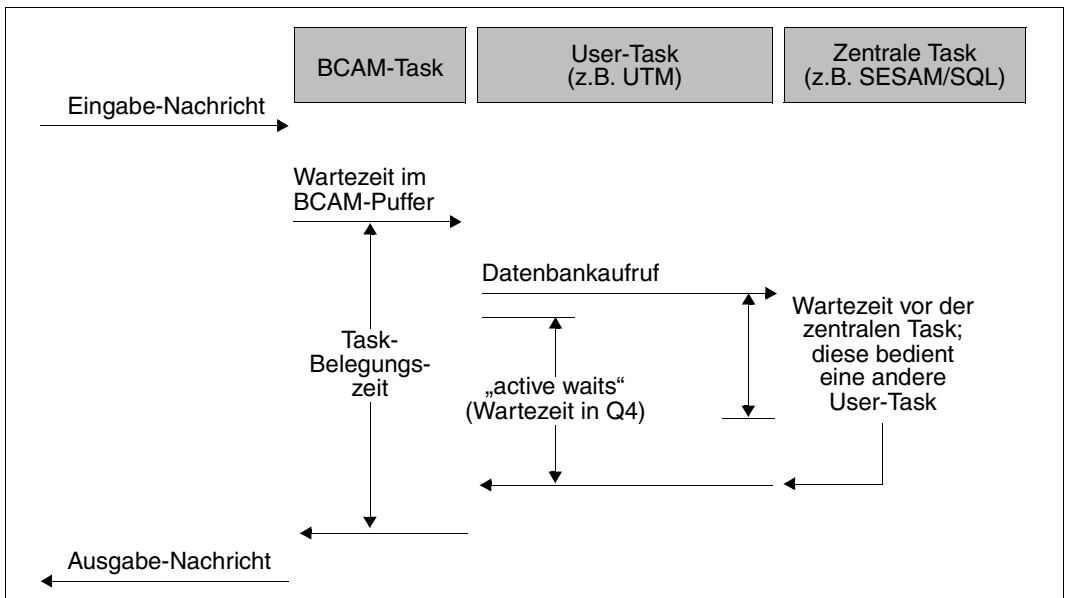


Bild 20: Zeitverhalten mehrstufiger Task-Konzepte

10.3.3.3 Hoher Betriebsmittelbedarf

Werden die Leistungserwartungen der Anwender trotz effizienter Nutzung der Betriebsmittel und kurzer Wartezeiten nicht erfüllt, so deutet dies auf einen zu hohen Betriebsmittelbedarf pro Anwenderanforderung hin (z.B. CPU-Zeitbedarf und/oder Anzahl Ein-/Ausgaben pro Transaktion).

Hinweise auf den Betriebsmittelbedarf ergeben sich aus den durch das SM2-Messprogramm TASK gelieferten Angaben über:

- CPU-Zeitverbrauch
- Anzahl SVC-Aufrufe in TU
- Anzahl DVS-Ein-/Ausgaben

Zur genaueren Untersuchung des Verhaltens von Anwenderprogrammen ist eine SM2-Benutzer-Task-Messung mit einer SM2-PA Auswertung erforderlich:

- Die PC-Statistik bietet die Möglichkeit, diejenigen Programmbereiche zu erkennen, die sehr häufig durchlaufen werden.
- Die SVC-Statistik enthält alle aufgerufenen SVC-Nummern und deren Aufrufadressen.

10.3.4 Einfluss des Netzes

Alle bisherigen Empfehlungen haben sich mit der Verbesserung der Effizienz des zentralen Verarbeitungsrechners befasst. Wie im [Kapitel „Leistungserwartungen aus Anwendersicht“ auf Seite 15](#) ausgeführt, sind gute Transaktionszeiten ein wesentlicher Produktivitätsfaktor. Im [Abschnitt „Transaktionsbetrieb und BCAM“ auf Seite 188](#) werden Maßnahmen zur Antwortzeitoptimierung im Transaktionsbetrieb und bei BCAM beschrieben.

[Bild 1 auf Seite 17](#) zeigt, dass in den Transaktionszeiten die Netzlaufzeiten enthalten sind. Diese Laufzeiten können größer sein als die eigentliche Antwortzeit im Verarbeitungsrechner. Es ist daher in jedem Fall von unbefriedigendem Verhalten zu klären, ob die schlechten Transaktionszeiten nicht durch das Netz verursacht werden.

Weisen die Reports der Reportgruppe BCAM-CONNECTION akzeptable Antwortzeiten aus und werden dennoch an den Terminals überwiegend lange Transaktionszeiten beobachtet, liegt ein Engpass im Netz vor.

Ein Netz ist ein komplexes Gebilde mit vielen Abhängigkeiten. Angesichts der Vielfalt möglicher Netzstrukturen soll nur auf die wesentlichen Merkmale und Randbedingungen hingewiesen werden. Die umfassende Behandlung solcher Probleme sprengt den Rahmen dieses Handbuchs.

Lastanforderungen

Lasttyp	Anforderung
TP-Betrieb oder Dialogbetrieb	kurze Antwortzeit
Datensicherung, File Transfer	Durchsatz
Internet, E-Mail	Durchsatz, Antwortzeit

Mögliche Engpässe

- Leistung des Netzanschlusses (z.B. HNC älteren Typs)
- Client System (z.B. Hardware-Leistung, Struktur der Anwendung, Networking-Parameter)
- Netzkonfiguration mit Zielkonflikten bezüglich Antwortzeit/Durchsatz (z.B. Parallelbetrieb von transaktions- und durchsatzorientierten Anwendungen)
- Netzstruktur (z.B. Konfiguration von Netzknotten, Router, Switches, Hubs)
- Netzüberlast durch Dritte (z.B. Internet)

10.4 Leistungsbedarf von SM2

Für die Erfassung der Messdaten benötigt SM2 die Ressourcen CPU und Plattenspeicher. Der Bedarf hängt ab:

- von der Art der Messprogramme und ihrer Parametrisierung
- von der Hardware-Konfiguration
- von Lastprofil (i.W. von der Anzahl Tasks und der Ein-/Ausgabe-Intensität)

Insgesamt ist der Ressourcen-Verbrauch durch SM2 relativ gering. Gelegentlich sollte überprüft werden, ob eine Einschränkung der Messprogramme zur Ressourcen-Entlastung nötig bzw. möglich ist. Eine Einschränkung der Messprogramme geht allerdings zu Lasten der Güte der Performance-Überwachung.

Messwerte

In einem Laborversuch wurde der Ressourcen-Bedarf von SM2 vermessen unter einer typischen OLTP-Last, die den Server zu ca. 70% auslastet.

Dabei wurde folgender CPU-Bedarf von SM2 in absoluten Prozentwerten gemessen:

- 0,5 - 1 % für die Basismessung (Messprogramme Standard und SYSTEM)
- zusätzlich 1 % für Antwortzeiterfassung und Messprogramm UTM
- zusätzlich 1 % für die Nutzung der Messprogramme PERIODIC-TASK und TASK
- zusätzlich 1 % für die Nutzung des Messprogramms STORAGE-SYSTEM (mit `//SET-STORAGE-SYSTEM-PARAMETERS ADDITIONAL-DATA=*SYMMETRIX(TYPE=(*PHYSICAL-DISK, *DIRECTOR))`), sowie dem Einschalten der Ein-/Ausgabe-Zeitmessung der Plattengeräte (mit `//SET-SAMPLING-DEVICE-PARAMETERS DISK-SERVICETIME=*ON`).

Diese Werte wurden mit dem Standard-Sampling-Intervall 800 ms gemessen. Bei Änderung des Sampling-Intervalls ändert sich der CPU-Bedarf von SM2 linear mit dem Anteil der periodisch erfassten Daten. Z.B. verringert sich bei der Basismessung mit Antwortzeiterfassung und Messprogramm UTM der CPU-Bedarf von SM2 von 1,5% auf knapp unter 1%, wenn das Sampling-Intervall auf 2000 ms geändert wird.

Für das Ablegen der Messdaten wird Plattenspeicherplatz benötigt. Die Angabe von Richtwerten in absoluten Maßzahlen ist nicht möglich, da der Bedarf zu stark von der jeweiligen Hardware- und Software-Last sowie der Konfiguration der Messprogramme abhängt. Es ist jedoch zu beachten, dass insbesondere das Messprogramm PERIODIC-TASK (und noch mehr das nicht mehr empfohlene Messprogramm DISK) die Messwertedatei um 50% oder mehr gegenüber der Basismessung anwachsen lässt.

10.5 Performance-Analyse mit COSMOS

COSMOS ist ein ereignisgesteuerter Monitor für die Erfassung detaillierter Messdaten zur gezielten Performance-Diagnose von BS2000/OSD-Systemen. Es ist ein Messprogramm von SM2 und kann somit über die SM2-Kommandos gesteuert werden.

COSMOS zeichnet den gesamten zeitlichen Ablauf von Systemereignissen auf. Es können ca. 90 verschiedene Arten von Ereignissen registriert werden, z.B. Start und Ende einer Ein-/Ausgabe, Systemaufrufe (SVCs), Erzeugung und Beendigung von Tasks. Jedes Ereignis („Event“) wird durch einen 4 Zeichen langen Namen identifiziert. Zur Erfassung der Ereignisse sind an ausgewählten Punkten im System COSMOS-Schnittstellen, so genannte „Hooks“, implementiert. Abhängig von den gewählten Parametern ist der Hook „geöffnet“, d.h. der Datensammelcode wird bei jedem Eintreten des entsprechenden Ereignisses aktiviert und ein Datensatz geschrieben. Beim Beenden von COSMOS werden alle zu diesem Zeitpunkt geöffneten Hooks geschlossen.

Die Daten können für alle Tasks oder für nach bestimmten Kriterien (Benutzerkennung, Kategorie, Jobname oder TSN) ausgewählte Tasks gesammelt werden.

Die erfassten Messdaten werden in Messwertedateien (auf Band oder Platte) geschrieben. Zur Auswertung der COSMOS-Messwertedateien sind spezielle Auswerteprogramme verfügbar. In der Regel sind Experten-Kenntnisse zur Aus- und Bewertung von COMOS-Messungen nötig.

11 Optimierungshinweise zur Erstellung von Anwender-Software

Dieses Kapitel gibt Empfehlungen für den optimalen Umgang mit Betriebsmitteln im Rahmen der Dateiverarbeitung sowie Hinweise für das performante Laden von Programmen/Produkten.

11.1 Wahl der Zugriffsmethode

UPAM und BTAM

UPAM und BTAM sind die Basis-Zugriffsmethoden für Platte bzw. Magnetband. Da sie für kleine Anwendungen kaum eine Rolle spielen, haben sie fälschlicherweise an Bedeutung verloren. Bei der Planung eines Verfahrens sollten sie in jedem Fall in Betracht gezogen werden. Sie bieten einige in anderen Zugriffsmethoden nicht verfügbare Funktionen. Häufig ist ihre Anwendung gegenüber anderen Zugriffsmethoden ohne einen Verlust an Anwenderfreundlichkeit bei deutlicher Leistungssteigerung möglich. UPAM z.B. bietet wahlfreie Zugriffe über die „Half Page Number“.

Gemeinsame Vorteile von UPAM und BTAM sind:

- einfache Schnittstelle für das Anwenderprogramm
Es gibt je nur einen Aktionsaufruf an das DVS. Die gewünschte Aktion wird durch aktuelle Parameter formuliert. Jede Schreib-Lese-Anforderung aus dem Programm bewirkt eine physikalische Ein-/Ausgabe.
Im Vergleich zu SAM/ISAM: Das datensatzbezogene Blocken/Entblocken entfällt, deshalb arbeiten UPAM und BTAM mit entsprechend kürzeren Pfadlängen.
- Wahlmöglichkeit zwischen synchronen und asynchronen Ein-/Ausgaben
Asynchrone Ein-/Ausgaben können die Laufzeit eines Programms erheblich abkürzen.
- vollständiger Durchgriff auf die Eigenschaften des Speichermediums, das optimal genutzt werden kann.
- Auftragskettung zur Einsparung von SVCs und gekettete Ein-/Ausgaben zur Einsparung von Systemaufwand (Chained-IO)
- Verarbeitung von Dateien, die mit anderen Zugriffsmethoden erstellt wurden

Zusätzliche Vorteile von UPAM sind:

- schlüsselähnlicher, wahlfreier Zugriff über „Half Page Number“
- Verarbeitung von inhaltlich geschädigten Dateien möglich
- Mehrfachbenutzbarkeit auch bei Aktualisierungen
- Möglichkeit zur Kopplung asynchroner Ein-/Ausgabe-Aufrufe mit Eventing

Zusätzliche Vorteile von BTAM sind:

- Blocklänge innerhalb der von der Hardware gegebenen Minimal- und Maximallänge beliebig wählbar
- Verarbeitung von Datenträgern aus fremden Betriebssystemen möglich

FASTPAM

Für die Zugriffsmethode FASTPAM gilt i.W. alles, was zu UPAM gesagt wird.

Zu beachten sind folgende Einschränkungen:

- Das Dateiformat muss vom Typ NK4 sein (BLKCTRL=NO, Blockgröße=ein Vielfaches von 4 KB).
- Zugriffe stets über absolute Blocknummer
- keine Synchronisierungsmechanismen beim Zugriff mehrerer Anwender auf eine Datei
- FASTPAM unterstützt den Multi-System-Verbund mittels Shared-Pubset, aber nicht Shared-Private-Disk bzw. Remote-File-Access.

Der Performance-Gewinn gegenüber UPAM ergibt sich dadurch, dass die für Ein-/Ausgaben erforderlichen Initialisierungs- und Validierungsroutinen nicht bei jeder Ein-/Ausgabe durchlaufen werden, sondern nur einmal vor der Dateieröffnung.

Der Anwender definiert vor dem OPEN eine Systemumgebung, bestehend aus ENVIRONMENT und IO-AREA-POOL. Dies sind System- und Anwenderspeicherbereiche, die bei den Dateizugriffen immer wieder benutzt werden.

Der maximale Performance-Gewinn ergibt sich, wenn die Systembereiche resident angelegt werden. Voraussetzung ist die Berechtigung im Benutzerkatalog (DMS-TUNING-RESOURCES=*EXCLUSIVE-USE) sowie eine entsprechende Versorgung des Operanden RESIDENT-PAGES im Benutzerkatalog und beim Starten der Task.

SAM

Die Zugriffsmethode SAM ist auf die sequenzielle Verarbeitung von Datensätzen abgestimmt. Sie sollte immer dann gewählt werden, wenn Dateien nicht mehrfachbenutzbar sein sollen und kein wahlfreier Zugriff gefordert ist. Die einfache und doch flexible Satzstruktur gestattet sowohl Sätze fester Länge als auch solche von variabler Länge. Beide Längentypen können je nach Anwendung zur Einsparung von Speicherkapazität beitragen. Der Verzicht auf Schlüsselzugriffe führt zu einer gegenüber ISAM wesentlich kürzeren Pfadlänge. Die Datei kann im Bedarfsfall einfach (z.B. mit einem Editor) in eine Schlüsseldatei überführt werden.

Vorteile von SAM sind:

- satzweises Lesen und Schreiben
- eingeschränkte Aktualisierungsmöglichkeiten mit hoher Performance (PUTX)
- kurze Programmaufzeiten durch Vorauslesen des nächsten Blockes
- Datei kann auch mit PAM bearbeitet werden
- hohe Leistung durch Ausschluss der Mehrfachbenutzbarkeit bei Aktualisierungen

ISAM

ISAM ist die höchstentwickelte Zugriffsmethode im BS2000. Sie bietet sehr viele Annehmlichkeiten, die über Speicheraufwand und Pfadlängen zu bezahlen sind. In der Praxis kommt es oft vor, dass Dateien aus Gründen der Kompatibilität oder der Zukunftssicherheit als ISAM-Dateien geführt werden, die aber nach ihrer Art und Anwendung typische SAM-Dateien sind (z.B. Quellprogramme). Wegen der leichten Konvertierbarkeit von SAM zu ISAM ist dies nicht erforderlich.

ISAM ist jedoch keine reine Schlüsselzugriffsmethode. ISAM ist optimiert auf Zugriffsfolgen, bei denen einem Schlüsselbegriff eine Reihe von sequenziellen Zugriffen folgt (index-sequenziell). Die Verteilung der Schlüsselhäufigkeit und besonders deren Änderung während der Lebensdauer einer Datei hat Einfluss darauf, wieviele physikalische Ein-/Ausgaben benötigt werden, um einen bestimmten Satz zu lesen. ISAM sollte nur eingesetzt werden, wenn eine oder mehrere der nachfolgenden Eigenschaften benötigt werden:

Vorteile von ISAM sind:

- Schlüsselzugriffe, die weitgehend unabhängig sind von der Schlüsselverteilung und ihrer Änderung
- effektives sequenzielles Lesen nach Schlüsselzugriff
- Mehrfachbenutzbarkeit auch bei Aktualisierungen
- ertragbare Verschlechterung der Zugriffsleistung auch nach extrem umfangreichen einseitigen Erweiterungen

NK-ISAM (Nonkey-ISAM) unterstützt sowohl Platten mit CKD-Format (Count Key Data) als auch mit FBA-Format (Fixed Block Architecture).

Durch die Einführung von ISAM-Pools werden die Index-Bereiche (teilweise auch Datenbereiche) im virtuellen Speicher gehalten. Dies ermöglicht bei index-sequenziellen Zugriffen eine starke Verringerung der physikalischen Ein-/Ausgaben und führt zu deutlichen Laufzeitverkürzungen.

BS2000/OSD automatisiert und optimiert das Anlegen von Pufferbereichen (NK-ISAM-Pools) für NK-ISAM-Dateien, die mit `SHARUPD=YES` geöffnet werden. NK-ISAM-Pools müssen dann nicht mehr konfiguriert werden.

NK-ISAM erlaubt den Einsatz von Sekundärschlüsseln. Werden neben dem Primärschlüssel zusätzliche Ordnungsbegriffe in Form von Sekundärschlüsseln pro logischem Satz geführt, so lässt sich ein gegebener Datenbestand nach verschiedenen Kriterien durchsuchen und verarbeiten.

Im Vergleich zu einer Anwendung ohne Sekundärschlüssel ändert sich der Betriebsmittelbedarf geringfügig, solange der so genannte Indexbaum nicht geändert wird, steigt aber stark an bei Änderungen des Indexbaumes (z.B. bei GET-ELIM, GETKY-ELIM, STORE).

PAM

Neben ISAM sind weitere Schlüsselzugriffsmethoden denkbar, die speziellere Anforderungen besser erfüllen können. BS2000/OSD bietet durch die Zugriffsmethode PAM die Möglichkeit, sich eigene Zugriffsmethoden zu implementieren. Dass dies mit geringem Aufwand möglich ist, zeigt das folgende Beispiel einer HASH-Zugriffsmethode.

Folgende Eigenschaften seien gefordert bzw. gegeben:

- 90% der Zugriffe sollen mit einer Ein-/Ausgabe möglich sein.
- Bestimmte hochfrequentierte Blöcke sollen auch ohne Einsatz von DAB ohne Ein-/Ausgabe gelesen werden können.
- Die Änderungsfrequenz ist hoch, aber es werden nur sehr selten Sätze gelöscht, erzeugt oder stark verlängert.
- Die Schlüsselverteilung ist bekannt und folgt einfach formulierbaren Gesetzen.
- Die Datei wird **nicht** sequenziell gelesen.
- Es steht genügend Speicherkapazität zur Verfügung.

Unter diesen Bedingungen ist durch einen einfachen Algorithmus aus dem Schlüssel die Nummer des Blockes zu ermitteln, in welchem sich der Satz befindet. Der Füllgrad der Blöcke ist so gering, dass die überwiegende Mehrzahl der Aktualisierungen den Block nicht zum Überlauf bringt. Im seltenen Fall eines Blocküberlaufs enthält der Block einen Verweis

auf den Folgeblock und dieser bei Überlauf auf einen weiteren Folgeblock. Die Pufferung von hochfrequentierten Blöcken macht wegen der Eigenimplementierung keinerlei Schwierigkeiten.

Obwohl für jede Schlüsselverteilung ein eigener Umsetzungsalgorithmus Schlüssel zu Blocknummer implementiert werden muss, kann dieses Verfahren gegenüber dem universellen ISAM bei großen Dateien erhebliche Leistungsgewinne bringen.

Datenbanken

Eine wesentliche Erweiterung der Zugriffsmethoden stellen die Datenbanken dar. Bei Datenbanken sind die Aufwände für Installation und Wartung der Datenbestände sowie die Pfadlängen beim Zugriff größer als bei anderen Zugriffsmethoden.

Dafür bieten Datenbanken folgende weitergehende Eigenschaften:

- Die Organisation der Daten und der Zugriffspfade können vom Anwender definiert und seinen Datenbeständen angepasst werden.
- Nicht beendete Transaktionen können zurückgesetzt werden.
- Durch Logging ist der Datenbestand immer bis auf den Zustand kurz vor dem Zusammenbruch rekonstruierbar.
- Zugriffsrechte können definiert und überwacht werden.
- Durch die Möglichkeit, komplexe Beziehungen zwischen den Datensätzen herzustellen, kann die Datenredundanz verringert werden.
- Durch Suchfragen nach mehreren Schlüsseln mit der Möglichkeit zu logischen Verknüpfungen können Anwenderprogrammeile eingespart werden.



Das Softwareprodukt LEASY ist eine Erweiterung der BS2000-Zugriffsmethoden für den TP-Betrieb.

Aufbauend auf ISAM ermöglicht LEASY die Rücksetzbarkeit von Transaktionen und die volle Rekonstruierbarkeit der Datenbestände. Zusätzlich wird ein Zugriff nach Sekundärschlüsseln sowie eine CALL-Schnittstelle angeboten.

Der Einsatz von LEASY ist immer dann zu erwägen, wenn über den TP-Betrieb hinaus keine Datenbankeigenschaften benötigt werden.

DIV

DIV (Data in Virtual) ist eine Zugriffsmethode, die sich von den traditionellen Zugriffsmethoden wie ISAM, SAM oder UPAM dadurch unterscheidet, dass sie ohne Strukturierung der Datei in Sätze oder Blöcke, ohne Ein-/Ausgabe-Puffer und ohne spezielle Ein-/Ausgabe-Makros (wie GET, PUT) auskommt.

DIV bearbeitet nur PAM-Dateien vom Typ NK4 (BLKCTRL=NO, die Blockgröße ist ein Vielfaches von 4 KB).

DIV betrachtet die Datei als eine lineare Byte-Folge. Mit Hilfe der DIV-Funktion MAP kann eine Datei oder ein Dateibereich einem Bereich im virtuellen Adressraum zugeordnet werden. Der einem Dateibereich zugeordnete virtuelle Adressraum bildet dann ein „Fenster“, in dem die Seiten der Datei automatisch erscheinen, wenn auf die entsprechenden Seiten im virtuellen Adressraum mit den üblichen CPU-Befehlen zugegriffen wird. In einem Fenster geänderte Daten können mit der DIV-Funktion SAVE in die Datei geschrieben werden.

Im eingeschwungenen Zustand liegt die Datei bzw. der Dateiteilbereich im Adressraum des Anwenders und ist somit dem Paging unterworfen. Bei großen Dateibereichen und nicht ausreichendem Hauptspeicher steigt die Paging-Rate entsprechend an.

Ein Performance-Gewinn ergibt sich, wenn wiederholt auf Daten im Fenster zugegriffen wird, die durch vorangegangene Zugriffe in das Fenster (über Paging) eingelesen wurden.

DIV-Lesen

Der maximale Performance-Gewinn tritt beim Zugriff auf Daten im Fenster ein, der Platzzugriff entfällt vollständig.

Muss der Datenzugriff über einen Page-Transfer abgewickelt werden, so ist die Pfadlänge pro Zugriff im Vergleich zum Lesezugriff über UPAM um ca. 20% kürzer.

DIV-Schreiben

Sicheres Schreiben wird realisiert, indem die modifizierte Seite nach jedem Schreibzugriff in die Datei zurückgeschrieben wird (DIV-SAVE). In diesem Fall ist die Pfadlänge pro Zugriff um ca. 35% länger als bei einem UPAM-Schreibzugriff. Der Mehraufwand kann nur durch eine Verringerung der Anzahl SAVE-IOs reduziert werden (z.B. einmalige Sicherung beim CLOSE der Datei). Ist beim Schreibzugriff ein Page-Transfer erforderlich, so ist die Pfadlänge um ca. 10% länger als beim UPAM-Schreibzugriff.

Die Zugriffsmethode hat großen Einfluss auf die Leistung des Gesamtsystems und sollte entsprechend sorgfältig ausgewählt werden.

11.2 Wahl des Bearbeitungsmodus

Im engen Zusammenhang mit den Zugriffsmethoden stehen die Bearbeitungsmodi einer Datei. Diese können zwar bei jeder Bearbeitung neu festgelegt werden, sind jedoch fast immer in das bearbeitende Programm eingebettet und daher festgeschrieben. Sie haben einen ähnlich großen Einfluss auf die Leistung des Systems wie die Zugriffsmethoden und sollten daher mit Sorgfalt ausgewählt werden. In allen drei Phasen der Dateibearbeitung (OPEN, Zugriff, CLOSE) werden explizit oder implizit Moduswahl-Entscheidungen getroffen.

11.2.1 OPEN

Durch den Typ des OPEN-Aufrufs wird bei der Programmerstellung festgelegt, welche Zugriffsmöglichkeiten anderen Anwendern der Datei offenstehen. Es sollte daher eine Selbstverständlichkeit sein, keine über die beabsichtigte Nutzung hinausgehenden Modi zu wählen. Wer nur lesen will und nicht mit INPUT eröffnet, sperrt andere lesende Anwender aus. Bei Dateien von allgemeinem Interesse lohnt sich der Aufwand der Fallunterscheidung „Lesen – Aktualisieren“ schon beim OPEN.

Das Eröffnen einer Datei ist ein komplexer Vorgang. Wegen des damit verbundenen Aufwands sollte er so selten wie möglich ausgeführt werden. Dem sind auch Forderungen nach einer möglichst kurzen Sperrdauer unterzuordnen. Gegebenenfalls muss auf die Zugriffsmethoden PAM oder ISAM mit dem Modus SHARUPD ausgewichen werden.

Kann ein gewisser Verlust an Aktualität hingenommen werden, lohnt sich die Installation einer Datei-Kopie für lesende Zugriffe. Diese sollte schreibgeschützt werden, um sie gegen Anwender abzusichern, die auch die Kopie schreibend eröffnen wollen.

Diese Lösung ist leistungsfähiger, weil der Modus SHARUPD den Aufwand für den Zugriff ansteigen lässt. Ist die Wahrscheinlichkeit einer Kollision klein, lohnen diese Mehraufwendungen nicht.

11.2.2 Zugriffe

Auch hier gilt die Regel, dass nicht mehr als funktional unabdingbar gesperrt werden sollte. Übertriebenes Sicherheitsdenken mindert die Leistung. Eventuell kann durch Zulassen von Überbuchungen oder Entflechten von Sätzen eine Sperre oder gar der SHARUPD-Modus ganz vermieden werden.

Das Erzwingen eines Konsistenzzustandes einer offenen Datei (RELSE-Aufruf) sollte wegen der damit verbundenen physikalischen Ein-/Ausgaben sorgfältig dosiert werden.

Schlüsselzugriffe in ISAM sind aufwendiger als sequenzielle Zugriffe. Gegebenenfalls kann man sie nach Umorganisation des Programms oder der Datei teilweise durch sequenzielle Zugriffe ersetzen.

Muss auf Dateien in einem Mehrrechnersystem zugegriffen werden, sollte man abschätzen, ob sich eine Vorabübertragung der Datei oder von Teilen von ihr lohnt. Das hängt von der Struktur der Datei und der Zugriffe ab. Von erheblicher Bedeutung ist auch, ob ein MRS-Zugriff über das Netz oder über von mehreren Rechnern ansprechbare Platten, die im Shared-Pubset- oder Shared-Private-Disk-Modus betrieben werden, erfolgen kann.

Werden Dateien aufgebaut oder erweitert, kann durch Umgehung der „Secondary Space Allocation“ eine Leistungsverbesserung erzielt werden.

Die dadurch entstehende kompaktere Dateistruktur hat langfristige und vielfältige positive Auswirkungen auf die Leistung.

An dieser Stelle sei noch erwähnt, dass mit DAB auch Teile einer Datei selektiv unterstützt und überwacht (Report „Reads bzw. Writes for internal area“ der Reportgruppe DAB von openSM2) werden können.

11.2.3 CLOSE

Bei geöffneten Dateien ist zu beachten, dass sie immer für bestimmte Gruppen von Anwendern gesperrt sind. Zudem belegen sie Speicher und ihr Abbild auf der Platte ist inkonsistent. Es empfiehlt sich daher, die Datei sofort zu schließen, sobald sie nicht mehr benötigt wird. Es sollten dadurch aber keine unnötigen OPEN-Aufrufe provoziert werden. Bei Banddateien ist mit dem CLOSE ein Rückspulen des Magnetbands verbunden, welches weitere Zugriffe zu dem betreffenden Laufwerk für einige Zeit blockiert. Deshalb ist es auch ratsam, den Aufruf so rechtzeitig zu geben, dass nachfolgende Operationen nicht behindert werden.

Ist die Datei in ihrem Inhalt stark geschrumpft und wird sie längere Zeit klein bleiben, dann sollte sie nach CLOSE mit `/MODIFY-FILE-ATTRIBUTES (Operand SPACE=*RELEASE(...))` bzw. dem Makro `FILE` (mit negativem Parameter `SPACE`) reduziert werden.

11.3 Performantes Laden von Programmen/Produkten

Zunächst werden grob die Vorgänge beim Laden eines Programms/Produkts erläutert. Davon ausgehend werden die Maßnahmen beschrieben, die der Anwender/Produktsteller berücksichtigen sollte, damit die Ladezeit eines Programms/Produkts optimiert werden kann.

Es wird dabei i.d.R. davon ausgegangen, dass die Programme bzw. Module in PLAM-Bibliotheken abgelegt sind. Das Binden der Programme/Module erfolgt deshalb mit BINDER, das Laden mit /START-EXECUTABLE-PROGRAM.

Die Ausführungen beschränken sich auf das Arbeiten mit LLMs. Auf das Binden und Laden von Phasen wird nicht explizit eingegangen, obwohl sicher einige der hier aufgeführten Beschleunigungsmaßnahmen ebenfalls angewendet werden können.

11.3.1 Prinzipieller Ablauf beim Binden/Laden eines Programms/Produkts

Schema der beteiligten Instanzen (die zu ladenden LLMs befinden sich in der Bibliothek, die im Ladeaufruf angegeben wird):

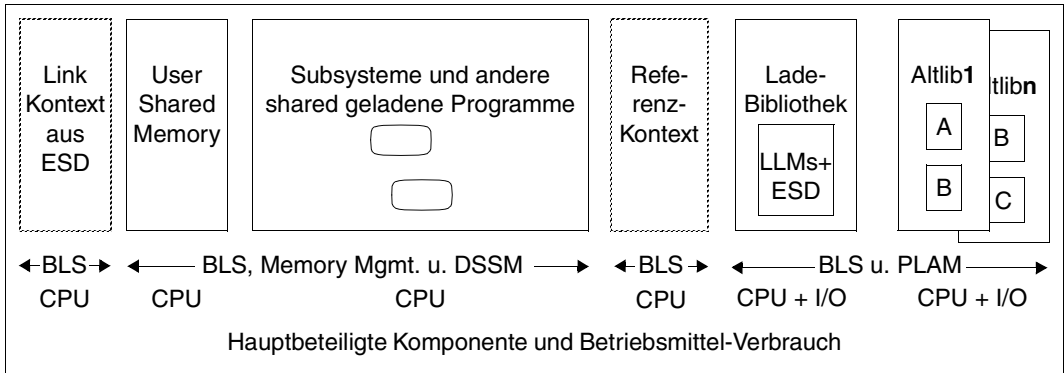


Bild 21: Ablauf beim Binden/Laden eines Programms/Produkts

Die Reihenfolge beim Auflösen (Resolve) der externen Referenzen geht von links nach rechts, d. h. die Instanzen werden in folgender Reihe durchsucht:

- Link-Kontext (gemäß ESD-Info in LLM); er ist leer, falls es sich um ein „stand alone“-Programm handelt (bei /START-EXECUTABLE-PROGRAM)
- User-Shared-Memory (= Memory-Pools); bei /START-EXECUTABLE-PROGRAM können mehrere selektiv angegeben werden (Default: ausgeschaltet)
- (vorgeladene) DSSM-Subsysteme und andere shared geladene Programme (Default: bei /START-EXECUTABLE-PROGRAM eingeschaltet)
- Referenz-Kontext im Speicher: wird im Laufe des Ladevorgangs dynamisch aufgebaut und zum Befriedigen von Externverweisen verwendet, die im Link-Kontext nicht aufgelöst werden konnten
- Bibliothek, die im Ladeaufruf angegeben wird
- Alternative Bibliotheken (AltLibs) nach spezifizierter Reihenfolge

11.3.2 Allgemeine Hinweise zur Beschleunigung des Ladevorganges

- Alle beteiligten PLAM-Bibliotheken sollten so gut wie möglich organisiert sein, d.h.:
 - sie sollten keine überflüssigen LLMs enthalten
 - sie sollten vollständig reorganisiert sein, sodass die erhaltenen LLMs nicht in unnötig viele Extents aufgeteilt sind
 - die Bibliothek selbst sollte aus so wenig Extents wie möglich bestehen
 - die Bibliothek sollte eventuell mit (STD,2) eingerichtet sein

Mit all diesen Maßnahmen wird die Anzahl der Ein-/Ausgaben reduziert und damit die Laufzeit verbessert; der Effekt ist natürlich abhängig von der Ausgangsstruktur. Eine mit (STD,2) eingerichtete Bibliothek wird um etwa 1 KB pro Element größer sein als die Original-Bibliothek; der Ein-/Ausgabe-Einsparungseffekt wird um so größer sein, je mehr (nicht zu kleine) Elemente die Bibliothek hat. Die Laufzeitverbesserung kann über 10% betragen.

- Beim Laden eines LLM-Objekts sollte immer das Kommando `/START-EXECUTABLE-PROGRAM` verwendet werden. Damit werden effektivere BLS-Suchverfahren angestoßen und etwa 10% der CPU-Zeit eingespart.
- Falls Module mit dem Attribut `READ-ONLY` versehen sind, sollte ein Slice für Read-Only-Module und ein Slice für Read-Write-Module erzeugt werden. Dies beschleunigt den Ladevorgang sowohl durch geringeren CPU-Bedarf als auch durch Einsparung von Ein-/Ausgaben.
- Die LLM-Größe und damit der PLAM-Aufwand werden reduziert, wenn alle Symbole, die für spätere `BINDER`-Läufe nicht mehr benötigt werden, `invisible` gesetzt werden; `BINDER`-Anweisung:

```
//MODIFY-SYMBOL-VISIBILITY SYMBOL-NAME=...,SYMBOL-TYPE=...,
SCOPE=...,VISIBLE=*NO,...
```

- Falls `Public-/Private-Slices` Anwendung finden, sollte `Format 2` zur CPU-Reduzierung von BLS und DSSM verwendet werden (siehe `BINDER`-Anweisung `//SAVE-LLM ... FOR-BS2000-VERSIONS=*FROM-V11`).

Mit dem Operanden `SUBSYSTEM-ENTRIES` bei der `BINDER`-Anweisung `//START-LLM-CREATION` kann ein symbolischer Name für die Verknüpfungen zwischen dem `Private`- und dem `Public-Slice` angegeben werden. Dieser Name muss auch im `DSSM`-Katalog definiert sein. (Mit dem alten Format wird für jede Verbindung `DSSM` aufgerufen.)

- Es ist darauf zu achten, dass in der Parameterdatei von `DSSM` der Parameter `LOGGING` **nicht** auf `ON` gesetzt ist (Default: `OFF`). `ON` würde dazu führen, dass beim Laden von Subsystemen `DSSM`-spezifische Daten protokolliert werden.

- RESOLVE-BY-AUTOLINK aus **mehreren** Bibliotheken:
Statt der BINDER-Anweisung `//RESOLVE-BY-AUTOLINK LIBRARY=A,B,C` ist es performanter, wie folgt zu spezifizieren:

```
//RESOLVE-BY-AUTOLINK LIBRARY=A  
//RESOLVE-BY-AUTOLINK LIBRARY=B  
//RESOLVE-BY-AUTOLINK LIBRARY=C
```

Dadurch werden beim dynamischen Nachladen z.B. Verweise, die beim Befriedigen aus Bibliothek B neu entstehen, nur noch in Bibliothek B (und evtl. C) und nicht zuerst in Bibliothek A (und dann in B und evtl. C) gesucht.

11.3.3 Strukturelle Maßnahmen zur Reduzierung des Betriebsmittelbedarfs

Es ist leider nur selten möglich, den Effekt jeder einzelnen der folgenden Maßnahmen genauer zu beziffern. Er hängt zu stark von der Struktur der Ladeobjekte, Subsysteme und Bibliotheken ab. Man kann jedoch i.d.R. davon ausgehen, dass pro Maßnahme ein Mindestgewinn von 10% eintritt.

Bibliotheken reduzieren bzw. deren Inhalt optimieren

- Mischen der Altlibs

Durch Mischen der Altlibs z.B. in eine einzige Bibliothek (dies könnte sogar die spezifizierte Lade-Bibliothek sein) werden die Suchvorgänge in BLS/PLAM stark reduziert, wodurch z. T. deutlich mehr als 10% an Ein-/Ausgaben und CPU-Zeit eingespart werden.

Das Mischen ist i.A. nur dann anzuraten, wenn keine namensgleichen CSECTs oder ENTRIES existieren. Falls doch, so muss beim Mischen darauf geachtet werden, dass die „richtigen“ CSECTs oder ENTRIES übernommen werden.

- Module aus Bibliotheken fest einbinden

Falls sich die Inhalte der Altlibs und der spezifizierten Bibliothek nie bzw. nur selten ändern, empfiehlt es sich eventuell, die Module in das Ladeobjekt fest einzubinden. Dadurch wird der CPU-Bedarf beim Laden verringert. Allerdings wird durch das statische Binden das Ladeobjekt größer und damit i.d.R. mehr Ein-/Ausgaben durchgeführt, was wiederum zur Verlängerung der Laufzeit führt.

Diese Maßnahme ist demnach günstig, wenn nur kleine Module betroffen sind und wenn die Altlib(s) besonders viele Entries enthalten.

Wird diese Variante der Performance-Steigerung gewählt, so ist natürlich darauf zu achten, dass der Bindevorgang bei jeder Altlib-Änderung erneut durchzuführen ist.

- Mischform

Falls sich z.B. der Inhalt von nur einer der beteiligten Altlibs häufig ändert, so empfiehlt es sich eventuell, die Mischform der beiden Methoden zu wählen.

Nachladen aus Shared-Code einschränken

- Kein Nachladen/Binden aus Shared-Code

Viele Programme benötigen kein Nachladen/Linken aus einem Subsystem/Shared-Programm oder User-Shared-Memory. Diese Programme sollten wie folgt gestartet werden:

```
/START-EXECUTABLE-PROGRAM . . . ,
  DBL-PARAMETERS=*PARAMETERS(RESOLUTION=*PARAMETERS(SHARE-SCOPE=*NONE)) ,
  . . .
```

Dadurch wird verhindert, dass DSSM von BLS aufgerufen wird, um die Subsysteme nach dem Entry zu durchsuchen, und es ergibt sich eine z. T. erhebliche Einsparung an CPU-Zeit.

- Nur aus System-Memory nachladen

Dieser Fall ist der Default-Fall. Es gibt keine Möglichkeit, die DSSM-Suche nach dem Entry auf bestimmte Subsysteme oder nur auf Shared-Programme zu beschränken.

- Nur aus User-Shared-Memory nachladen

Bei /START-EXECUTABLE-PROGRAM bietet der Parameter DBL-PARAMETERS=*PARAMETERS(RESOLUTION=*PARAMETERS(SHARE-SCOPE=*MEMORY-POOL(. . .))) die Beschränkung auf User-Shared-Memory. Es ist dabei zusätzlich möglich, sich auf ausgewählte Memory-Pools zu beschränken.

- Vorladen des User-Shared-Memory

Sofern möglich, sollten alle shareable Teile der benutzten Anwendungen vorgeladen werden (Beispiel: FOR1-RTS). Dadurch werden sie nur einmal geladen, und bei jedem weiteren Ladevorgang muss nur noch die (aufwandsarme) Verknüpfung durchgeführt werden. Damit können meist deutlich mehr als 10% der Ein-/Ausgaben und des CPU-Bedarfes eingespart werden.

ESD-Information eliminieren

- Vollkommene Eliminierung

Diese Maßnahme ist nur für „standalone“-Programme durchführbar. Dies sind LLMs, die von anderen Programmen nicht aufgerufen werden und die vollständig vorgebunden sind. Für sie sollte zur Beschleunigung des Ladevorgangs die ESD- und sonstigen BLS-Informationstabellen durch einen entsprechenden BINDER-Aufruf eliminiert werden. Dazu ist der BINDER wie folgt aufzurufen:

```
/START-BINDER
//START-LLM-UPDATE LIB=<original-lib>,ELEMENT=<elem-name>
//SAVE-LLM LIB=<new-lib>,TEST-SUPPORT=*NO,MAP=*NO,
          SYMBOL-DICTIONARY=*NO,LOGICAL-STRUCTURE=*NO,
          RELOCATION-DATA=*NO
//END
```

Durch diese Parameterwahl werden nur die zum Laden notwendigen BLS-Strukturen eines einfachen Programmes erzeugt. Damit wird das Objekt kleiner. Beim Laden des Programms werden somit CPU und weitere Ein-/Ausgaben eingespart. Alle für die Bearbeitung notwendigen Informationen sind danach nicht mehr vorhanden.

Das bedeutet:

- Diese Parameter können nicht für Ladeobjekte, die in mehrere Slices aufgeteilt sind, verwendet werden.
- Mit solch einem Objekt können keine LLM-Updates durchgeführt werden.
- Relocatable Objekte (z.B. als Teil eines Runtime-Systems) können nicht erzeugt werden.

Die (zusätzliche) Parameter-Einstellung `REQUIRED-COMPRESSION=*YES` sollte nicht gewählt werden, da dadurch zwar max. 10% der Ein-/Ausgaben eingespart, aber etwa 40% mehr CPU-Zeit benötigt werden.

- Teilweise Eliminierung

Beim Mischen von LLMs oder Sub-LLMs in einen Großmodul mit einer einzigen CSECT sollte bei der BINDER-Anweisung `//MERGE-MODULES` angegeben werden, welche ESD-Informationen im Extern-Adressbuch bleiben sollen. Damit kann der BLS-Aufwand beim Laden stark reduziert werden.

11.3.4 Beschleunigung des Ladevorganges von C-Programmen

Das Laden von C-Programmen wird beschleunigt, wenn die wenigen nicht vorladbaren Module aus dem C-RTS zum übersetzten Programm dazugebunden werden. Dies wird durch folgende zusätzliche BINDER-Anweisung erreicht:

```
//RESOLVE-BY-AUTOLINK LIB=<Partial-Bind-Lib>
```

Die <Partial-Bind-Lib> wird unter dem Namen SYSLNK.CRTE.PARTIAL-BIND installiert. Außerdem ist das Ausblenden von Extern-Namen zur Unterdrückung von DUPLICATE SYMBOLS notwendig:

```
//MODIFY-SYMBOL-VISIBILITY SYMBOL-NAME=*ALL,VISIBLE=*NO
```

Durch beide Anweisungen werden die CPU-Zeit und vor allem die Laufzeit beim Laden von C-Programmen stark reduziert, wobei das Ladeobjekt nur wenige Seiten (ca. 18) größer als das übersetzte Programm ist.



Damit die CRTE in den Shared-Code geladen wird, müssen die Subsysteme CRTEC und CRTECOM geladen sein.

11.3.5 Nutzung von DAB

Falls das gebundene Objekt und/oder die benutzten Bibliotheken sehr groß sind und deshalb eine hohe Anzahl von Ein-/Ausgaben beim Laden ausgeführt werden muss, sollte der Einsatz von DAB (Disk Access Buffer) geprüft werden.

Mit DAB können bei häufig referenzierten Dateien die Ein-/Ausgaben so stark reduziert werden, dass im Extremfall die Laufzeit nur noch vom CPU-Bedarf abhängt.

Für das Ladeobjekt ist dabei ein Cache-Bereich zu wählen, der genauso groß wie das Objekt ist. Dies ist besonders einfach, wenn das Objekt das einzige Element der Bibliothek ist. Dann wird mit /START-DAB-CACHING AREA=*FILE(FILE-AREA=<lib>),CACHE-SIZE=*BY-FILE ein Hauptspeicher-Bereich von der Größe der Bibliothek reserviert. Dieser wird beim ersten Laden des Objektes gefüllt, sodass bei allen folgenden Ladevorgängen keine zeitintensiven Ein-/Ausgaben mehr notwendig sind.

Die Cache-Bereiche für die Bibliotheken können i.d.R. kleiner gewählt werden als die Gesamtgröße der Bibliotheken, da meist nicht alle enthaltenen Module nachgeladen werden. Dazu ist bei /START-DAB-CACHING die Cache-Größe (in KB oder MB) explizit anzugeben. Mit openSM2 oder /SHOW-DAB-CACHING kann die Cache-Trefferrate jedes Cache-Bereichs überwacht werden. Die Hit-Rate sollte mindestens 80% betragen. Ist sie niedriger, muss der Bereich vergrößert werden.

12 Anhang

12.1 Richtwerte für die BS2000/OSD-Server

In den folgenden Tabellen sind einige Leistungsdaten und Richtwerte für S- und SQ-Server zusammengefasst. Die Empfehlungen gelten für einen typischen TP-Betrieb mit üblicher Hintergrundlast.

- Relativer Performance Faktor RPF und Anzahl der CPUs.
- Typischer Hauptspeicherausbau.
Der tatsächliche Hauptspeicherbedarf hängt ab von der Last, mit der der Server betrieben wird. TP- und Dialog-Lasten haben üblicherweise einen deutlich größeren Hauptspeicherbedarf als Batch-Lasten.

Auf SQ-Servern wird der Hauptspeicher zwischen BS2000/OSD und Dom0/X2000 (I/O-Prozessor) aufgeteilt. Eine Empfehlung für Ausbau und Aufteilung des Hauptspeichers zeigen die entsprechenden Tabellen.
Hinweise darauf, wann mehr als der dort empfohlene Hauptspeicher eingesetzt werden soll, finden Sie im [Abschnitt „Hauptspeicher auf SQ-Servern“ auf Seite 32](#).
- Maximal empfohlene Anzahl der DVS-Ein-/Ausgaben.
Ein IT-System wird wirtschaftlich genutzt, wenn eine ausgewogene Mischung von rechenintensiven und Ein-/Ausgabe-intensiven Programmen abläuft. Die maximal empfohlene Anzahl von DVS-Ein-/Ausgaben pro Sekunde stellt keine technische Obergrenze dar, sondern gewährleistet, dass nur ein bestimmter Teil der CPU-Leistung für die Durchführung von Ein-/Ausgaben verbraucht wird. Eine Überschreitung ist ohne weiteres möglich, sofern Engpässe bei den Peripherie-Geräten vermieden werden.
- Maximal empfohlene Anzahl der Paging-Ein-/Ausgaben.
Die für den Transaktions- bzw. Dialogbetrieb angegebenen Werte für die maximal empfohlene Paging-Rate stellen sicher, dass nur ein gewisser Teil der CPU-Leistung für Paging verbraucht wird. Um zeitliche Verzögerungen durch Paging (besonders bei Transaktionsbetrieb) zu vermeiden, sollte angesichts des verfügbaren Hauptspeichers die Unterschreitung der genannten Werte angestrebt werden.
- Maximal empfohlene Anzahl SVCs in TU.

Wenn die maximale Anzahl SVC-Aufrufe pro Sekunde erreicht wird, dann beträgt der Aufwand für die Unterbrechungsanalyse und Abschlussbehandlung der Systemroutine (also nicht der Aufwand für die Systemroutine selbst) ca. 5% der CPU-Zeit.

Richtwerte für S-Server

S-Server	RPF	Anzahl CPUs	Typische Hauptspeichergröße in GB	Maximal empfohlene Anzahl (pro Sekunde)			
				DVS-Ein-/Ausgaben	TP-Betrieb	Dialogbetrieb	SVCs in TU
S165 - 1RB	160	1	2	3200	100	300	8000
- 10A	225	1	2	4500	100	300	11000
- 10B	270	1	4	5100	100	300	13500
- 10C	310	1	4	5600	100	300	15000
- 10D	350	1	4	6000	100	300	17000
- 20A	400	2	4	6800	100	300	20000
- 20B	490	2	6	8300	100	300	24000
- 20D	640	2	6	10000	100	300	32000
- 30D	910	3	8	14000	100	300	45000
S175 - 10A	170	1	2	3400	100	300	8500
- 10B	240	1	4	4500	100	300	12000
- 10C	300	1	4	5400	100	300	15000
- 10D	340	1	4	5800	100	300	17000
- 10E	390	1	4	6600	100	300	19500
- 20B	460	2	4	7700	100	300	23000
- 20C	570	2	6	8900	100	300	28500
- 20E	740	2	8	11000	100	300	37000
- 30E	1040	3	12	15000	100	300	52000

Tabelle 2: Richtwerte für S-Server

(Teil 1 von 2)

S-Server	RPF	Anzahl CPUs	Typische Hauptspeichergroße in GB	Maximal empfohlene Anzahl (pro Sekunde)				
				DVS-Ein-/Ausgaben	Paging-Ein-/Ausgaben		SVCs in TU	
					TP-Betrieb	Dialogbetrieb		
S200	- 20	860	2	8	13000	100	300	43000
	- 30	1170	3	12	17000	100	300	58000
	- 40	1520	4	16	21000	100	300	75000
	- 60	2050	6	16	26000	100	300	102000
	- 80	2600	8	20	31000	100	300	130000
	- 100	3100	10	24	37000	100	300	155000
	- 120	3600	12	24	43000	100	300	180000
	- 140	4100	14	32	45000	100	300	205000
	- 150	4300	15	32	47000	100	300	215000
S210	- 20	990	2	12	15000	100	300	49500
	- 30	1390	3	16	20000	100	300	69500
	- 40	1770	4	16	24000	100	300	88500
	- 50	2130	5	20	27000	100	300	106500
	- 60	2480	6	24	30000	100	300	124000
	- 80	3150	8	32	37000	100	300	157500
	- 100	3750	10	40	43000	100	300	187500
	- 120	4300	12	40	47000	100	300	215000
	- 140	4750	14	48	51000	100	300	237500
- 150	5000	15	48	53000	100	300	250000	

Tabelle 2: Richtwerte für S-Server

(Teil 2 von 2)

Richtwerte für SQ-Server

SQ-Server	RPF	Anzahl CPUs	Typische Hauptspeichergroße in GB ¹	Maximal empfohlene Anzahl (pro Sekunde)				
				DVS-Ein-/Ausgaben	Paging-Ein-/Ausgaben		SVCs in TU	
					TP-Betrieb	Dialogbetrieb		
SQ100	-10A	12	1	8	850	7	21	600
	-10B	20	1	8	1000	9	28	1000
	-10C	35	1	8	1200	13	40	1750
	-10D	60	1	12	1500	70	250	3000
	-20A	85	2	12	2000	100	300	4250
	-20D	110	2	12	2400	100	300	5500
	-30D	155	3	16	3200	100	300	7750
	-40D	200	4	16	3800	100	300	10000
SQ200	-10A	12	1	8	850	7	21	600
	-10B	20	1	8	1000	9	28	1000
	-10C	42	1	16	1400	13	40	2100
	-10D	75	1	16	1800	70	250	3750
	-10E	125	1	16	2800	70	250	6250
	-20A	175	2	16	3600	100	300	8750
	-20E	225	2	16	3900	100	300	11250
	-30E	320	3	24	5500	100	300	16000
	-40E	410	4	24	7000	100	300	20500
	-50E	490	5	24	8200	100	300	24500
	-60E	570	6	32	9500	100	300	28500
	-80E	700	8	32	11500	100	300	35000

Tabelle 3: Richtwerte für SQ-Server

(Teil 1 von 2)

SQ-Server	RPF	Anzahl CPUs	Typische Hauptspeichergröße in GB ¹	Maximal empfohlene Anzahl (pro Sekunde)			
				DVS-Ein-/Ausgaben	TP-Betrieb	Dialogbetrieb	SVCs in TU
SQ210 -10A	12	1	16	850	7	21	600
-10B	20	1	16	1000	9	28	1000
-10C	42	1	16	1400	13	40	2100
-10D	75	1	16	1800	70	250	3750
-10E	125	1	16	2800	70	250	6260
-10F	175	1	16	3600	70	250	8750
-20A	240	2	16	4200	100	300	12000
-20F	320	2	16	5500	100	300	16000
-30F	460	3	32	7700	100	300	23000
-40F	600	4	32	10000	100	300	30000
-50F	720	5	32	11800	100	300	36000
-60F	840	6	32	13400	100	300	42000
-80F	1050	8	48	16200	100	300	52500
-100F	1240	10	48	18600	100	300	62000
-120F	1420	12	48	20400	100	300	71000
-160F	1750	16	64	23700	100	300	87500

Tabelle 3: Richtwerte für SQ-Server

(Teil 2 von 2)

¹ Um den für BS2000-Gastssysteme verfügbaren Speicher zu ermitteln, ist von dieser Größe der Anteil für Dom0/X2000 (Standard: 30%) und gegebenenfalls der Hauptspeicheranteile für Linux- oder Windows-Gastssysteme abzuziehen. Ferner ist zu beachten, dass ca. 40% (abhängig vom Systemparameter JTABSMEM, siehe [Seite 348](#)) des für BS2000/OSD zur Verfügung stehenden Hauptspeichers als JIT-Puffer verwendet wird und daher dem Betriebssystem und den Anwendungen nicht direkt zur Verfügung steht, siehe [Abschnitt „Hauptspeicher auf SQ-Servern“ auf Seite 32](#).

Beispiel: An einer SQ210-40F (Hauptspeicher: 32 GB) werden 8 GB des Speichers für ein Linux-Gastssystem verwendet. BS2000/OSD und die BS2000-Anwendungen können dann nach folgender Berechnung etwa 8,6 GB Hauptspeicher nutzen: 32 GB - 8 GB (Linux-Gastssystem) - 32 * 0,3 GB (Dom0/X2000) = 14,4 GB * 0,6 (JIT) = 8,64 GB.

12.2 Performance-relevante Systemparameter

Mit Hilfe der BS2000/OSD-Systemparameter (die im Parametersatz SYSOPT-CLASS2 der Parameterdatei SYSPAR.BS2.version definiert sind) kann das Systemverhalten den jeweiligen Erfordernissen angepasst werden.

In diesem Abschnitt werden die folgenden, performance-relevanten Systemparameter beschrieben:

BMTNUM (Seite 337)	EACTETYP (Seite 343)	MSGCENTL (Seite 343)
CATBUFR (Seite 337)	EAMMEM (Seite 344)	MSGCENTN (Seite 343)
CONSDDE7 (Seite 338)	EAMMIN (Seite 344)	MSGDLAM (Seite 343)
DEFLUID (Seite 339)	EAMSEC (Seite 344)	MSGFILii (Seite 343)
DESTLEV (Seite 339)	EAMSIZ (Seite 344)	MSGNOFL (Seite 343)
DMPRALL (Seite 340)	ETMFXLOW (Seite 345)	SSMAPRI (Seite 346)
DMSCALL (Seite 340)	L4MSG (Seite 343)	SSMASEC (Seite 346)
DMMAXSC (Seite 340)	L4SPDEF (Seite 345)	TEMPFILE (Seite 347)

Die vollständige Liste der Systemparameter und ihre detaillierte Beschreibung finden Sie im Anhang des Handbuchs „Einführung in die Systembetreuung“ [10].

BMTNUM und CATBUFR

Verwaltung und Zugriffe zum Dateikatalog TSOSCAT werden intern durch die Katalogverwaltung (Catalog Management System, CMS) ausgeführt. Funktionsbeschreibung und weitere Anregungen für das Tuning werden im [Abschnitt „Einrichten von Systemdateien“ auf Seite 156](#) behandelt.

Die Systemparameter BMTNUM und CATBUFR sollten immer zusammen betrachtet werden. Bei der Entscheidung, welche Werte diesen Parametern zugeordnet werden, ist zu berücksichtigen, mit wie vielen Pubsets das System betrieben werden soll (Einfluss siehe bei CATBUFR).

Parametername	Werte-Bereich	STD-Wert
BMTNUM	0 ... 255	32

Der Parameter BMTNUM legt die Anzahl der Ein-/Ausgabe-Puffer und Pufferverwaltungstabellen (Buffer Management Tables, BMT) für die Katalogverwaltung (CMS) fest. Diese Anzahl Puffer wird einmal für jeden Pubset und einmal global für alle Private Volumes angelegt.

Jeder Block des TSOSCAT kann max. 13 Datei- oder 18 JV-Einträge enthalten. Zu jedem CMS-Ein-/Ausgabe-Puffer (4096 Byte) wird eine Tabelle (BMT) angelegt (ca. 178 Bytes). Über sie wird der aktuelle Zustand des Blocks im Puffer verwaltet.

Der Standardwert für BMTNUM ist 32.

Jede Erhöhung vergrößert auch die Wahrscheinlichkeit, dass ein gesuchter Datei- oder JV-Eintrag vom CMS ohne Durchführung einer physikalischen Eingabe-Operation gefunden wird. Besonders positiv macht sich diese Strategie bemerkbar, wenn häufig auf benachbart abgelegte Katalogeinträge und somit auf ein und denselben Katalogblock zugegriffen wird.

Parametername	Werte-Bereich	STD-Wert
CATBUFR	N/Y	N

Durch Belegen des Parameters CATBUFR mit dem Wert Y werden die CMS-Puffer und die BMTs resident angelegt (im Klasse-3-Speicher).

Mit dem Wert N (Standard) werden sie seitenwechselbar angelegt (im Klasse-4-Speicher).

Die pubset-spezifische Einstellung der Anzahl CMS-Puffer und deren Speicherklasse erfolgt mit Hilfe der Kommandos

```
/ADD-MASTER-CATALOG-ENTRY
/MODIFY-MASTER-CATALOG-ENTRY
/IMPORT-PUBSET
```

und den Operanden `NUMBER-OF-BUFFERS` und `RESIDENT-BUFFERS`.

Dabei wird in folgender Reihenfolge nach definierten Werten gesucht:

1. Explizite Parameterangabe bei `/IMPORT-PUBSET`
2. Angabe bei `/ADD-MASTER-CATALOG-ENTRY` bzw. `/MODIFY-MASTER-CATALOG-ENTRY`
3. Angabe über die Systemparameter `BMTNUM` und `CATBUFR`

Werden keine Privatplatten eingesetzt, so kann mit `BMTNUM=0` die Pufferbelegung für Privatplatten verhindert werden. Mit `/MODIFY-MASTER-CATALOG-ENTRY` (für Home- und Paging-Pubset) bzw. `/IMPORT-PUBSET` muss dann die Anzahl der CMS-Puffer und deren Speicherklasse spezifiziert werden.

Die Größe des Adressraumbedarfs pro Pubset oder der Private Volumes für die Puffer der Katalogverwaltung erhält man durch folgende Formel:

$$\text{CMS-Puffer (Byte)} = (4096 + x) * \text{Anzahl Puffer} + 384$$

$x = 178$ für Pubsets; $x = 200$ für Private Volumes

Werden z.B. bei kleineren Servern viele speicherresidente Puffer angelegt, so werden zwar die Katalogoperationen schneller, aber eine eventuell erhöhte Paging-Rate kann diesen Vorteil ausgleichen. Bei starker Katalogbeanspruchung ist der Einsatz des Produkts SCA (Speed Catalog Access) unerlässlich.

CONSDDE7

Parametername	Werte-Bereich	STD-Wert
CONSDDE7	N/Y	N

Der Systemparameter `CONSDDE7` dient zur Reduzierung des Meldungsumfangs auf der Bedienstation.

Die Systemmeldung `DMS0DE7 SAM FILE CLOSED` wird bei Y auf `SYSOUT` und Bedienstation, bei N nur auf `SYSOUT` ausgegeben.



Wenn für `SPOOL` ein Magnetband als Ausgabegerät zugewiesen ist, erscheint diese Meldung am Ende jeder `SPOOL`-Ausgabe. Standardwert ist N.

DEFLUID

Parametername	Werte-Bereich	STD-Wert
DEFLUID	:catid:\$userid	\$TSOS

In den Kommandos `/LOAD-` bzw. `/START-(EXECUTABLE-)PROGRAM`, `/CALL-PROCEDURE`, `/ENTER-PROCEDURE` und `/ENTER-JOB` kann beim Operanden `FROM-FILE` auch dann die Angabe der Katalog- und Benutzerkennung entfallen, wenn die benötigte Datei nicht unter der Kennung des Aufrufers katalogisiert ist.

In einem zweiten Suchlauf (Secondary-Read) wird in der Kennung gesucht, die bei diesem Systemparameter angegeben ist.

Die Möglichkeit, hier eine andere Kennung als den Standardwert `$TSOS` anzugeben, dient zur Entlastung der Kennung `TSOS`. Es wird empfohlen, alle vom Rechenzentrum den Anwendern zur Verfügung gestellten Programme und Verfahren unter einer eigenen Bibliotheks-Benutzerkennung zu speichern, und diese dem System mit der Option `DEFLUID` bekanntzugeben.

DESTLEV

Parametername	Werte-Bereich	STD-Wert
DESTLEV	0/1/4/5/6	0

Mit `DESTLEV` wird festgelegt, ob Datei-Extents beim Freigeben vom System (`/DELETE-FILE` oder `/MODIFY-FILE-ATTRIBUTES` mit Speicherplatzfreigabe) mit binären Nullen überschrieben werden sollen.

Bei großen Dateien kann dieser Vorgang sehr lange dauern (im Minuten-Bereich).

DMPRALL, DMSCALL, DMMAXSC

Diese drei Systemparameter sind aufeinander abgestimmt festzulegen. Sie dienen als Voreinstellungen für die Platzzuweisung beim Einrichten und Erweitern von Dateien. Die Werte gelten für alle Private Volumes. Für Pubsets gelten sie nur, wenn im MRSCAT-Eintrag des Pubsets nichts anderes festgelegt wurde. Zusätzlich wird die beim Einrichten des Pubsets mit SIR angegebene Allocation-Unit abhängig vom Datenformat (K, NK2, NK4) berücksichtigt.

Dabei wird in folgender Reihenfolge nach definierten Werten gesucht:

1. Direkte Angabe in den Kommandos `/CREATE-FILE` oder `/MODIFY-FILE-ATTRIBUTES`
2. Eintrag im MRSCAT
 - Erfolgt die Angabe mit `/MODIFY-MASTER-CATALOG-ENTRY` (Operand `ALLOCATION`), so wird die Änderung erst nach dem nächsten Importieren des Pubsets gültig.
 - Bei Verwendung von `/MODIFY-PUBSET-SPACE-DEFAULTS` (Operand `PRIMARY-ALLOCATION` / `SECONDARY-ALLOCATION` / `MAX-ALLOC`) wird die Vereinbarung sofort gültig.
3. Angabe über Systemparameter

Wenn die so ermittelten Werte kein Vielfaches der beim Einrichten des Pubsets mit SIR festgelegten Allocation Unit sind, dann werden sie auf das nächste Vielfache aufgerundet.

Primärzuweisung

Parametername	Werte-Bereich	STD-Wert
DMPRALL	3..65535	3

Primärzuweisung für eine Datei in PAM-Blöcken, wenn bei `/CREATE-FILE` oder `/MODIFY-FILE-ATTRIBUTES` (bzw. im FILE-Makro) der Operand `SPACE` nicht versorgt wurde.

Sekundärzuweisung

Parametername	Werte-Bereich	STD-Wert
DMSCALL	3..65535	3

Sekundärzuweisung für eine Datei in PAM-Blöcken, wenn bei `/CREATE-FILE` oder `/MODIFY-FILE-ATTRIBUTES` (bzw. im FILE-Makro) diese im Operand `SPACE` nicht versorgt wurde.

Nach jeder Dateierweiterung wird der Betrag des Zuwachses verdoppelt gegenüber dem Wert bei der vorherigen Erweiterung. Secondary-Allocation (SA) bei (i+1)-ter Erweiterung gleich zweimal Secondary-Allocation bei i-ter Erweiterung: $SA(i+1) = 2 * SA(i)$

Diese Formel wird so lange angewendet, bis der doppelte Wert der aktuellen Sekundärzuweisung den Wert DMMAXSC überschreitet. Danach gilt: $SA(i+1) = SA(i)$

Maximalwert

Parametername	Werte-Bereich	STD-Wert
DMMAXSC	3..65535	48

Wenn der doppelte Wert der aktuellen Sekundärallokierung den Wert von DMMAXSC überschreitet, dann wird der Wert für die Sekundärallokierung nicht mehr verändert.

Beispiel

Dateierweiterungen mit den Werten: DMPRALL 15, DMSCALL 3, DMMAXSC 48.

In diesem Beispiel könnte DMMAXSC jeden Wert zwischen 48 und 96 enthalten, ohne dass sich das Verhalten ändert.

15 + 3 = 18	(1. Dateierweiterung)
18 + 6 = 24	(2. Dateierweiterung)
24 + 12 = 36	(3. Dateierweiterung)
36 + 24 = 60	(4. Dateierweiterung)
60 + 48 = 108	(5. Dateierweiterung)
108 + 48 = 156	(6. Dateierweiterung, keine Veränderung der Sekundärallokierung, da der Wert dann größer wäre als DMMAXSC)



Beim Einrichten einer Datei auf Platte oder Verändern ihrer Größe ist DVS-intern der kleinste Betrag, um den eine Änderung wirksam wird, eine Allocation-Unit.

Die kleinste Allocation-Unit beträgt für

Private Volumes: 3 PAM-Blöcke

Pubsets: 3 PAM-Blöcke (K- und NK2-Format)
4 PAM-Blöcke (NK4-Format)

Jede dynamische Dateierweiterung führt zu einer Reihe von Verwaltungs-Ein-/Ausgaben im System (Aktualisierung der Einträge in der Benutzererkennung, im Dateikatalog, im F1-Kennsatz bei Private Volumes) und verursacht somit einen beträchtlichen Systemaufwand. Um diesen Aufwand zu minimieren, sollte der DMSCALL-Parameter auf mindestens vier Allocation-Units gesetzt werden.

Um die Anzahl der dynamischen Dateierweiterungen zu reduzieren (bzw. zu vermeiden), steht jedem Anwender bei `/CREATE-FILE` und `/MODIFY-FILE-ATTRIBUTES` der Operand `SPACE=*RELATIVE(PRIMARY-ALLOCATION=xx, SECONDARY-ALLOCATION=yy)` zur Verfügung.



Jeder Anwender sollte grundsätzlich versuchen, die Datei schon bei der Erstellung durch eine entsprechende Primärzuweisung in der voraussichtlichen Endgröße einzurichten.

Beispiel

Ein Programm schreibt mit dem PAM-Makroaufruf sequenziell 456 Blöcke. Mit dem AINF-Makroaufruf wird die Anzahl der Ein-/Ausgabe-Operationen ermittelt.

/CREATE-FILE . . . ,SPACE=REL (PRI-ALLOC= . . . , SEC-ALLOC= . . .)		Anzahl phys. Ein-/Ausgaben lt. AINF	Anzahl Extents lt. /SHOW-FILE-ATTR
3	3	480	20
6	6	476	18
30	30	467	9
192	192	459	3
456	30	456	1

Abhängig von der jeweiligen Plattenbelegung können die Ergebnisse etwas streuen.

EACTETYP, L4MSG, MSGCENTL, MSGCENTN, MSGDLAM, MSGFILi, MSGNOFL

Diese Parameter gehören zusammen:

Parametername	Werte-Bereich	STD-Wert	Bedeutung
EACTETYP	0/1/2/3	0	Legt fest, welche der folgenden Meldungen über die Bedienstation ausgegeben werden sollen: BLS0500, BLS0517, BLS0519, BLS0523, BLS0524, BLS0526, BLS0539, BLS0551, BLS0552. 0 keine der Meldungen 1 nur die Meldung BLS0519 2 alle oben genannten Meldungen 3 alle Meldungen außer BLS0519
L4MSG	0/1	0	Steuert die Ausgabe der Meldung EXC044E "ACCEPT ALLOCATION REQUEST...". 0 Meldung wird nicht ausgegeben 1 Meldung wird ausgegeben
MSGCENTL	36...2500	200	Legt die Länge eines Eintrags im Klasse-4-Speicher für die Meldungsbearbeitung fest. Der Wert muss ein Vielfaches von 4 sein. In diesem Bereich des Klasse-4-Speichers werden die zuletzt und am häufigsten benutzten Meldungen zwischengespeichert, um Dateizugriffe zu sparen.
MSGCENTN	0...32767	32	Legt die Anzahl von Einträgen im Klasse-4-Speicher (s.o.) für die Meldungsbearbeitung fest. <i>Hinweis:</i> Der Bedarf an Klasse-4-Speicher kann durch Multiplizieren der Werte von MSGCENTL und MSGCENTN berechnet werden.
MSGDLAM	0...99	0	Anzahl der Meldungsdateien, die über DLAM verarbeitet werden sollen.
MSGFIL01 MSGFIL02 MSGFIL03 MSGFIL04 : MSGFIL15 *)	dateiname(n)		Anzahl und Dateinamen der Meldungs-Ausgabedateien. Für die Parameter MSGFIL01 und MSGFIL02 gilt folgende Standardeinstellung: MSGFIL01: SYSMES.BS2CP.version MSGFIL02: SYSMES.EKP.01 Für die Parameter MSGFIL03 bis MSGFIL15 gibt es keine Standardnamen, d.h. diese Dateien werden nicht eingerichtet. Der Dateiname muss vollqualifiziert sein und kann nur mit der Benutzerkennung \$TSOS katalogisiert werden.
MSGNOFL *)	0...15	2	Anzahl der Meldungsdateien, die durch die Parameter MSGFILxx spezifiziert wurden.

*) für Details siehe [Abschnitt „Meldungsdateien“ auf Seite 169](#)

EAMMEM, EAMMIN, EAMSEC, EAMSIZ

Parametername	Werte-Bereich	STD-Wert
EAMMEM	0...2730	0
EAMMIN	4...64512	3000
EAMSEC	1...64512	200
EAMSIZ	4...64512	64512

Mit diesen Parametern wird die Größe der Datei `SYSEAM` definiert. Diese Parameter gelten für alle Dateien `:cat id:SYSEAM`, wenn sie auf mehreren Pubsets eingerichtet sind und im MRSCAT-Eintrag pro Pubset nichts anderes festgelegt ist. Zusätzlich wird die beim Einrichten des Pubsets mit SIR angegebene Allocation-Unit abhängig vom Datenformat (K, NK2, NK4) berücksichtigt.

Die Datei `SYSEAM` wird in EAM-Allocation-Units verwaltet. Bei einer DMS-Allocation-Unit von 3 PAM-Seiten ist die EAM-Allocation-Unit ebenfalls 3 PAM-Seiten; bei allen anderen DMS-Allocation-Units ist die EAM-Allocation-Unit 4 PAM-Seiten. Mit dem Wert `EAMMIN` wird die Minimalgröße festgelegt. Bei Bedarf wird sie um den Wert `EAMSEC` (Anzahl in EAM-Allocation-Units) erweitert, bis die Maximalgröße von 64.512 EAM-Allocation-Units erreicht ist. Sollte die Datei `SYSEAM` schrumpfen (z.B. bei `/DELETE-SYSTEM-FILE SYSTEM-FILE=*OMF, /EXIT-JOB` u.a.), wird sie in Einheiten von 8 Allocation-Units verringert, jedoch nicht unter den Wert von `EAMMIN`.

Der Wert von `EAMMIN` ist so einzustellen, dass im Normalbetrieb 80% aller Anforderungen zum Speichern von Blöcken in der Systemdatei `SYSEAM` befriedigt werden können, ohne dass eine Erweiterung der Datei notwendig wird. Als Erfahrungswert für die Ersterstellung ist eine Größe von 4000 Allocation-Units angezeigt.

Wird der Wert für `EAMMIN` zu klein gewählt, so treten als Folge eine höhere Systembelastung durch viele dynamische Dateierweiterungen und ein unkontrolliertes Wachsen der Datei `SYSEAM` ein.

Durch geeignete Wahl von `EAMMIN` und `EAMSEC` bzw. des Operanden `EAM (MINIMAL-SIZE, SECONDARY-ALLOCATION)` bei `/ADD-MASTER-CATALOG-ENTRY` und `/MODIFY-MASTER-CATALOG-ENTRY` kann das Pulsieren der Dateigröße von `SYSEAM` auf ein Minimum beschränkt oder ganz unterbunden werden.

Wesentlich ist es, die Anzahl der erforderlichen Sekundärzuweisungen pro Pubset so gering wie möglich zu halten.

`EAMSIZ` ist der Maximalwert an Allocation-Units innerhalb der Datei `$TSOS.SYSEAM`, die einem Anwender allein zur Verfügung stehen.

Richtwerte für die EAM-Systemparameter sind stark von der zu erwartenden Last abhängig. Die Werte für `EAMMIN` und `EAMSEC` sollten beide als Vielfache von 8 gewählt werden, weil dies eine optimale Anpassung an die interne Tabellenstruktur von EAM bewirkt.

Der Wert EAMMEM legt die Größe des Klasse-4-Speichers (in Units) fest, der für EAM benutzt wird. Er sollte ein Vielfaches von 8 sein. Die pubset-spezifische Festlegung erfolgt durch den Operanden `EAM=*PARAMETERS(VIRTUAL-MEMORY=...)` bei `/ADD-MASTER-CATALOG-ENTRY` bzw. `/MODIFY-MASTER-CATALOG-ENTRY`.

Dieser Bereich wird nur für den Home-Pubset angelegt und ist Teil der Datei `SYSEAM`. Zuerst wird der Klasse-4-Speicherbereich gefüllt und erst danach – wenn dieser voll ist – auf die Platte geschrieben.

Eine sinnvolle Nutzung (Beschleunigung von Übersetzungsläufen durch Einsparung von Ein-/Ausgaben) ist nur möglich, wenn ausreichend Hauptspeicher zur Verfügung steht. Andernfalls wird der Vorteil der Einsparung von physikalischen Ein-/Ausgaben auf die Datei `SYSEAM` durch erhöhtes Paging der Klasse-4-Speicherseiten kompensiert.

ETMFXLOW

Parametername	Werte-Bereich	STD-Wert
ETMFXLOW	127...256	256

Tasks mit einer externen Priorität größer oder gleich ETMFXLOW erhalten im Funktionszustand TU keinen Alterungszuschlag. Dadurch wird für diese Tasks die dynamische Einstellung der internen Priorität (siehe [Abschnitt „Einführung in das PRIOR-Konzept“ auf Seite 203](#)) abgeschaltet. Ein eventuell im Funktionszustand TPR gewonnener Prioritätszuwachs verfällt beim Übergang nach TU.

Tasks mit einer externen Priorität größer oder gleich ETMFXLOW eignen sich zur Definition einer Hintergrundlast. Für die Hauptanwendung(en) muss eine entsprechend große externe Prioritätsdifferenz gewählt werden. Der Einsatz dieser Funktion ist sehr sorgfältig durchzuführen und die Auswirkungen müssen überwacht werden.

L4SPDEF

Parametername	Werte-Bereich	STD-Wert
L4SPDEF	66...n	2500

Der Parameter L4SPDEF bestimmt den Grenzwert für das Erreichen der Sättigungsstufe 4 in Pubsets. Die Stufe gilt als erreicht, wenn weniger PAM-Blöcke frei sind als hier festgelegt. Es wird empfohlen, die Standardeinstellung zu übernehmen.

Der Wert ist nur gültig, solange keine andere pubset-spezifische Festlegung mit `/ADD-MASTER-CATALOG-ENTRY` bzw. `/MODIFY-MASTER-CATALOG-ENTRY`, Operand `ALLOCATION=*PARAMETERS(SATURATION-LEVEL4=...)`, erfolgt.

SSMAPRI, SSMASEC

Parametername	Werte-Bereich	STD-Wert
SSMAPRI	3..65535	24
SSMASEC	3..65535	24

Mit diesen Parametern wird für Systemausgabedateien die Größe der Primär- und Sekundärzuweisung festgelegt. Diese Parameter gelten für Dateien, die mit folgenden Kommandos angelegt werden:

```
/ASSIGN-SYSOUT
/ASSIGN-SYSLST
```

Jede Erstbenutzung dieser Systemausgabedateien führt zum Einrichten einer katalogisierten Datei in der mit SSMAPRI und SSMASEC definierten Größe. Die Dateinamen sind:

```
S.OUT.tsn.yyyy-mm-dd.hhmmss.cnt
S.LST.tsn.yyyy-mm-dd.hhmmss.cnt
```

- SSMAPRI** Primärzuweisung in PAM-Blöcken für die task-spezifischen SPOOLOUT-Dateien bzw. für umgewiesene Systemdateien, die mit einem /ASSIGN-...-Kommando erzeugt wurden.
- SSMASEC** Sekundärzuweisung in PAM-Blöcken für die task-spezifischen SPOOLOUT-Dateien bzw. für umgewiesene Systemdateien, die mit einem /ASSIGN-...-Kommando erzeugt wurden.

Die optimalen Werte für Primär- und Sekundärzuweisung dieser SPOOL-Ausgabedateien sind stark von der Art der Last abhängig. Liegen keine abweichenden Erfahrungen vor, sollten die Standard-Werte von jeweils 24 PAM-Blöcken beibehalten werden.

TEMPFILE

Parametername	Werte-Bereich	STD-Wert
TEMPFILE	C'#/'@'/'NO'	C'NO'

Das Verfahren „temporäre Dateien“ (TEMPFILE) bietet den Anwendern die Möglichkeit, mit katalogisierten Dateien und/oder Jobvariablen zu arbeiten, die bei Angabe von /EXIT-JOB bzw. /LOGOFF vom System automatisch gelöscht werden.

Das Verfahren ist recht aufwändig, d.h. es wird viel Systemadressraum und CPU-Leistung benötigt. Dafür leistet es ein „automatisches Aufräumen“ von nicht benötigten Arbeitsdateien. Der Systembetreuer kann mit dieser Option bestimmen, ob das Verfahren angewendet werden darf und wie die Datei- und JV-Namen gekennzeichnet werden.

Ist die Option nicht versorgt, wird das Verfahren „temporäre Dateien“ nicht in das System eingebunden (Standard).

Der Katalogname für temporäre Dateien oder JVs lautet:

S.mmm.nnnn.filename

S.mmm.nnnn.jv-name

mmm wird ersetzt durch die Systemkennung (sysid) des Home-Pubsets

nnnn wird ersetzt durch die TSN der Task des Dateibenutzers

12.3 Puffergröße der CISC-Firmware

Folgende Systemparameter steuern auf SQ-Servern ab SQ200 gemeinsam die Hauptspeichernutzung für die Just-in-Time-Übersetzung von /390-Code durch die CISC-Firmware:

JTABSMEM

Parametername	Werte-Bereich	STD-Wert
JTABSMEM	0...65535	0

Mit diesem Parameter wird der maximale Speicherplatz für den JIT-Puffer festgelegt. Der Speicher wird resident angefordert und muss daher auf den Speicherausbau der Maschine abgestimmt sein.

Der Parameter ist mit 0 vorbelegt und die CISC-Firmware berechnet sich in diesem Fall den maximalen Speicherplatz zu: $\text{MEMORY-SIZE} * 40 \%$.

Der berechnete Wert für JTABSMEM wird mit der Meldung HJT0039 auf der Konsole protokolliert. Durch die Angabe eines Wertes von 1 - 65535 kann die Speichergröße auch fest vorgegeben werden. Der Parameter kann im laufenden Betrieb erhöht werden.

Bei Annäherung an den Grenzwert wird die Meldung HJT0035 mit einem entsprechenden JIT-Memory-Saturation-Level ausgegeben. Bei Erreichen von Level 1 wird empfohlen, den Grenzwert zu erhöhen. Bei Erreichen von Level 2 wird dringend empfohlen den Grenzwert zu erhöhen, da sonst größere Performance-Einbußen zu erwarten sind.

JTSTDMEM

Parametername	Werte-Bereich	STD-Wert
JTSTDMEM	1...65535	16

Dieser Parameter JTSTDMEM legt fest, wie viel JIT-Puffer das System jeder Task innerhalb des globalen JIT-Puffers maximal zur Verfügung stellt, sofern task-spezifisch keine andere Einstellung über /MODIFY-DBL-DEFAULTS erfolgt ist.

Die Standardgröße der task-spezifischen JIT-Puffer beträgt 16 MB. Da die CISC-Firmware nur so viel Speicher anfordert, wie zum Ablauf des aktuellen /390-Programms notwendig ist, wird empfohlen den STD-Wert nicht zu verändern. Bei Erreichen des Wertes JTSTDMEM wird die Meldung HJT0032 ausgegeben, der task-spezifische JIT-Puffer zurückgesetzt und neu aufgebaut. Da die damit verbundene Performance-Minderung normalerweise gering ist, wird die Meldung nur in die CONSLOG-Datei ausgegeben.

Nur in sehr seltenen Fällen (z.B. bei besonders großem Working-Set-Bedarf einer Task) kann es vorkommen, dass eine Vergrößerung des JIT-Puffers sinnvoll ist.

Zur task-spezifischen Vergrößerung des JIT-Puffers muss die neue Größe mit folgendem Kommando angegeben werden (nn = Anzahl MB):

```
/MODIFY-DBL-DEFAULTS SCOPE=*CMD-CALLS(CISC-COMPILATION=*YES(WORKSPACE=nn))
```

Die Änderung wirkt für alle DBL-Aufrufe (/START-, /LOAD-(EXECUTABLE-)PROGRAM, BIND) der Task. Das Kommando muss in alle Enter-Jobs bzw. Prozeduren eingefügt werden, wenn dort Programme, die einen größeren JIT-Puffer benötigen, geladen werden.

JTMAXMEM

Parametername	Werte-Bereich	STD-Wert
JTMAXMEM	1...65535	128

Der Parameter JTMAXMEM legt den Maximalwert fest, der task-spezifisch über /MODIFY-DBL-DEFAULTS eingestellt werden kann.

JTSHMEM

Parametername	Werte-Bereich	STD-Wert
JTSHMEM	0...256	64

Legt fest, wieviel Speicherplatz JITSYS zur Ablage von Share-Kompilaten verwenden soll (in MB).

Share-Kompilate entstehen bei der Emulation von in Klasse-4-Speicher geladenen Subsystemen. Sind keine solchen Subsysteme vorhanden bzw. sollen diese nicht „shared“ übersetzt werden, kann mit JTSHMEM = 0 die Erzeugung von „shared“ Kompilaten verhindert werden. Der Wert JTSHMEM sollte in Abstimmung mit dem insgesamt zur Verfügung stehenden Klasse-3-Speicher gesetzt werden.

Die bei JTSHMEM angegebene Speichergröße wird bei Initialisierung von JITSYS sofort allokiert. Eine Änderung des Werts im laufenden Betrieb wird bis auf weiteres nicht unterstützt. Die Einstellung des Wertes kann nur in 4er-Schritten erfolgen, andere Werte werden auf die nächste 4-MB-Grenze aufgerundet.

Eine Änderung des Standardwertes von JTSHMEM sollte synchron zum Parameter BIG-SHARE-SHRSIZE im Parametersatz MEMORY des Startup-Parameterservice erfolgen.

Zum Parameter SHXSIZE des Startup-Parameterservice besteht ein (indirekter) Zusammenhang:

Wenn SHXSIZE sehr groß gewählt und auch tatsächlich eine entsprechende Menge von Programmen „shared“ geladen wird, dann ist es sinnvoll, auch JTSHMEM zu erhöhen.

Literatur

Die Handbücher finden Sie im Internet unter <http://manuals.ts.fujitsu.com>. Handbücher, die auch in gedruckter Form vorliegen, können Sie unter <http://manualshop.ts.fujitsu.com> bestellen.

- [1] **ARCHIVE**
Benutzerhandbuch
- [2] **BCAM**
Benutzerhandbuch
- [3] **BS2000/OSD-BC**
Dateien und Volumes größer 32 GB
Benutzerhandbuch
- [4] **Business Server der SQ-Serie**
Bedienen und Verwalten
Benutzerhandbuch
- [5] **DAB (BS2000/OSD)**
Disk Access Buffer
Benutzerhandbuch
- [6] **BS2000/OSD-BC**
Dienstprogramme
Benutzerhandbuch
- [7] **DRV (BS2000/OSD)**
Dual Recording by Volume
Benutzerhandbuch
- [8] **BS2000/OSD-BC**
DVS-Makros
Benutzerhandbuch
- [9] **BS2000/OSD-BC**
Einführung in das DVS
Benutzerhandbuch

- [10] **BS2000/OSD-BC**
Einführung in die Systembetreuung
Benutzerhandbuch
- [11] **FDDRL (BS2000/OSD)**
Benutzerhandbuch
- [12] **HIPLEX MSCF (BS2000/OSD)**
BS2000-Rechner im Verbund
Benutzerhandbuch
- [13] **HNC**
High-Speed Net Connect
Benutzerhandbuch
- [14] **HSMS (BS2000/OSD)**
Hierarchisches Speicher Management System
Benutzerhandbuch
- [15] **BS2000/OSD-BC**
Kommandos
Benutzerhandbuch
- [16] **LEASY (BS2000/OSD)**
Programmschnittstelle und Konzepte
Benutzerhandbuch
- [17] **BS2000/OSD-BC**
Migration Guide
Benutzerhandbuch
- [18] **openSM2 (BS2000/OSD)**
Software Monitor
Benutzerhandbuch
- [19] **openUTM**
Konzepte und Funktionen
Benutzerhandbuch
- [20] **openUTM**
Anwendungen administrieren
Benutzerhandbuch

- [21] **openUTM**
Anwendungen generieren
Benutzerhandbuch
- [22] **openUTM (BS2000/OSD)**
Einsatz von openUTM-Anwendungen unter BS2000/OSD
Benutzerhandbuch
- [23] **PCS (BS2000/OSD)**
Performance Control Subsystem
Benutzerhandbuch
- [24] **ROBAR (BS2000/OSD, Linux)**
Steuerung von MBK-Archivsystemen
Benutzerhandbuch
- [25] **SDF (BS2000/OSD)**
SDF-Verwaltung
Benutzerhandbuch
- [26] **SESAM/SQL-Server (BS2000/OSD)**
Datenbankbetrieb
Benutzerhandbuch
- [27] **SESAM/SQL-Server (BS2000/OSD)**
Performance
Benutzerhandbuch
- [28] **SHC-OSD / SCCA-BS2**
Storage Management für BS2000/OSD
Benutzerhandbuch
- [29] **SM2-PA (BS2000/OSD)**
SM2-Programmanalysator
Benutzerhandbuch
- [30] **SPACEOPT (BS2000/OSD)**
Optimierung und Reorganisation von Platten
Benutzerhandbuch
- [31] **BS2000/OSD-BC**
Systeminstallation
Benutzerhandbuch

- [32] **BS2000/OSD-BC**
System Managed Storage
Benutzerhandbuch
- [33] **VM2000 (BS2000/OSD)**
Virtuelles Maschinensystem
Benutzerhandbuch

Stichwörter

\$SYSJS 201
\$VMCONS 138
/390-Code 29, 348

A

ADM-PFA-Caching 176, 180
Adressraum 143
affine Tasks 234
AINF (Makro) 342
Änderungen 12
Antwortzeit 16, 17
 Optimierung 188
 Optimierung mit PCS 224
Anwender-Software 315
 Bearbeitungsmodus 321
 performantes Laden 323
 Zugriffsmethode 315
Anwenderdateien 173
Anwenderprogramme 15
Anzahl Terminals (K) 20
ARCHIVE 99
Auftrag 19
Auslastung von Terminals (U) 21
AutoDAB 180, 294
Autolink 326

B

Batch-Auftrag 199
 Auswahlparameter 197
 im XCS-Verbund 198
Batch-Betrieb 25
Batch-Last 25
Batch-Task 243
Batch-Verarbeitung 19
BCAM 188, 195
BCAM-Pool 190, 195
BCAM-Transportsystem 17, 18
Bedienzeit
 Hardware 268
 Software 268
Belegung 236
Benutzeradressraum 143
Benutzerkatalog 146
Betriebsarten von Platten 145
Betriebsmittel
 CPU 244
 Hauptspeicher 244
Betriebsmittelauslastung 271, 285
Betriebsmittelbelegung 236
Betriebsmittelreservierung 236
Betriebsmittelverbrauch 286
Bibliothek 327
Blockgröße 57
BMTNUM (Systemparameter) 157, 160, 337
BTAM (Zugriffsmethode) 315
Bucket 191
Buffer Management Tables 337

C

C-Programme 330
C40 26
Cache 45, 46, 47, 182, 330
Cache-Hit 177
Cache-Miss 177
Caching
 ADM-PFA 176
 User-PFA 176
Catalog Management System 157
CATBUFR (Systemparameter) 160, 337
Chained-IO 315
CISC-Firmware 29, 348
CONSDDE7 (Systemparameter) 338
Control-Volume-Set 161
Controller 45, 47
COSMOS 313
CPU
 Auslastung bei Multiprozessor-Systemen 28
 Leistung 25
 Leistungssteigerung 27
 Verfügbarkeit 27
 Verwaltung 258
 Zeit bei Multiprozessor-Systemen 27
CPU-Pool 122
CPU-Quote 119, 130

D

DAB, siehe Disk Access Buffer
Data-Management 141
Datei 286
 Anwenderdatei 173
 Extent 175
 Reorganisation 175
 schließen 322
 SJMSFILE 200
 SYSEAM 167, 344
 temporär 347
 Zugriffe 322
Dateikatalog
 Formate 157
 SF-Pubset 157
 SM-Pubset 161

Datensicherung

 Differenzsicherung 98
 FDDRL 105
 HSMS/ARCHIVE 99
 logisch 98
 physikalische 98
 Vollsicherung 98
 Volumen 98
Datenträgerüberwachung 236
DCAM-Schnittstelle 17
DEFLUID (Systemparameter) 339
Dehnfaktor 23
Dehnung 23
Denkzeit 17, 20, 23
DESTLEV (Systemparameter) 339
Dialog-Task 243
Dialoganwendung 199
Dialogbetrieb 16, 164, 331
Differenzsicherung 98
Direct Attached Storage (DAS) 38
Directory, HSMS 99
Disk Access Buffer 147, 286, 294, 322, 330
DIV (Zugriffsmethode) 320
DMMAXSC (Systemparameter) 340
DMPRALL (Systemparameter) 340
DMSCALL (Systemparameter) 340
Dual Recording by Volume 145, 148
Durchsatz 13
 Optimierung mit PCS 224
 Steigerung bei Multiprozessor-Systemen 28
Durchsatzrate 19

E

EACTETYP (Systemparameter) 170, 343
EAMMEM (Systemparameter) 168, 344
EAMMIN (Systemparameter) 168, 344
EAMSEC (Systemparameter) 344
EAMSIZ (Systemparameter) 344
EC (Equivalent Copy) 50
Ein-/Ausgabe 265, 315
Ein-/Ausgabe-Unterbrechung 241
Engpass-Analyse mit SM2 282
ESD-Information 329
ETERNUS CS HE 109

ETERNUS DX 145, 175
ETERNUS LT40 109
ETMFXLOW (Systemparameter) 345

F

F1-Kennsatz 341
FASTPAM (Zugriffsmethode) 316
FC-Anschluss 108
FC-Peripherie 33, 36
FDDRL 105
File-Server 86
Fragmentierung 175
Funktionszustand 239
 MEH 239
 SIH 29, 239
 TPR 29, 239
 TU 29, 239

G

Gastsystem 117, 129
 Systembetreuung 135
GB 13
Gerät 33
Geräteverwaltung 236
Gigabit Ethernet 89, 92
Globalspeicher 176, 289
 Caching 176
GSVOL 176

H

Hardware-Bedienzeit 46, 268
Hauptspeicher 31, 176, 244, 246, 252, 280
 Caching 176
 Seitenverwaltungs-Algorithmus 256
Hauptspeichernutzung 31
HIPERBATCH 185
HIPERFILE-Konzept 173, 176, 294
 Cache-Hit 177
 Cache-Hit-Rate 177
 Cache-Miss 177
 Caching-Modi 177
 Read-Cache 177
 Read-Write-Cache 179
 Write-Cache 178

HSMS 99
HSMS-Directory 99
Hypervisor
 VM2000 117
 Xen 117

I

I/O-Prozessoren 27
I/O-Zeit 46
ICMX-Schnittstelle 18
IDCAM-Schnittstelle 18
Inbound Flow Control 190
ISAM (Zugriffsmethode) 317
ISAM-Pool 286, 293, 318
ITAM-Schnittstelle 17
ITIAM-Schnittstelle 18

J

JIT-Puffer 29, 348
JIT390 29
JMU 200
Job 19
Job-Klasse 197
Job-Management 141, 197
Job-Scheduler 201
Job-Scheduling-Strategien 197
Job-Streams 201
JTABSMEM (Systemparameter) 348
JTMAXMEM (Systemparameter) 349
JTSHMEM (Systemparameter) 349
JTSTDMEM (Systemparameter) 348
Just-In-Time-Übersetzung 29, 348

K

Kanal 35
 Typ FC 33, 45, 108
Kanalperipherie 33
Katalogintensität 174
Kategoriesteuerung 205
KB 13
KDCFILE 193
Klasse-2-Systemparameter
 siehe Systemparameter 336
Klassen-Scheduler 202

L

L4MSG (Systemparameter) 343
L4SPDEF (Systemparameter) 345
LAN-Anschluss
 S-Server 88
 SQ-Server 92
Langzeitmessung mit SM2 274
Lastanalyse 208
Lasttyp 57
Lastverhalten 271
LEASY 319
Leistungserwartungen 20, 22
 für Online-Betrieb 24
Leistungswerte (Beispiele) 26
LLM-Objekte 325, 329
logische Datensicherung 98
LUN 48

M

Maschinenfehler-Unterbrechung 242
Maßeinheit 13
MB 13
MBK-Archivsystem
 Scalar 109
MEH (Funktionszustand) 239
Meldungsdateien 169
Meldungsgewicht 169
Messmonitor 272, 273
Micro Time Slice 260
Migration 39
Monoprozessor 27
MRS-Katalog 157
MSGCENTL (Systemparameter) 172, 343
MSGCENTN (Systemparameter) 172, 343
MSGDLAM (Systemparameter) 343
MSGFILii (Systemparameter) 169, 343
MSGMAKER 169
MSGNOFL (Systemparameter) 169, 343
Multiprozessor-Systeme 27
 CPU-Auslastung 28
 CPU-Zeit 27
 Durchsatz 28

N

Nearline-Peripherie 97
Net-Storage 86
Netzanalyse 188
Netzlaufzeit 17, 188, 190
Netzqualität 188
NFS 86
Nucleus Device Management 236

O

OLTP-Last 25
OLTP-Transaktion 188
Online-Anwendung 16
Online-Peripherie 97
openSM2, siehe auch SM2
openUTM 188, 191
Overrun-Intervall 191, 195

P

Paging Area 162, 279
Paging Load Control Function 254
Paging-Rate 299
Parallel Access Volume 55, 266, 270, 295, 298
Parallelität 99
Parameterdatei SYSPAR.BS2.version 336
Parametersatz SYSOPT-CLASS2 336
PCI-Controller 92
PCS 137
PCSDEFINE (Dienstprogramm) 226
Performance Control System 222, 276
Performant File Access 184
PFA-Caching 186
physikalische Datensicherung 98
PLAM-Bibliotheken 325
Platten, Betriebsarten 145
Plattenspeichersystem
 DMX 45
 DMX-3 70, 83
 DMX-4 70, 83
 Eigenschaften 45
 ETERNUS DX 45
 ETERNUS DX8000 59
 Komponenten 45
 Konfiguration 48

Plattenspeichersystem

- Replikation 50
 - S-Server 59
 - SQ-Server 79
 - VMAX 45, 70, 83
- Plattenspiegelung 145, 149
- Preemption 251
- Preemption Control Function 253
- PRIOR 203, 276, 300
- Lastanalyse 208
 - Scheduling-Algorithmus 235
- Priorität 245
- Prioritäten-Steuerung 203
- Prioritätenvergabe 219
- Private Volume 145, 148, 236
- Private-Slices 325
- Problemanalyse 271
- Programm-Unterbrechung 241
- Public Volume 145, 146
- Public-Slices 325
- Pubset 86, 146

R

- RAID-Gruppe 48
- RAID-Level 48
- Rate Adaption 111
- Read Ahead 46
- Read-Cache 177
- Read-Hit 46
- Read-Miss 46, 47
- Read-Write-Cache 179
- Readme-Datei 11
- REC (Remote Equivalent Copy) 50
- Referenzmaschine C40-F 26
- Rekonfiguration 236
- Relativer Performance Faktor 25
- Remote System Call 56
- Remote-Cache 50
- Reorganisation 175
- Replikation 50, 145, 149
- Reports 274
- residenter Hauptspeicher 144

S

- SAM (Zugriffsmethode) 317
- SCA (Speed Catalog Access) 338
- Scalar 109
- Scheduling
- festе CPU-Zuordnung 120
 - Zeitscheibe 118
- Scheduling-Algorithmus (PRIOR) 235
- Schwellwert 192
- Seitenwechselbereich 162
- Server-Wechsel 40
- SF-Pubset 145, 146, 152
- Shared Code 328
- Shared-Private-Disk 322
- Shared-Private-Volume 237
- Shared-Pubset 147, 322
- Sicherungsvolumen 98
- SIH (Funktionszustand) 239, 291
- Single Recording by Volume 148
- SM-Pubset 145, 147, 153
- SM2
- COSMOS 313
 - Engpass-Analyse 282
 - Leistungsbedarf 312
 - Messmonitor 272, 273
 - Messprogramme 17, 222
 - Reports 274
 - Ressourcen-Bedarf 312
- SnapOPC+ 50
- SNMP 188
- SOCKETS-Schnittstelle 18
- Software-Bedienzeit 268
- Space Control 167
- SPACEOPT 175
- Speed Catalog Access 150
- SRDF (Symmetrix Remote Data Facility) 50
- SSMAPRI (Systemparameter) 346
- SSMASEC (Systemparameter) 346
- Stand-by-Pubset 152
- Standard-Benutzerkennung 339
- Standard-Job-Scheduler 201
- Standard-Katalogkennung 339
- Standard-Volume-Label 236
- Storage-Klasse 153, 154

- Storage-Services 153
 - Streaming-Grenze 111
 - Stromausfall 46
 - Subsysteme 142
 - SVC 241, 286, 301
 - Symmetrix 145, 149, 175, 288
 - Host-Komponente 50
 - Symmetrix Remote Data Facility 50
 - SYSEAM 167, 344
 - SYSOPT-CLASS2 (Parametersatz) 336
 - SYSPAR.BS2.version (Parameterdatei) 336
 - System Managed Storage 147
 - System Service Slice 254
 - System Working Set 256
 - System-Job-Scheduler 201
 - System-Task 204, 243
 - Systemadressraum 143
 - Systemdateien 156
 - Systemkern 141
 - Systemmeldungen 170
 - Systemparameter
 - BMTNUM 157, 160, 337
 - CATBUFR 160, 337
 - CONSDDE7 338
 - DEFLUID 339
 - DESTLEV 339
 - DMMAXSC 340
 - DMPRALL 340
 - DMSCALL 340
 - EACTETYP 170, 343
 - EAMMEM 168, 344
 - EAMMIN 168, 344
 - EAMSEC 344
 - EAMSIZ 344
 - ETMFXLOW 345
 - JTABSMEM 348
 - JTMAXMEM 349
 - JTSHMEM 349
 - JTSTDMEM 348
 - L4MSG 343
 - L4SPDEF 345
 - MSGCENTL 172, 343
 - MSGCENTN 172, 343
 - MSGDLAM 343
 - MSGFILii 169, 343
 - MSGNOFL 169, 343
 - SSMAPRI 346
 - SSMASEC 346
 - TEMPFILE 347
 - Systemsoftware 15
 - Systemüberwachung 274
 - Systemumgebung 141, 142
- ## T
- TAC-Klassen 194
 - TANGRAM 233, 258
 - Task 243, 244, 286
 - Preemption 251
 - Warteschlangen 203, 261
 - Task In Control 261
 - Task-Kategorien 204
 - Task-Management 141, 203
 - Task-Prioritäten 199, 205
 - Taskgruppen 234
 - TCP-Transportservice 18
 - TEMPFILE (Systemparameter) 347
 - temporäre Dateien 347
 - Terminal
 - Anzahl (K) 20
 - Auslastung (U) 21
 - Wirkungsgrad (E) 20
 - Thin Provisioning 51
 - TimeFinder 50
 - TP-Anwendung 199
 - TP-Betrieb 16, 25, 164, 188, 331
 - TPR (Funktionszustand) 239
 - Transaktion 16
 - Transaktionsbetrieb, siehe TP-Betrieb
 - Transaktionsrate 16
 - Transaktionszeit 16, 17, 20, 23
 - Trend-Abschätzung 271
 - TSOSCAT 146, 157, 337
 - TU (Funktionszustand) 239

U

Übertragungsrate 13
UNIX-Dateisystem 86
Unterbrechung 239
UPAM (Zugriffsmethode) 315
User Shared Memory 328
User-PFA-Caching 176, 184
User-Task 243

V

Verweilzeit 19
virtuelle Maschine (VM) 117
VM2000 117, 276, 285
 \$VMCONS 138
 Anzahl Gastsysteme 129
 CPU-Leistung 118
 CPU-Pool 122
 CPU-Quote 119, 130
 Hauptspeicher 124
 Hypervisor 117
 IO-PRIORITY 127
 Overhead 117
 PCS 137
 Peripherie 125
 Scheduling 118
 Systembetreuung 135
 VM-Gruppe 121
Vollsicherung 98
Volume, logisch 52
Volume-Set 153, 155, 161

W

Waiting Time 255
Warteschlangen 203
Wirkungsgrad für Terminal (E) 20
Working-Set-Bedarf 164, 211
Write-Cache 178
Write-Hit 46

X

XCS-Verbund 198
Xen-Hypervisor 117

Z

Zeitdefinitionen
 bei Batch-Auftrag 19
 bei Online-Anwendung 17
Zeitgeber-Unterbrechung 241
Zugriffsmethode 315
 BTAM 315
 Datenbanken 319
 DIV 320
 FASTPAM 316
 HASH 318
 ISAM 317
 SAM 317
 UPAM 315
Zugriffssysteme 145

