English

BS2000/OSD

BS2000/OSD

Performance Handbook

User Guide

Valid for:

BS2000/OSD-BC V9.0
OSD/XC V9.0
X2000 V5.3

Edition March 2013

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:
manuals@ts.fujitsu.com

Certified documentation according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2008.

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

Copyright and Trademarks

Contents

1 Preface

The aim of every IT department is to satisfy the requirements of the system users while at the same time keeping costs to a minimum.

The following aspects make the adjustment of the system environment (hardware and software) a dynamic process:

- The steady increase in real-time processing with all its options of directly handling customer service and office work.
- The constant increase in applications and functions (data networks, distributed processing).
- The interrelationships between these functions (because of their mutual dependence).
- The growth of existing applications as a result of increased business volume.
- The load fluctuations in daily business.

The following steps should be taken to ensure that the system is used economically:

1. Determine precise performance expectations.
2. Define the configuration.
3. After taking up production, **carry out monitoring runs** to determine whether the performance expectations are being met.
4. If not, concentrate on those bottlenecks whose removal promises the greatest improvement in performance.
5. Once the bottlenecks have been discovered and removed, **repeat the monitoring runs**, as this often reveals other bottlenecks which were previously concealed.
6. Even after performance expectations have been met, it is essential to carry out **periodical monitoring runs**, for only in this way is it possible to detect tell-tale signs of saturation in the main resources due to the increasing workload and to avoid critical situations in the system.

1.1 Objectives and target groups of this manual

The “Performance Handbook” is intended to assist the system user in assessing and improving the performance level of his/her data processing system. It is aimed in particular at personnel in data centers and the system support.
Information for the fine-tuning of both configuration and software make it easier to use BS2000/OSD more cost-effectively.

This Performance Handbook is a component part of the basic documentation for BS2000/OSD-BC and thus also for OSD/XC. It provides a straightforward summary of the performance aspects of IT operation for BS2000/OSD servers.

1.2 Summary of contents

This manual describes the potential for the fine adjustment of the system configuration and software for ensuring that the required performance is provided and for optimizing the use of resources. Where necessary a distinction is made between the different BS2000/OSD servers.

This manual does **not** provide an introduction to business server operation. Such topics are the subject of the basic documentation on the hardware, the operating system, and the system support software in the current version.

The chapter “Performance expectations from the user viewpoint” describes the most important criteria for assessing the performance of an IT system and what they mean.

The chapters „Performance behavior of the servers“ and „Performance behavior of the peripherals“ describe the significance of the Relative Performance Factor (RPF) for the CPU performance and the performance behavior of the input/output peripherals. Using standard values, the chapters explain how to design the input/output configuration.

The chapters “Data backup”, “Performance aspects in VM2000 operation” and “Performance behavior of the software” describe special features related to performance which must be taken into account when using the BS2000 software products.

The chapters „System control“ and „Internal procedures“ describe the influence system parameters on the system throughput and internal procedures in BS2000/OSD which facilitate the understanding and interpretation of measurement figures.

The chapter “Use of the openSM2 monitoring system” contains recommendations for parameter settings for routine monitoring and bottleneck analysis, explains characteristic reports and describes the basic procedure for analyzing system-oriented and user-oriented performance problems.

The chapter “Optimizing performance when developing application software” contains recommendations relating to the selection of the file access method and structure, as well as notes on improving performance when loading programs and products.

At the end of the manual you will find an appendix and a reference section that will make it easier for you to work with the manual.

Readme file

The functional changes to the current product version and revisions to this manual are described in the product-specific Readme file.

Readme files are available to you online in addition to the product manuals under the various products at *http://manuals.ts.fujitsu.com*. You will also find the Readme files on the Softbook DVD.

Information under BS2000/OSD

When a Readme file exists for a product version, you will find the following file on the BS2000 system:

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

This file contains brief information on the Readme file in English or German (<lang>=E/D). You can view this information on screen using the `/SHOW-FILE` command or an editor. The `/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>` command shows the user ID under which the product’s files are stored.

Additional product information

Current information, version and hardware dependencies, and instructions for installing and using a product version are contained in the associated Release Notice. These Release Notices are available online at *http://manuals.ts.fujitsu.com*.

1.3 Changes since the last edition of the manual

The following major changes have been made since the last edition of this manual:

- The manual has been updated to cover BS2000/OSD V9.0.
- New business servers: S175, S210, SQ200 and SQ210
- New disk storage systems: ETERNUS DX
- New tape drive: LTO-5.
- SX servers are no longer supported by BS2000/OSD V9.0 and higher. Consequently their description has been removed from this manual. Exception: Information on migration from SX to SQ servers.
- The Symmetrix 8000 disk storage system is obsolete and is no longer described in this manual.
- Connection via a type S channel is no longer supported by the new tape and disk storage systems. The type S channel is therefore no longer included in performance measurements. It is no longer mentioned in comparisons in this manual.

1.4 Notational conventions

On account of the frequency with which the names are used in this guide, the following abbreviations have been adopted for the sake of simplicity and clarity:

- **S server** for the S series business servers
- **SQ server** for the SQ series business servers
- **BS2000** for the operating system BS2000/OSD
- **SM2** for the openSM2 measurement system, provided **no** distinction need be made between the openSM2 measurement system and the SM2 monitor.

The following conventions are used in this Migration Guide:

to indicate particularly important information

[ ] References to related documents are provided in abbreviated form in the text. The full title of each publication referenced by a number is provided in full after the corresponding number in the “Related publications” section.

Commands to which reference is made in this manual are described in the “Commands” manual [15] and the macros mentioned are described in the “DMS Macros” manual [8]. The metasyntax of the commands and macros is given in the corresponding manuals.

Note on the units of measurement used

This manual uses both German and English notation. However, for throughput and transfer rates the notation Kbytes, Mbytes or Gbytes is always used. The following definitions apply:

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 Performance expectations from the user viewpoint

2.1 Characteristic values for describing the performance of an IT system

Every IT system can be divided hierarchically into the following levels:

- application programs
- system software
- Hardware

The user's requirements (as represented by the application programs) are transformed into tasks at the system software level, and into instructions and I/O operations at the hardware level.

The system software no longer views the application programs as its workload, but rather the number of concurrent tasks and the differing demands they make.

At hardware level the workload appears as a set of commands, input/output operations and storage reservations. At user level many different workloads can result in the same hardware workload.

For the user, the following questions are paramount when judging the performance of an IT system:

- How long does it take the IT system to process the user's requests?
- How many requests per unit of time can the IT system handle?

The performance capacity on the hardware level (e.g. number of instructions per second) or on the system software level (e.g. efficiency of task management in the form of job queue transfers per second) is only of secondary importance to the user. The behavior of the application software affects this performance capacity. Therefore these components must also be taken into consideration.

Depending on the application type (online application, batch processing), the following characteristic values are used **to describe performance at the user level**.

2.1.1 Online application

In both transaction and interactive mode, the unit of IT work is the **transaction**. User requests in the form of inputs are represented in the system by means of transactions.

In **TP mode** (also known as transaction mode or inquiry and transaction mode) the terminal user can only communicate with programs that have been predefined by the application. Normally, a large number of data display terminal users work with a relatively small number of **application programs**.

In **interactive mode** (also known as dialog mode or timesharing mode), each terminal user formulates his/her own application, which is then used to solve the particular problem in dialog. Programs used to control the individual dialog applications are normally **system programs** for creating, testing and modifying files or programs.

The time span between the user's input and the end message from the system is referred to as the **transaction time**. It may be made up of a number of individual responses, each of which has its own response time.

Characteristic values in an online application

- Transaction rate:
  Sum total of successfully terminated transactions per unit of time.
- Response time:
  Time taken by **one** processing operation by the server.
- Number of terminals
  (equivalent to the number of active terminal users).
- Efficiency per terminal.

Diagram of time definitions

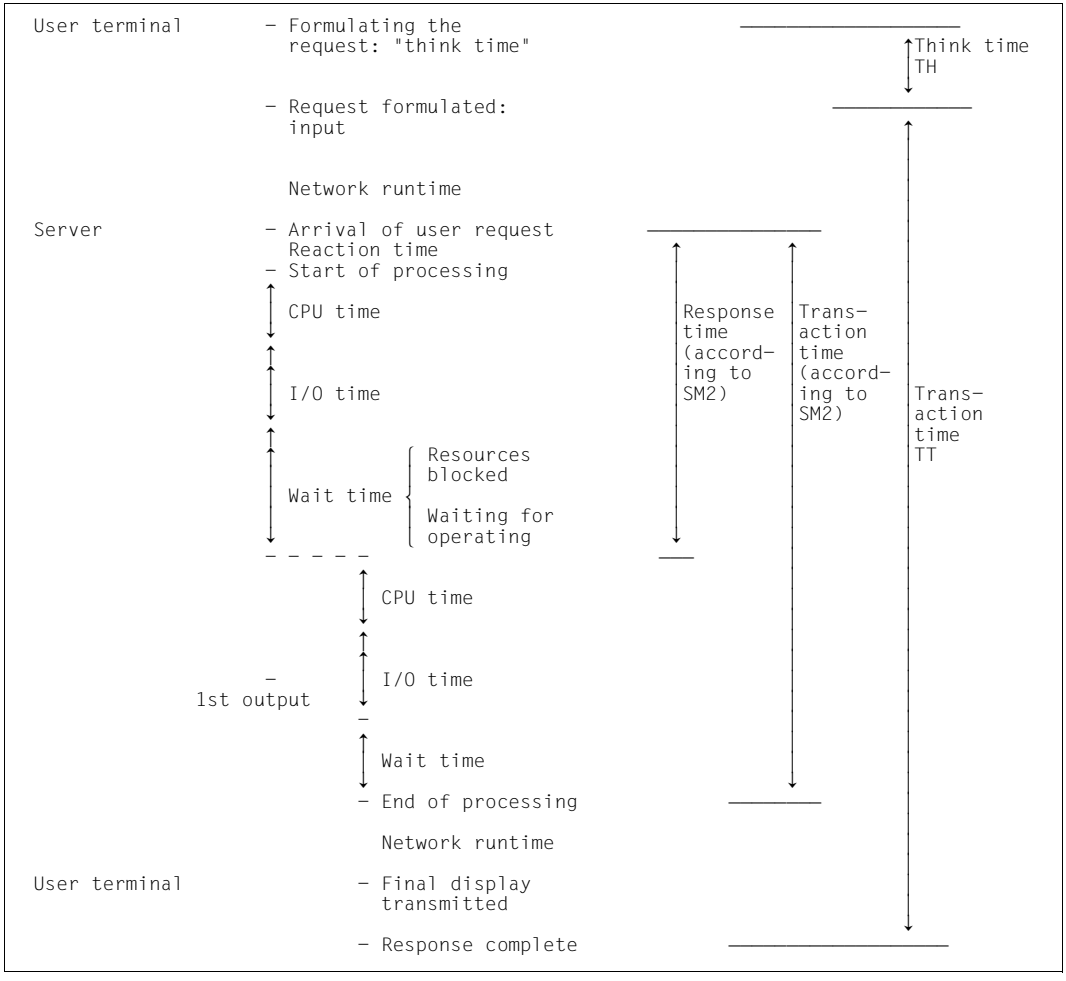


Figure 1: Time definitions in an online application

If there is only **one** response per transaction (which is usually the case in TP mode), the response time is identical with the transaction time.

The time definitions shown (response time, transaction time, think time) and the transaction rate can be recorded by the SM2 monitoring program BCAM-CONNECTION, but only for applications that work with the BCAM transport system via the IDCAM or ITIAM program interface.

For applications that work with the BCAM transport system via the SOCKETS program interface, the SM2 monitoring program BCAM-CONNECTION returns connection-specific values via PORT numbers:

- INPROC:
  Time from the receipt of a message by BCAM until it is fetched by the application. The INPROC time comprises the INWAIT time (i.e. the time between BCAM indicating that the message has arrived and the application fetching the message).
- REACT:
  Time between the send call and the immediately preceding receive call of an application. In TP mode this time can be regarded as the response time.
- OUTPROC:
  Time from the send call until the message's last byte is transferred to the network.

The TCP transport service works on a stream-oriented rather than a message-oriented basis. It is therefore not possible to obtain the transfer time through BCAM for transactions with several responses per input. Nor is the think time recorded.

If the input message length is short (< 500 bytes), the number of TSDUs received (Transport Service Data Units; jobs to BCAM) per second can be interpreted as a measurement of the transaction rate.

The measured values INPROC, INWAIT, REACT and OUTPROC are also obtained for applications that work with the BCAM transport system via the ICMX, IDCAM and ITIAM program interfaces.

2.1.2 Batch processing

With batch processing, the unit of IT work is the **job**.

Diagram of time definitions

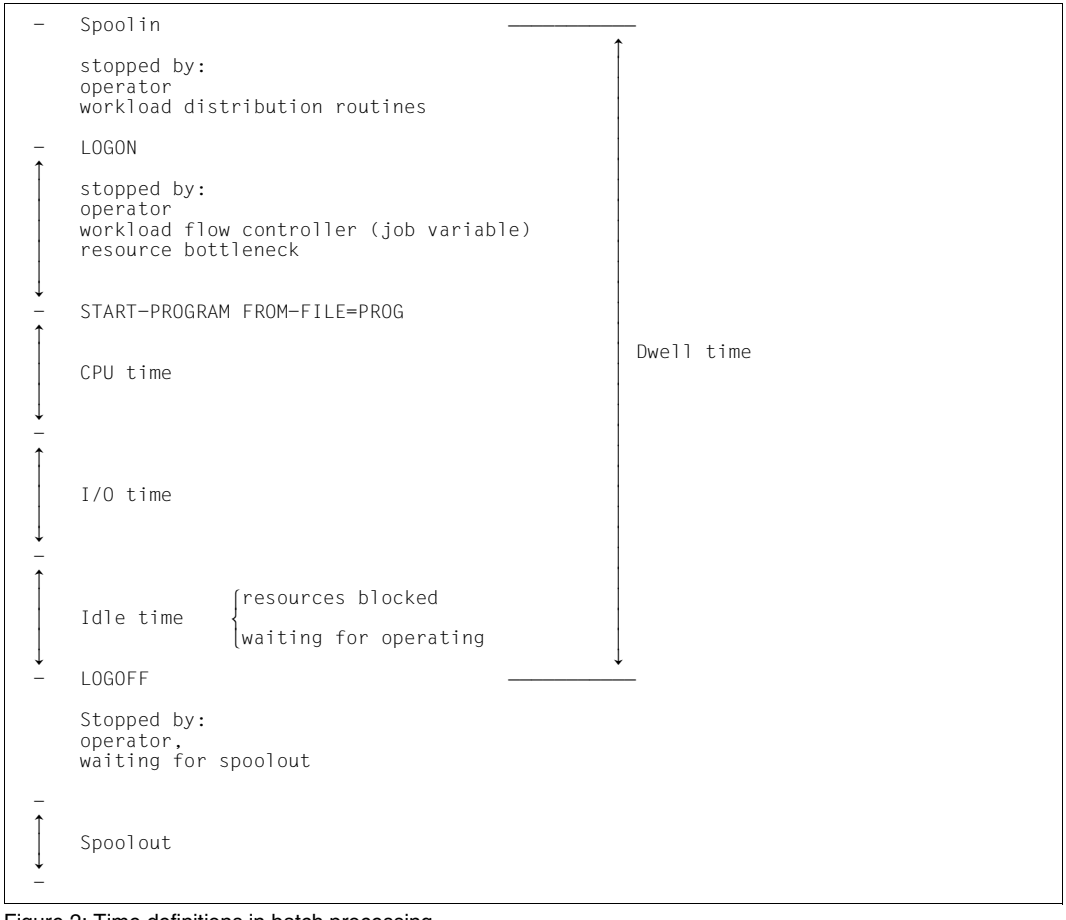


Figure 2: Time definitions in batch processing

Characteristic values in batch processing

- Throughput rate:
  Sum total of successfully executed jobs per unit of time
- Dwell time:
  Time required to process one job

2.2 Formulating a performance expectation

The following fundamental relation holds between the transaction time, the transaction rate and the number of terminals:

```
(Think time + Transaction time) * Transaction time = Number of terminals
```

The term “transaction time” summarizes the system's reaction to a wide variety of requirements. For this reason it is difficult to decide whether a transaction time is appropriate or not. Due to the costs involved in operating the terminal and the user's working hours, the following method is recommended:

The terminal cannot be used during the transaction time as it is waiting for the server. During the think time the terminal and user interwork by reading the output or typing in a new entry. The proportion of think times in the total busy time of the terminal is the efficiency value E for terminals. If the average value for all terminals drops below 75%, this is an indication of performance problems. Of course, pauses for a “breather” must not be taken into account when calculating think times.

The following values are used:

TH	(Think Time)	the think time
TT	(Transaction Time)	the transaction time
TR	(Transaction Rate)	the transaction rate

According to the definition, the following is true for the **efficiency value (E) for terminals**:

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

Let the number of transactions to be performed per second for an application be TR. Assuming the number of terminals required for this application is K, then

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

```
K = TH / E * TR
```

If an efficiency value of 0.75 is assumed for terminals, the formula is:

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

The table below contains the values for 1/E for various efficiency values.

E [%]	70	75	80	85	90	95
1/E	1.43	1.33	1.25	1.18	1.11	1.05

The formula E = TH / (TH + TT) shows that relatively long transaction times (or response times in the case of only **one** response per transaction) can be accepted in the event of long think times, while in the event of short think times the transaction times have a lasting effect on the efficiency value and hence on the number of terminals required. As the network runtime is included in the transaction time, terminals with short think times and extensive outputs should be connected via high-speed lines.

The number of terminals required is determined by the utilization (U) of the terminal as well as by the efficiency value. The utilization is the proportion of the total time spent by the machine either waiting for a response (transaction time) or preparing a new input (think time). Time taken for a “breather”, for instance, reduces the utilization without the device really being free. In the case of a data entry application, documents are generally processed from a stack. Workstations can therefore be utilized to a very high level.
It is possible to achieve terminal utilization of 90% under these conditions. A slight loss of 10% is allowed for to accommodate “breathers”. Utilization must be set lower for other applications. In the case of TP mode with customer traffic, it is desirable to eliminate inconvenient waiting times for the customers. These occur when there is no terminal free and hence the customer cannot be served immediately. The situation is similar in the case of program development. In cases such as this, terminal utilization (U) should not exceed 60%.

The **number of terminals (K)** required can then be calculated by:

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

Example

The following two applications exist:

The first application is a data entry application with a think time of 20 seconds and a transaction time of 1 second. On average 90% of all data entry stations are always in use. 7,200 documents must be processed per hour.

The second application is a seat booking application and involves dealing with the public. Think time is 120 seconds and transaction time is 3 seconds. In order to avoid noticeable delays for customers who wish to book, terminals are only utilized to 60% of their capacity. 360 bookings per hour must be processed.

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

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

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

$K_1 = (20 * 2) / (0.95 * 0.90) = 47$ terminals

$K_2 = (120 * 0.1) / (0.97 * 0.60) = 21$ terminals

A total of 68 terminals are required for both applications. You must ensure that the values for the transaction time contained in the efficiency value for terminals are feasible for the server.
This constitutes the performance expectation which needs to be calculated.

It can only be met if the resources required by the application do not exceed the limits of the server and the server has a certain amount of reserve capacity. Here the load on those resources which are heavily used by the application concerned plays a particularly important part. If the application involves a large number of I/O operations, then you must ensure that utilization of the disk access systems is kept at a low level. The situation is not improved by additional spare capacity in the server.
The opposite is true of applications which require a large number of calculations. Paging is unhelpful in both cases.

There is a natural lower limit for all transaction times. This is reached when a terminal performs its transaction using a server which is otherwise completely free of other operations and using data transmission paths which are equally free of other operations. It is, of course, not cost-effective to use an IT system like this. If other terminals and, where appropriate, other applications are added, the time required for execution is extended due to mutual obstruction by the transactions.

A **dilation factor (D)** is defined:

```
D = TT (current workload) / TT (empty server)
```

This dilation factor is the price is to be paid for the economic utilization of the system. To what extent a particular dilation factor is acceptable depends in turn on the think time and the optimal transaction time TT (unoccupied server). Dilation factors of 10 are still acceptable in the case of short transaction times and long think times. If, in the case of a think time of only 10 seconds, the optimal transaction time is even as much as 3 seconds, then a dilation factor of three is already too large. The **optimal transaction time** can either be measured on a server otherwise free of other operations or be estimated using the following formula:

```
TT (empty server) = CPU + n * IO * NL
```

where:

CPU	CPU time required for the transaction
IO	mean time for an I/O operation
n	number of I/Os (including paging) required for the transaction
NL	network runtime for the input **and** output messages

All these values can be altered by means of modification to the hardware or the software. In the case of the CPU and I/O times, several tasks generally have to be taken into account, in transaction mode, for example, UTM and DBH tasks.

The dilation factor measured by SM2 does not take the network runtime into account. Only the internal server data is processed.

Example 1

Database query with the following values:

```
CPU          0.1s
IO         0.004s
NL           0.1s
n              60
```

```
TT (empty server) = 0.1s + 60 * 0.004s + 0.1s = 0.44s
```

Using a dilation factor of 3 for the server and the network results in a $TT_{current}$ of $3 * 0.44 = 1.32 \approx 1.3$. Using a think time of 40 seconds, the efficiency value for terminals can be calculated at 97%:

$E = TH / (TH + TT_{current}) = 40s / (40s + 1.3s) = 0.97$

If the think time is only 20 seconds, the efficiency value for terminals drops to 94%. In this case you should check whether the I/O times could be reduced, for instance by reducing the number of inputs/outputs or quicker peripherals.

Example 2

Scrolling in the editor with the following values:

```
CPU        0.01s
IO        0.004s
NL           1s
n             2

TT (empty server) = 0.1s + 2 * 0.004s + 1s = 1.02s
```

The important factor here is the transmission time for the output screen. Even if the disks on the server are overloaded (e.g. 0.025 s per I/O operation), the transaction time is only marginally increased to 1.06 s. A think time of 3 seconds results in an efficiency value for terminals of 75%. This is unsatisfactory.

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

A doubling of the line speed in this case results in an optimal transaction time of 0.53 seconds. If it is assumed that no dilations occur during the transmission of messages, then this transaction time remains valid even when the server has a realistic workload. In this case the efficiency value for terminals rises to 85%.

These two examples show how varied the causes for inadequate performance can be. Overloading can occur in the CPU or in the peripherals, as can combinations of bottlenecks. For help in locating and solving problems like this, see section “Performing a bottleneck analysis” on page 282 and chapter “Optimizing performance when developing application software” on page 315.

The formulas and examples above demonstrate the way in which users should formulate their performance expectations for online operation. Their intention is always to minimize costs. If savings (e.g. in staff and terminals) on the one hand are set off against increased costs (e.g for faster lines, faster processors, faster disks or program alterations) on the other, then one always arrives at a sound performance expectation. Any decisions reached should always be examined at regular intervals in the light of fluctuations in the price of hardware.

Of course, subjective considerations such as the psychologically damaging effect of occasional long transaction times must not be ignored completely. However, performance expectations based upon these considerations are misleading. Eliminating individual long transaction times is generally a very complicated operation and does not normally lead to overall savings.

3 Performance behavior of the servers

A table containing the performance data and the standard values of the various server models is provided in the section “Standard values for BS2000/OSD servers” on page 331.

3.1 CPU

3.1.1 Relative Performance Factor (RPF)

In the BS2000 world the performance of a server is expressed in terms of its RPF (Relative Performance Factor). The RPF value indicates the performance capability of the CPUs under a realistic load.

The RPF is determined by measuring with two benchmark loads. An OLTP load (TP benchmark with UTM and SESAM/SQL applications) and a batch load (SKBB benchmark with COBOL programs).

A performance value is obtained for batch mode (RPF_{SKBB} using jobs per CPU second as the unit of measurement) and TP mode (RPF_{TP} using transactions per CPU second as the unit of measurement).

The result for TP mode accounts for 75% and the result for batch mode for 25% of the overall RPF value.

Here are some examples of the calculation of the RPF value (the measured values are compared with the values measured for the reference machine C40-F (RPF=1)):

Model	**Number of CPUs**	**RPF_{SKBB}** (RPF value with SKBB benchmark)	**RPF_{TP}** (RPF value with TP benchmark)	**RPF_{total}** (weighted mean from RPF_{SKBB} and 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

Table 1: Examples for the RPF value

It is not ideal to express the performance behavior of a server in a single figure. As you can see from the table above, the performance depends to a certain extent on the application profile.

Due to the particular architecture of SQ servers and the load-dependent efficiency of the CISC firmware (see page 29), the performance of SX and SQ servers is more heavily dependent on the application profile, resulting in a wider performance bandwidth.

3.1.2 Multiprocessor systems

Multiprocessor systems consist of 2 to 16 central processing units and 1 to 8 I/O processor units. The advantages lie in increased availability and a corresponding improvement in performance.

Efficient use of multiprocessor systems requires multiprocessor-capable applications. This means that the application must be able to process the requirements not only in parallel, but also without any significant mutual impairment.

Increased availability

Given appropriate hardware redundancy, whenever one of the hardware components breaks down, the system performs automatic reconfiguration without interrupting the current session.

Improved performance

In multiprocessor mode, i.e. under the control of **one** BS2000 operating system, a number of tasks can be processed in parallel (depending on the number of CPUs). To ensure that this increased performance capacity can be utilized, there must always be enough tasks available for parallel processing.

It should be remembered that each task can only make use of the capacity of **one** CPU. This means that improvements in response time behavior are limited, the only possibility being the reduction by the other CPUs of the wait times for CPU allocation. The same applies to batch applications: the runtime of a batch application can only be significantly reduced by parallelizing the pure computing requirement.

Note that the amount of CPU time used for a task is (with the same work) generally somewhat higher than on the corresponding uniprocessor. The higher the number of CPUs, the more sharply the overhead for the main memory synchronization rises, combined with a reduction in the cache hit rate.

The CPU time will increase as follows:

– with 2 CPUs by 5 to 10%
– with 4 CPUs by 10 to 15%
– with 6 CPUs by 15 to 20%
– with 8 CPUs by 20 to 30%
– with 10 CPUs by 25 to 35%
– with 12 CPUs by 30 to 40%
– with 16 CPUs by 35 to 45%

Provided the load is distributed over tasks which run in parallel and are, if possible, independent of each other, the transaction rate can be considerably increased. The following improvements in the throughput rate can be achieved on S servers in comparison to a uniprocessor:

– with 2 CPUs by a factor of 1.7 to 1.9
– with 4 CPUs by a factor of 3.3 to 3.6
– with 6 CPUs by a factor of 4.6 to 5.2
– with 8 CPUs by a factor of 5.8 to 6.6
– with 10 CPUs by a factor of 6.8 to 7.8
– with 12 CPUs by a factor of 7.8 to 8.8
– with 16 CPUs by a factor of 8.8 to 9.9

In order to satisfy the increased resource requirements caused by raising the number of tasks, extension of the main memory and an increase in the number of disk drives are absolutely essential.

Recommended maximum CPU workload

In the case of response-time-critical applications, with uniprocessors a CPU workload of 70% should not be exceeded for the main application, because as the workload increases, the wait time in the queue ahead of the CPU rises disproportionately (see also section “Investigating system-oriented performance problems” on page 291).

With multiprocessor systems the probability of finding a free CPU increases as more CPUs become available. Therefore higher workloads can also be tolerated. With response-time-critical applications the following CPU workloads should not be exceeded for the main application:

– 75% with 2 CPUs
– 80% with 4 CPUs
– 85% with 6 CPUs
– 90% with 8 or more CPUs

3.1.3 CPU performance of SQ servers

Basic information on the SQ servers is provided in the manual ”Operation and Administration” [4]. The sections below deal with the SQ server components which have an effect on the CPU performance of the SQ servers.

CPUs

The CPUs available are split logically into CPUs for applications under BS2000/OSD (BS2000 CPUs) and CPUs on which BS2000 I/Os and administration tasks for the SQ server are handled by X2000 (I/O processors).

Depending on the model, up to 16 CPUs (SQ210-160F) can be used as BS2000 CPUs. 25% of the available CPUs are used as I/O processors.

A special X2000 firmware layer supports execution of the BS2000/OSD operating system and the compatible execution of /390 applications on the BS2000 CPUs, and also implementation of input/output operations on the I/O processors.

CISC firmware, JIT390

The CISC firmware (CISCFW) is the firmware component for mapping nonprivileged /390 code to x86-64 code. It complements x86-64 mode and allows existing /390 code to be run on an object-compatible basis (synonym in /390 mode: compatibility mode) on x86-64 hardware.

As of SQ200, the CISC firmware contains the component JIT390, a Just-In-Time /390 code compiler which converts the /390 code to x86-64 code at execution time. A code block is compiled only when it is executed and stored in a local task JIT buffer. When the code block is executed again the code which has already been compiled and optimized is executed directly from the JIT buffer. Reusing predefined and optimized code sections is considerably faster than renewed interpretation and at the same time places less of a load on the processor. The JIT buffer is created as resident memory for the purpose of further optimization.

Existing (nonprivileged) customer applications still run on an object-compatible basis in /390 code using the CISC firmware. This applies both for applications which were generated with ASSEMBH and for programs generated with the BS2000 compilers for higher-level programming languages. SQ servers also support ESA data spaces in customer applications.

The effective performance of an SQ server depends on the number of system calls which occur within a /390 application and on how greatly the /390 application profits from the efficiency of the CISC firmware.

System calls are processed in the server's TPR or SIH processor state (x86-64 mode). The applications run in the TU processor state (/390 mode).

The efficiency of the CISC firmware thus depends on how the JIT buffer is used by the load concerned: on the one hand on the repetition rate of specific /390 code parts (low CPU requirement), and on the other hand on the size of the program parts of a /390 application translated into x86-64 code (higher load on main memory and main memory cache).

CPU performance

The nominal RPF values (see “Standard values for SQ servers” on page 334) do not always allow reliable conclusions to be drawn with regard to runtimes, transaction rates or CPU time consumption. Depending on the application, these values can be higher or lower.

The RPF values measured for an SQ server apply for applications with a TU percentage of 40 - 50%. In the case of applications with a TU percentage outside this range, the performance of an application can deviate from the nominal RPF values.

See also the section “Migration from S servers to SQ servers” on page 41.

3.2 CPU-oriented memory

3.2.1 Main memory

In particular for TP operation, a paging rate that is too high is critical (see the maximum recommended number of paging I/O operations in the section “Standard values for BS2000/OSD servers” on page 331). Too high a paging rate always results in unsatisfactory response times. If the critical number of paging I/O operations is repeatedly reached or even exceed, the main memory must be expanded.

In VM2000 operation the main memory should be distributed as efficiently as possible over the guest systems before a general upgrade of the main memory configuration.

Monitoring main memory utilization

The utilization of the main memory should be continuously monitored by systems support using SM2 in order to detect an impending main memory bottleneck in good time.

The following key figures are of significance for calculating and evaluating the memory utilization (see also “Paging management algorithm (System working set method, SYS-WS)” on page 256):

- NPP (Number of Pageable Pages): Corresponds to the size of main memory which is available minus the number of resident pages.
- SWS (System Working Set): Number of globally used pages Corresponds to the NPP minus the number of freely usable pages in the so-called free pool.

The closer you are to a memory bottleneck or the critical paging rate, the more accurate is the result of the evaluation method recommended here.

The utilization level of main memory (in percent) is obtained using the following formula:

```
N = SWS * 100 / NPP %
```

Measurements for N up to 75% are regarded as uncritical.
When the values for N exceed 90% a main memory upgrade should be considered. This is particularly the case when the new version of a software product is to be introduced and this software is used by several tasks.

3.2.2 Main memory on SQ servers

The main memory is distributed between BS2000/OSD and domain 0 (Dom0) and X2000 (I/O processor).
A recommendation for the main memory configuration of SQ servers which is appropriate for most applications is shown in section “Standard values for SQ servers” on page 334.
The SQ server standard models offered are delivered with this recommended memory configuration.

In most cases the default settings for main memory management do not need to be changed. Expansion of the main memory needs to be considered in the following cases:

- Expansion of the entire main memory

  When multiple BS2000 or Linux/Windows guest systems are used, the size of the entire main memory must be selected to ensure that the total main memory requirement of all guest systems which operate in parallel does not exceed the total main memory which is available. The main memory volume for Dom0/X2000 (default: 30% of the total main memory) must also be taken into account here.

- Main memory size of BS2000/OSD

  For each VM with a BS2000 guest system at least 512 MB of main memory is required for the guest system and micro kernel together. The main memory required by the applications is not included in this figure. The total main memory required for a guest system depends largely on the applications and the performance required, Use of an extremely large number of or of extremely large tasks or processes always calls for a larger main memory configuration.

- JIT buffer

  The JIT buffer requirement for the JIT compiler must be borne in mind. By default, 40% of the BS2000/OSD main memory is reserved for this purpose. This memory is not directly available to the operating system and applications.
  Further information is provided in section “Buffer size of the CISC firmware” on page 348.

3.3 Input/Output

3.3.1 Input/Output on S servers

On the S servers, devices can be operated via channel connection technology (channel peripherals). Fibre channel peripherals (FC peripherals) can be connected via the type FC channel.

3.3.1.1 Type FC channel

Data transfer is serial via a fiber-optic cable with a maximum data rate of 100 Mbytes per second.

The essential features of the type FC channel are:

- full-duplex operation
- a maximum of **eight** I/O operations (of different control units) can be executed simultaneously
- effective data rate for large I/O blocks: more than 90 Mbytes per second possible
- connection of FC devices possible via FC switch (HNC and SKP can also be connected directly
- maximum distance of an FC switch: 550 m
- distances of up to 100 km are bridged without any major performance losses
- only disks with FBA formatting are supported (see section “Properties of external disk storage systems” on page 45).

The following disk storage systems and devices are supported via the type FC channel:

- ETERNUS DX
- Symmetrix DMX, VMAX
- ETERNUS CS HE:
  VTA (Virtual Tape Appliance), SBU (Smart Backup Unit), VTC (Virtual Tape Controller)
- MTC devices LTO-x in the MTC archive systems Scalar i500, i2000, i6000 and 10K
- High-Speed Net Connect HNC as of HNC-IV 91853
- Service/Console Processor SKP 3970-51 or higher

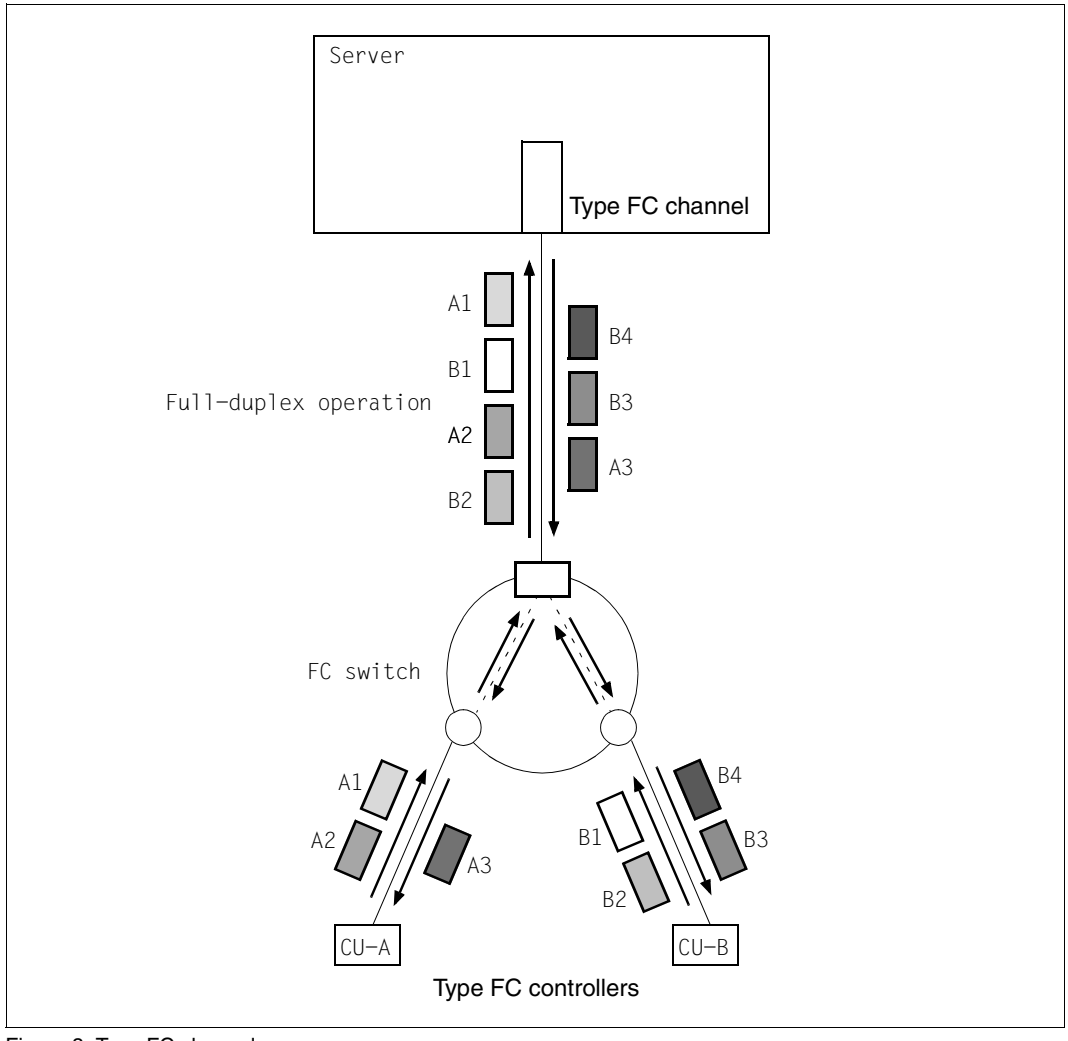


Figure 3: Type FC channel

The throughput on an S server displays the following characteristics:

- In the case of throughput-oriented loads (i.e. with large blocks of 32 KB or more), the throughput is limited by the performance capability of the FC connection to the disk storage system (1 Gbit/s).

  When only reading or writing takes place, a throughput of more than 90 MB/s is possible.

  In the case of access with a proportion of 25% write accesses, the duplex feature of the type FC channel can be utilized, which enables a throughput of 110 MB/s to be achieved.
- In OLTP mode (i.e. with small blocks of 2 - 4 KB), the throughput is limited either by the performance capability of the CPUs, of the channel or of the disk storage system.
- Type FC channel scale very well. In other words, n channels of the type FC permit an almost n-fold throughput. A prerequisite for this is that the BS2000 CPUs and the peripherals are sufficiently powerful.

Due to the concurrent execution of up to eight channel programs, the utilization of a type FC channel cannot be measured directly with SM2. The utilization is therefore determined indirectly by comparing the actual number of I/O operations with a maximum throughput value that could be achieved in laboratory measurements.

3.3.1.2 Type S channel

This channel type is only still supported in BS2000/OSD for reasons of compatibility. State-of-the-art tape storage systems can no longer be connected via type S channels. Type S channels are being replaced by type FC channels, which offer considerably better performance.

With transaction-oriented load cases some 4 type S channels can be replaced by one type FC channel. With throughput-oriented load cases up to 7 type S channels can be replaced by one type FC channel.

3.3.2 Input/output on SQ servers

Devices can be connected to SQ servers in different ways.

- via local SAS connections (e.g. ETERNUS JX40)
- via Fibre Channel (FC peripherals, e.g. ETERNUS DX)
- via Ethernet (e.g. OnetStor)

Input/output on FC peripherals is performed via I/O processors and the PCI controllers of the SQ server.

3.3.2.1 I/O processors (X2000)

All input/output operations to the peripherals are directed via the I/O processors (X2000). The relevant interfaces are supported in X2000.

The peripherals are connected via PCI controllers:

- SAS:
  Connection of the internal system disks (not released fur use under BS2000/OSD), the internal disk storage system ETERNUS JX40, the internal LTO drive (SQ200 and higher), and the internal MTC system ETERNUS LT40
- Fibre Channel:
  - ETERNUS DX
  - Symmetrix DMX, VMAX
  - ETERNUS CS HE: VTA (Virtual Tape Appliance), SBU (Smart Backup Unit), VTC (Virtual Tape Controller)
  - MTC devices LTO-x in the MTC archive systems Scalar Scalar i500, i2000, i6000 and 10K
  - MTC system ETERNUS LT40 with LTO-x drives.
- LAN controllers (Ethernet, Fast Ethernet, Gigabit Ethernet)

The performance of the I/O processors is very good and sufficient for all SQ servers:

- Disks:
  The following data was measured on an SQ210-80F which was connected to a disk storage system via 4 FC connections each operating at 8 Gbits. The disks within the disk storage system could be reached via 2 paths. Purely arithmetically this configuration would permit a maximum throughput of around 3,000 MB/s to be achieved.
  [The results measured on an SQ210-80E which was connected to a disk storage system via 4 FC connections each operating at 2 Gbits are also displayed in square brackets. Purely arithmetically this configuration would permit a maximum throughput of around 800 MB/s to be achieved.]

  – When only one task and small blocks (2 KB) are used, the I/O rate when reading is over 2,700 [2,500] I/Os per second, and when writing over 2,300 [1,700] I/Os per second.
  – When 32 tasks are processed in parallel and small blocks (2 KB and 4 KB) are used, the maximum I/O rate is over 48,000 [30,000] I/Os per second.
  – When only one task and large blocks (160 KB) are used, the throughput when reading is around 100 [85] MB/s, and when writing around 125 [80] MB/s.
  – When 32 tasks are processed in parallel, the maximum throughput - depending on the actions (reading, writing, mixed load) - with block sizes or 32 KB and higher is approx. 1,900 [740] MB/s. In view of the theoretical maximum throughput which can be achieved, this is a good result.

  The following always applies: As the block size increases, the I/O rate decreases and the throughput increases.

- Tapes:
  Import/output to tape is optimized in the I/O processor. The throughput to a powerful tape (e.g. on ETERNUS CS HE) is around 200 MB/s.

- Networking:
  The following data was measured on an SQ210-160F.
  [The data measured on an SQ200-80E is also displayed in square brackets.]

  – A maximum transaction rate of almost 88,500 [33,000] TA/s is achieved.
  – With parallel connections the 1-Gbit LAN on both servers, both when sending and when receiving, can be utilized fully (up to 120 MB/s in each direction).
  – The throughput and CPU requirement when sending depend on the message length (8, 16, 32, 64 KB). The longer the message, the higher the throughput and the lower the CPU requirement. When messages are received, this dependence does not exist; the network is always fully utilized.

  Sufficient CPU capacity is required to fully utilize high-performance networks. The transaction rate is limited only by the BS2000 CPUs.

3.3.2.2 SAS connection

As a Direct Attached Storage (DAS) the ETERNUS JX40 disk system expands the storage capacity of the SQ servers by up to 72 terabytes via an extremely high-speed SAS connection (6 Gbit/s).

On an SQ200-80E with ETERNUS JX40, a maximum transaction rate of 30,000 TA/s (with 2-KB blocks) and a maximum throughput of over 570 MB/s (with 32-KB blocks) and of over 900 MB/s (with 160-KB blocks) was achieved with 25% write accesses and a 100% read hit rate with 32 parallel tasks.

The performance of an I/O processor is sufficient even for large I/O loads on the largest SQ server.

When the optional ETERNUS JX40 disk storage system is used which incorporates a backup battery unit (BBU) for the write cache and is connected via SAS RAID, a BBU Relearn Cycle is performed every 30 days. The BBU Relearn Cycle takes up to 15 hours. It discharges and charges the BBU fully several times. During this time the write cache is disabled; the additional acceleration which it provides is then not available. In certain applications this can result in longer runtimes or response times. The BBU Relearn Cycle can be configured on a customer-specific basis by customer support so that, for example, the BBU Relearn Cycle is only executed at the weekend.

3.3.2.3 FC connection

On SQ servers, PCI controllers with a capacity of up to 8 Gbit/s can be used.

In the case of disk inputs/outputs via FC, the performance of an I/O processor on an SQ server is comparable with the measured values of the SAS connection.

The performance of an I/O processor is sufficient even for large I/O loads on the largest SQ server.

3.4 Migration

Irrespective of the source and target HSIs, there are some general aspects which should be borne in mind when migrating a server:

Number of BS2000 CPUs of a server

When the multiprocessor level of a server changes when migration takes place, this can also result in a change to the performance behavior of the software running on the server. In such a case check the relevant parameters and settings which control the performance behavior of the software.
In particular check the settings for OLTP mode, e.g. the number of UTM tasks used, the number of TAC classes, the number of tasks per TAC class, the number of DBHs, and the buffer sizes of the database system.
Adjust the parameters if necessary.

Detailed information on this is provided in chapter “Performance behavior of the software” on page 141 and in the manuals on openUTM, in the “SESAM/SQL Server Performance” manual [27], and in the ORACLE manuals.

Main memory size

When migration takes place, check whether the main memory needs to be increased to enhance the performance:

- to avoid unnecessary paging I/Os
- to expand the buffer for OLTP mode
- to expand the caches for the software product DAB

Operating mode: Native or under VM2000

Take note of the following points if, in the course of migration, native mode is to be switched to virtual mode under VM2000:

- Virtual mode causes an overhead which must be allowed for accordingly when calculating the performance of the new server. The amount of overhead depends on the configuration, see page 117.
- Take note of a few constraints when configuring VM2000 to avoid unnecessary overhead. See section “Optimum configuration of VM2000” on page 132.

Initial status

The initial status in ongoing operation is the basis for these considerations. For this purpose the most important application cases, such as OLTP mode or critical batch applications, must be monitored on the original server using openSM2. The results are then evaluated and which server will be most suitable as a target server is forecast.

The following must always be observed in the case of server migration:

- Current CPU workload of the original server (high / low)
  In OLTP mode the server's CPUs should have a workload of less than 70%.
  In batch mode a CPU workload of up to 100% is generally maintainable.
- Planning the extra CPU capacity required when changing a version of BS2000/OSD
  The extra capacity required depends on the version. On average 1 - 2% per version must be estimated.
- Taking into account the system's own growth requirements
  Example: 5% per year for 3 years, i.e. around 15% growth requirements.
- Influence of adding new applications or relocating existing ones
  If applications are omitted from the new server or new ones are added, this must be taken into account.

If required, it is recommended that the offering of the BS2000 Optimization Service and the BS2000 demo center should be used:

– Performance consulting when migrating

– Test measurements for customer applications on the target hardware

Please contact your sales representative for information.

3.4.1 Migration from S servers to SQ servers

The following must be observed:

- Mono performance
  When migration is to take place from an S server (uniprocessor) to an SQ server (multiprocessor), the total RPF value can remain unchanged, but the uniprocessor performance will be lower in most cases. This must, for example, be taken into account for batch applications which cannot be operated in parallel.
- CPU performance: TU portion
  As explained in section “CPU performance of SQ servers” on page 29, the CISC firmware ensures that customer applications execute unchanged on SQ servers. The RPF value of the SQ server applies for a TU portion of 40% to 50% of the entire CPU requirement. When the TU portion is 50% or higher, an additional performance requirement of approx. 10% should be provided for the new SQ server.
- Peripherals: no channels
  No channels are available on SQ servers. For performance reasons a conversion to SAS or Fibre Channel is advantageous.
  Data migration may be required, see the ”Migration Guide” [17].
- Peripherals: networking
  The SQ server’s integrated LAN controllers are available.
  An HNC is not required, nor can it be used.
- Memory requirements
  As a rule the standard main memory configuration of the SQ servers is sufficient.
  For the exceptions see section “Main memory on SQ servers” on page 32.
- Object format
  System exit routines must be cleaned up and recompiled, see the “Migration Guide” [17].

Example: calculating the RPF value for a target SQ server

The following assumptions apply for the S servers used to date:

- S155-10A (204 RPF)
- Load: OLTP mode (UTM/SESAM)
- CPU workload 80% (50% of that TU)
- BS2000/OSD V8.0
- Growth requirements15%

This results in the following calculation for the target SQ server:

204 RPF * 1.1 (higher performance requirement because of TU portion 50%) * 1.02 (migration from BS2000/OSD V8.0 to V9.0) = 229 RPF

In order to keep the workload to the 65% recommended for TP mode despite the growth:

229 RPF * 1.2 (target workload 65%) * 1.15 (growth requirements) = 316 RPF

In other words an SQ210-20F server with 320 RPF would be suitable.
The switch from a uniprocessor to a multiprocessor with two CPUs must be borne in mind here.

In rare cases application-dependent special influences can reduce the CPU performance, e.g.:

– Intensive use of decimal, floating point or EX commands
– Self-modifying code
– Mixed code and data on the same memory page
– Alignment violations
– Very frequent SVC calls

3.4.2 Migration from SX servers to SQ servers

The following must be observed:

- Mono performance
  When migration is to take place from an SX server (uniprocessor) to an SQ server, it will always be possible to provide the uniprocessor performance again.
- CPU performance: TU portion
  As explained in section “CPU performance of SQ servers” on page 29, the CISC firmware ensures that customer applications execute unchanged on SX and SQ servers. The RPF value of the SQ server applies for a TU portion of 40% to 50% of the entire CPU requirement.
- Peripherals: no channels
  No channels are available on SQ servers. For performance reasons a conversion to SAS or Fibre Channel is advantageous.
  Data migration is normally not required, see the ”Migration Guide” [17].
- Peripherals: networking
  The SQ server’s integrated LAN controllers are available.
  An HNC is not required, nor can it be used.
- Peripherals: LTO drive
  The LTO-n drive incorporated in the SQ server permits continued use of LTO-(n-1) volumes from the initial system. LTO-(n-2) volumes can only be read and should therefore be migrated as soon as possible to LTO-n volumes. Backup copies on LTO-(n-3) volumes can no longer be read with LTO-n drives. In this case a migration must be performed in the initial system.
- Memory requirements
  As a rule the standard main memory configuration of the SQ servers is sufficient.
  For the exceptions see section “Main memory on SQ servers” on page 32.
- Object format
  System exit routines must be cleaned up and recompiled, see the “Migration Guide” [17].

Example: calculating the RPF value for a target SQ server

The following assumptions apply for the SX servers used to date:

- SX160-10C (105 RPF)
- Load: OLTP mode (UTM/SESAM)
- CPU workload 80% (50% of that TU)
- BS2000/OSD V7.0
- Growth requirements15%

This results in the following calculation for the target SQ server:

105 RPF * 1.04 (migration from BS2000/OSD V7.0 to V9.0) * 1.15 (growth requirements) = 125 RPF

In other words an SQ210-10E server with 125 RPF would be suitable.

4 Performance behavior of the peripherals

4.1 Properties of external disk storage systems

This section describes the properties of the following external disk storage systems which are relevant to performance insofar as they are necessary for understanding their performance behavior from the viewpoint of BS2000/OSD:

- ETERNUS DX
- Symmetrix DMX and VMAX

4.1.1 Components of disk storage systems

The most important components of the disk storage systems which are relevant to performance are:

- server connection
- cache
- disks
- controllers

Server connection

ETERNUS DX or Symmetrix disk storage systems are connected to a BS2000/OSD server via a type FC channel (S servers, performance: 1 Gbit/s) or an FC connection (SQ servers, performance: up to 8 Gbit/s)

The number of connections required depends on the configuration of the disk storage system. The connection via multiple paths must be taken into account. From the performance viewpoint type FC channels or FC connections are recommended without restriction.

The performance of the channels or of the I/O processors in the case of SQ servers is described in chapter “Performance behavior of the servers” on page 25.

Cache

The component of a disk storage system which most influences the performance is the cache. Each input/output request is subject to caching. The hardware service time (I/O time) depends largely on whether the data is located in the cache or must first be read from or transferred to the disk.

If the data block which is being searched for is not contained in the cache, it must be read from the disks (read miss). Such a read miss takes considerably more time than a read hit, in the case of which the data is already in the cache. When sequential processing is recognized, the data is read ahead in order to prevent a read miss.

Since the cache of the disk storage systems is protected against a power failure by batteries, write requests are regarded as completed when they have been written to the cache and verified (write hit rate = 100 %). A write I/O operation therefore takes almost as long as a read hit. Writing to disk takes place asynchronously to the I/O operation. If not enough space is available in the cache, so-called “delayed fast writes” occur, i.e. the cache contents must first be saved before new data can be read into the cache (write hit rate < 100 %). This typically occurs when large quantities of data are reconstructed.

The hit rate is the share of cache hits in the total number of inputs/outputs to/from the disk storage system. Sometimes a distinction is also made between the read hit rate and the total hit rate. The total hit rate contains the cache write hits. Only the read hit rates are examined in this manual.
Vendors of external disk storage systems often specify the “total hit rate”, which also contains the cache write hits and is therefore higher than the read hit rate examined here.

A high hit rate (> 90 %) can also be achieved for read requests by using large caches. A read hit rate of 80 % or higher is normally regarded as good performance for a disk storage system.

Large cache configurations in the ETERNUS DX and Symmetrix disk storage systems are urgently recommended for performance-critical applications.

Disks

The following are generally known for physical disks:

- storage type (HD/SSD)
- storage capacity (in Gbyte, e.g. 300 Gbyte)
- rotational speed (in “rotations per minute”, e.g. 15,000 rpm).

The following are also decisive for the performance of a disk:

- internal connection with connection type (e.g. Fibre Channel, SAS, Serial ATA (SATA)), nearline SAS (NL-SAS)) and data transfer rate (e.g. 8 Gbit/s)
- cache on the drive (drive cache)

Generally the user only sees the performance of the physical disks in the case of a read miss.

From the BS2000/OSD viewpoint no restrictions exist for using the disk sizes offered today.

The data of performance-critical applications should **not** be placed on SATA or NL-SAS disks.

Controllers

The controllers of the disk storage systems process the server’s I/O requests, control cache access and handle access to the physical disk drives.

The number of controllers and the number of paths to the servers which are controlled and to the internal disks depend on the model.

4.1.2 Configuring disk storage systems

This section describes the aspects of configuring a disk storage system which are relevant to performance insofar as they are important from the BS2000/OSD viewpoint.

4.1.2.1 RAID groups and LUNs

During configuration, multiple disks are combined to form RAID groups. These RAID groups have a standard RAID level (RAID1, RAID1/0, RAID5, RAID6).

So-called Logical Unit Numbers (LUNs) are defined for these RAID groups. It is typical that multiple LUNs (logical volumes) are configured on the same physical disk. For information on the effects of logical volumes on the performance, see “Logical volumes and physical disks” on page 52.

The configuration in BS2000/OSD uses these LUNs with LUN numbers and disk type (e.g. D3435).

4.1.2.2 RAID level

All ETERNUS DX and Symmetrix disk storage systems are high-availability disk storage systems which not only comply with the RAID standards originally defined by Berkeley University, but also meet all the availability levels specified by the RAID Advisory Board. They differ only in their configuration options. The RAID level defines the method for creating failure-tolerant and high-performance disk storage systems. The following RAID levels are normal in the disk storage systems for BS2000/OSD:

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

RAID1

Two identical disk drives in each case, original and mirror disk, form a RAID1 group (referred to below as “RAID1(1+1)” for short).

The data these contain is identical. Consequently only 50% of the disk capacity is available to the user. Mirroring does not affect the write performance. In the case of read requests the disk storage system attempts to optimize the operation and read from the more favorable disk. If a disk fails, operations can continue largely without any loss of performance until the disk has been replaced.

RAID1/0

Compared to RAID1, in RAID1/0 there is an additional distribution of the data over multiple disks in the form of “data striping”. A RAID1/0 group therefore consists of multiple - often 4 - disks to which the original data is distributed by means of “data striping”. Plus the same number of disks containing the mirrored data (referred to below, for example, as “RAID10(4+4)” for short).

On Symmetrix systems the term “meta volumes” is also commonly used for RAID1/0. As with RAID1, only 50% of the installed disk capacity is available to the user.
However, data mirroring with RAID1/0 offers the best performance as it increases parallelism for data access and also utilizes the disks more evenly.
With RAID1/0 it is even possible for several disks to fail in the event of a slight loss of performance provided the original disk and mirror disk do not fail simultaneously.

RAID1/0 is recommended for particularly heavy demands with respect to the write performance.

RAID5

RAID5 implements common parity checking for a number of disk drives. As with RAID1/0 the data is distributed in blocks over the disk drives of the RAID5 group using “data striping” and saved with the parity information, which is also distributed over all the drives (“rotating parity”). As the parity information must also be calculated and written when data is written, RAID5 is somewhat more complex than RAID1/0. But as with RAID1/0, thanks to “data striping” it offers advantages in the event of parallel access.

A RAID5 group frequently consists of 4 disks, three quarters of the total capacity being made available for user data and one quarter of the capacity being used for the parity information (referred to below as “RAID5(3+1)” for short). When one disk fails, worse performance must be anticipated than with RAID1/0.

RAID6

As with RAID5 the data stream is divided into stripes. However, not just one but two error correction values are calculated. Data and parity information is thus distributed over the disks in such a way that both sets of parity information are contained on different disks (“dual rotating parity”). In RAID6 up to two disks can fail. The effort for resynchronization, in particular when two disks have failed, is, however, considerably higher than with RAID5.

4.1.2.3 Replication functions of external disk storage systems

External disk storage systems offer functions for local replication within a disk storage system and functions for remote replication between disk storage systems at different geographical locations.

The following functions are available for local replication:

- For ETERNUS DX, the EC (Equivalent Copy) function
- For ETERNUS DX S2 and higher, also the SnapOPC+ function
- For Symmetrix, the TimeFinder product family

The following functions are available for remote replication:

- For ETERNUS DX, the REC (Remote Equivalent Copy) function
- For Symmetrix, the SRDF product family

The software product SHC-OSD is the BS2000 host component for external disk storage systems. It provides information services and commands for controlling the replication functions of the disk storage systems. A detailed description of the replication functions is contained in the “SHC-OSD” manual [28].

Performance behavior with REC / SRDF

If a copy of the data is kept in a remote second disk storage system, each write access is entered first in the local cache and then in the remote cache.

This means that using SRDF at least doubles the hardware service time for write accesses. An increased protocol time for processing the function also has to be taken into account.

Another major factor is the distance between the locations. Double the runtime of light in glass (outward and return directions) must be added in each case.

The write hit rate is usually 100 % (provided there are sufficient “breathers” for saving the cache contents), i.e. each write access is performed quickly. With small blocks, the doubling of the hardware service time is normally of no consequence, but with large blocks (e.g. copy operations) there is a clear increase in the hardware service time.

An additional increase in the hardware service time can occur if the performance capability of the remote data links is insufficient.

In the case of asynchronous replication (Symmetrix: SRDF/A), only the “protocol time” is accrued. The input/output is therefore delayed considerably less.

For details see “I/O peripherals workload” on page 292.

4.1.2.4 Thin Provisioning

The thin provisioning function permits the capacity of disk storage systems to be used efficiently. Devices (LUNs) with preconfigured virtual capacity are offered to the application, while internally the disk storage system provides the necessary physical capacity. Thin provisioning requires a corresponding configuration in the disk storage system.

SHC-OSD supports thin provisioning for the disk storage systems ETERNUS DX S2 and higher and for Symmetrix (where it is called “virtual provisioning”). SHC-OSD has monitoring and information functions for thin provisioning.

A detailed description of the thin provisioning function is contained in the “SHC-OSD” manual [28].

4.1.3 Configuration in BS2000/OSD

Logical volumes

Individual disks are only visible to a certain extent on the current disk storage systems. BS2000/OSD sees “logical volumes” which are assigned to the LUNs (Logical Unit Numbers). The assignment of LUNs to physical disks is largely hidden for BS2000/OSD. The disk storage systems ensure efficient operation by means of RAID levels.

“Large” logical volumes can also be used. These are volumes which are more than 32 Gbyte in size, see the manual “Files and Volumes greater than 32 GB” [3]. The recommendations in section “High-performance operation” on page 55 must then be complied with to permit high-performance operation.

Logical volumes and physical disks

From the BS2000/OSD viewpoint logical volumes are independent units on which I/O operations can be initiated in parallel. When the current high-capacity disks are used, multiple logical volumes are normally located on a disk drive.
When I/O operations to /from logical volumes which are located on the same disk drive are initiated in parallel, the hardware service time is inevitably extended. Whether the extension of the hardware service time leads to an appreciable performance loss depends on the cache hit rate and the utilization of the disk drive – possibly also by users outside BS2000/OSD.

In the case of physical disks which are shared by multiple programs, the times of the I/O requests are not generally evenly distributed. With heavy total workloads, this leads to intolerably long wait periods in front of the disks, in keeping with the laws of queue formation. In other words, the software service time per I/O operation is considerably longer than the hardware service time (see also section “Performing I/O operations” on page 265). In order to minimize these wait periods, the workload on shared disks should not exceed 30%. This demand is based on the assumption that the wait time in front of the disk should never be higher than a third of the hardware service time

Requirements for configuring the disk storage system

From the BS2000/OSD viewpoint the disk peripherals can be configured in accordance with various considerations:

- high capacity (static storage space in the disk storage system, in GBytes)
- high throughput (in Mbyte/s)
- short I/O times (in ms per I/O operation)

As BS2000/OSD has no knowledge of the assignment of logical volumes to physical disks, certain performance requirements must be taken into consideration when the disk storage system is generated. Specifically, these are the

- volume size
- number of I/O paths
- RAID level
- type of server load
  (OLTP mode, throughput-oriented batch mode - the main application must be borne in mind)
- I/O load of the particular volume
  (I/O operations per second or general statement: low / high - the peak load must be borne in mind)

In many cases specific configuration of the disk storage system can or must be left to the maintenance staff. In particular it must be ensured that multiple logical volumes with a heavy load are not placed on the same physical disk. In the case of mixed configurations with other systems, the special characteristics of BS2000 operation should be pointed out (e.g. very high and very even load which also requires the peripherals to perform reliably).

Data format

You are recommended to use NK data format. Compared to K data format, in which the key information must also included, NK data format makes better use of the disk capacity and the cache memory, thus achieving shorter hardware service times.

Measurements

You can ascertain the following with SHC-OSD:

- Which logical volumes are configured on a disk (not in the case of thin provisioning)
- The physical disk(s) on which a BS2000 volume resides (not in the case of thin provisioning)
- For ETERNUS DX: the RAID group in which a BS2000 volume resides
- For ETERNUS DX: which other volumes reside in the same RAID group

When the software product SHC-OSD is installed, openSM2 collects measured values relating to Symmetrix systems in the report group `STORAGE-SYSTEM-SYMMETRIX`, e.g. the number of read and write operations and the number of read hits and consequently the read hit rate. Information can also be ascertained for physical devices, e.g. the throughput rate. Further information on this subject is provided in chapter “Use of the openSM2 monitoring system” on page 271.

You require the software product openSM2 (Open Systems) to monitor ETERNUS DX disk storage systems.
The drive workload on ETERNUS DX cannot be determined using BS2000 resources and can therefore not be measured using openSM2. It consists of the sum of the I/O loads of the relevant logical volumes.

4.1.4 High-performance operation

To permit high-performance operation of disk storage systems, the server must be connected via a type FC channel or FC connection.

A significant increase in performance can also be achieved by using the BS2000 function Parallel Access Volume (PAV) on S servers or the Remote System Call (RSC) function on SQ servers (used by default). Suitable RAID levels must be selected to enable the potential of PAV to be utilized.

Fibre Channel connection type

To operate high-performance peripherals, the server must be connected via a type FC channel (S-Server) or an FC connection.

The performance capability of the current peripherals cannot be utilized fully with type S channels.

Parallel Access Volume (PAV, DPAV)

In the case of single disk access (default), only one I/O request for a logical volume is processed at a time on S servers. Further requests are serialized in BS2000/OSD.

Alternatively the BS2000 function PAV supports parallel disk access on S servers. PAV is used to assign a number of device addresses (alias devices) to a single logical volume (base device). This enables multiple I/O requests to be initiated simultaneously on a volume. Multiple cache hits can then be processed simultaneously in the disk storage system, while a physical disk access can be executed in parallel in the event of a cache miss. Detailed information on PAV can be found in the “Introductory Guide to Systems Support” [10].

In the case of a type FC channel, alias devices must be generated in BS2000/OSD. No intervention is required in the controller.

“Static PAV” requires foresighted planning with regard to the future device workload. Devices with a heavy workload must be assigned the correct number of alias devices in advance.

In the case of “dynamic PAV” for a type FC channel, the IORM utility routine automatically assigns alias devices (DPAV devices) to the volumes with a heavy workload which profit most from this. Alias devices are not generated. I/O bottlenecks are alleviated in this way.

Remote System Call (RSC)

A function which is similar to PAV is provided for SQ servers in the form of the function Remote System Call (RSC) for disks. This permits up to six simultaneous I/O operations to/from a volume or device. Serialization takes place in the disk storage system. RSC is used by default; the user does **not** need to specify any special setting in the configuration to use RSC or to permit parallel I/O operations.

PAV and RAID level

Use of PAV or RSC enables improved I/O times and higher throughput to be achieved. The RAID level determines the actual performance enhancement (see section “PAV and RAID level” on page 63):

- Better I/O times are achieved for all RAID levels.
- Higher throughput is achieved only if the RAID level used can also utilize the parallelism of PAV or RSC. This is possible above all with RAID levels with “data striping”, in other words RAID1/0, RAID5 and RAID6. This only applies with restrictions for RAID1.

For high-performance application RAID1/0 or, if the disk capacity is extremely important and the random write load is not too high, RAID5 or RAID6 is recommended, on S servers together with DPAV.

Recommendation for high-performance operation of disk storage systems

- Connection via type FC channel or FC connection only
- As far as possible use new disk storage systems:
  - with the latest firmware version
  - with sufficient cache memory (must be large enough to ensure that a read hit rate of at least 80% is achieved in OLTP mode)
  - with high-speed disks (at least 10,000 rpm)
  - large disks can be used
- RAID level RAID1/0 or, if the random write load is not too high, RAID5 or RAID6
  - use of PAV, or preferably DPAV, on S servers,
  - then “large” volumes (with more than 32 Gbyte) can also be used
- Block size of 160 KB per I/O operation in the case of throughput-oriented applications, if possible
- general increase in parallelism
  (in the operating system, utility routines and applications)

4.1.5 Load types

Applications

The configuration of the disk peripherals is influenced both by both the load profile and by the time requirements of the application. In practice there are essentially two load types, which can be characterized as follows:

	OLTP mode (database applications)	**Applications with large block size (batch processing, data backup)**
Typical block size per input/output	“small”: 2 KB or 4 KB	“large”: 16 KB or more
Load type	Input/output rate (I/Os/s)	Throughput (Mbyte/s)
Typical input/output operations	Random access	Random access Sequential access

Applications in BS2000/OSD can use block sizes of up to 160 KB for each I/O operation. Certain disk types even permit block sizes of up to 480 KB. This enables the throughput to be increased significantly.

Applications can implement the performance enhancements using PAV or RSC and the corresponding RAID levels only if they are able to process requests in parallel without these seriously impairing each other. This is generally the case in OLTP mode. The current versions of the data backup products HSMS/ARCHIVE and FDDRL are also prepared for this. In batch mode the performance enhancements can only be implemented in the case of parallel I/O operations.

Depending on the application type, different performance targets are focused on. In the case of data backup and in batch mode a high throughput is aimed for; in OLTP mode good response times are expected. The I/O operations play an important part in the throughput rate/dwell time and the response time/transaction time (see section “Characteristic values for describing the performance of an IT system” on page 15). In the following the I/O times from the viewpoint of the user (“software service time”) and the time solely for hardware operation (“hardware service time”) are examined (see section “Performing I/O operations” on page 265). The “software service time” mainly comprises the hardware service time and the wait times for initiating the I/O operation.

Standardized load types

Standardized load cases are used for the measurements in the sections below:

- “Random25”

  Random accesses with 25% writing and 75% reading. With short blocks this case reflects OLTP mode.

- “Sequential write”, “Sequential read”

  Sequential writing and reading. With large blocks this case reflects batch processing.

4.2 Disk storage systems connected to S servers

This section contains measurement results and conclusions for the performance of the ETERNUS DX and Symmetrix disk storage systems when connected to S servers via type FC channels.

4.2.1 ETERNUS DX

This section contains measurement results and conclusions for the performance of the ETERNUS DX8400 disk storage system when it is connected to S servers via type FC channels.

The new generation ETERNUS DX8700 S2 is now available. This generation stands out on account of the significantly larger caches and special performance optimizations for better I/O throughput when data is processed sequentially (throughput increased by up to 300%). The performance in random processing is comparable with that of ETERNUS DX8400. ETERNUS DX8700 S2 has not yet been included in the measurements shown.

The following configuration was used to make the measurements below:

- S200-20 server under VM2000 (VMs with `IO-PRIORITY` privilege).
- Connection of the S server to the disk storage system via 4 type FC channels each with a performance of 1 Gbit.
- The ETERNUS DX8400 disk storage system with a 32 GB cache and physical disks each with a capacity of 450 GB and operating at 15,000 rpm
- 16 volumes on disks of the type D3435 with data format NK2 and with little mutual interference (few logical volumes on a physical device)

You are recommended to use ETERNUS DX disk storage systems with RAID1/0. Measurements of various RAID variants yield the best results for RAID1/0.

4.2.1.1 Throughput with one task

The two tables below show the throughput (the I/O rate in I/Os per second and throughput in Mbyte/s) which a user task can achieve in the case of disk accesses to the ETERNUS DX8400 disk storage system.

The throughput was measured for the standardized load types with different block sizes and a read hit rate of 100%:

One task: I/O operations/s	Block size (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

One task: Mbyte/s	Block size (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 Throughput with up to four type FC channels

The two tables below show the throughput (the I/O rate in I/Os per second and throughput in Mbyte/s) in the case of disk accesses to the ETERNUS DX8400 disk storage system when up to four type FC channels are used.

The throughput was measured for the load type “Random25” with different block sizes and a read hit rate of 100%:

“Random25”: I/O operations/s	**Block size (KB)**			
	2	**16**	**32**	**160**
1 type FC channel	7,175	5,410	3,140	710
4 type FC channels	28,725	20,885	12,305	2,800

“Random25”: Mbyte/s	**Block size (KB)**			
	2	**16**	**32**	**160**
1 type FC channel	13	84	98	111
4 type FC channels	56	326	384	437

Type FC channel scale very well. In other words, n channels of the type FC permit an almost n-fold throughput. A prerequisite for this is that the BS2000 CPUs and the peripherals are sufficiently powerful.

Compared with the throughput of the Symmetrix VMAX disk storage system (see page 72), these values are lower. The reason for this is that for the measurements on Symmetrix VMAX a cache was used which was three times the size of that used for the measurements on ETERNUS DX8400.
25% write I/Os and 75% read I/Os take place in the “Random25” load. In view of the significantly larger cache, in the steady state (almost) all read I/Os from the cache can be dealt with, thus resulting in shorter I/O times and a greater throughput.
This effect is also to be expected with the larger caches of ETERNUS DX8700 S2.

4.2.1.3 Hardware service times

Hardware service times when reading

The table below shows the I/O times (hardware service times) achieved for read hits and read misses in typical configurations with typical workloads. The measurements relate to RAID level RAID1/0 (see the recommendation on page 59).

The hardware service times depend not only on the I/O rate but also on the configuration of the disk storage system. In small configurations the times can be somewhat more favorable. In very large configurations the times can be worse.

I/O time when reading (ms)	Block size (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

These times apply for an average load in which neither disks nor channels are overloaded.

The following differences were measured for other RAID levels:

- RAID1: approx. 5% better
- RAID5: approx. 10% worse
- RAID6: approx. 10% (long blocks) to 15% (short blocks) worse

Hardware service times in OLTP mode

With high read miss rates, the I/O times for a typical OLTP load (“Random25”) are generally approx. 10% worse than the I/O times when reading (see section above).

4.2.1.4 PAV and RAID level

For disk storage systems, not only a connection via type FC channel is recommended for S servers, but also the use of PAV/DPAV (Dynamic Parallel Access Volume) together with a suitable RAID level. This section shows how the I/O times and the throughput (I/O rates) can be enhanced by using PAV.

The diagrams below show the I/O times (software times = sum of the hardware service times and software service times) for various RAID levels and with a load which is typical for OLTP mode:

- Short blocks (2 Kbytes)
- Standardized load type “Random25”
- Read hit rate 80%

The following configuration was used to make the measurements:

- As described on page 59.
- The measurement was made for each RAID group on a volume.
  4 tasks were active per volume.

As the load increased (I/O rate) the I/O times (hardware + software) were compared at various RAID levels. Measurements were made for each RAID level using a different number of alias devices.

RAID groups generated on the DX8400 with 16 volumes each:

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

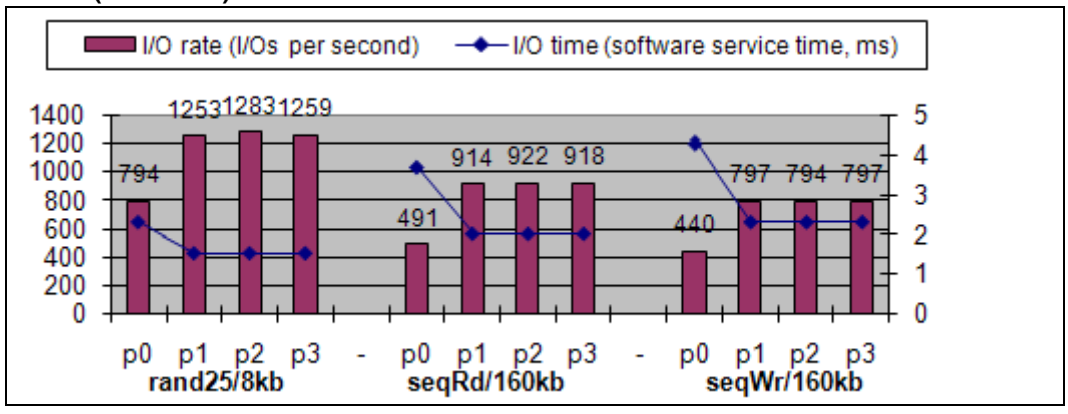
PAV variants:

- p0: no PAV
- p1-3: PAV with1-3 alias devices

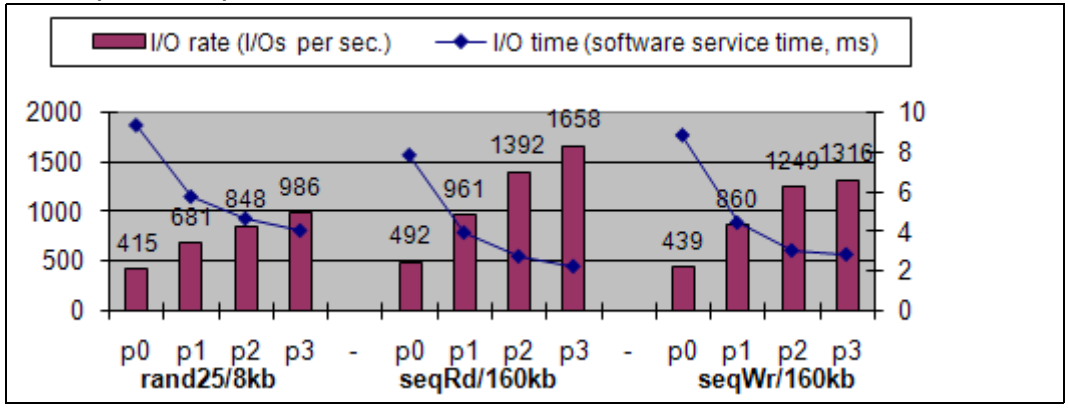
Load cases investigated:

- rand25/8kb: Random25 with 8 KB blocks (mapping of TP mode)
- seqRd/160kb: Sequential read with160 KB blocks (throughput-oriented load)
- seqRd/160kb: Sequential write with160 KB blocks (throughput-oriented load)

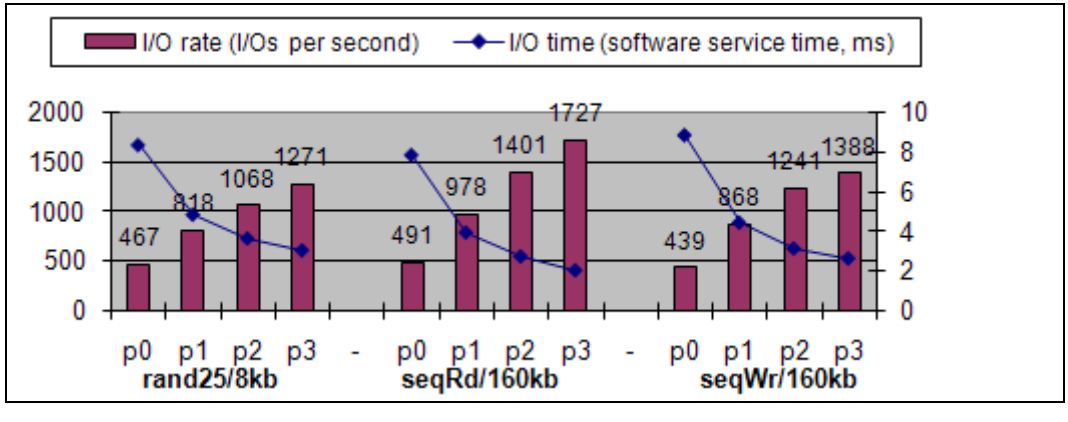
RAID1 (1+1 disks)



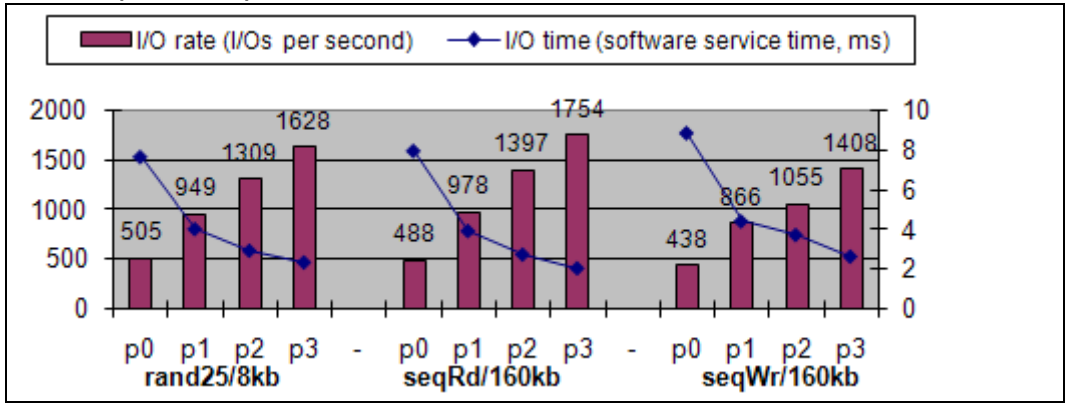
RAID5 (3+1 disks)



RAID6 (6+2 disks)



RAID1/0 (4+4 disks)



RAID1/0, RAID6, RAID5:
In the case of this RAID level with data striping, considerable improvements are apparent both in the I/O rates and in the I/O times. When the load is correspondingly high, the improvements increase with the number of alias devices and apply for transaction- and throughput-oriented load cases.

With RAID1 (no data striping) one alias device is sufficient. Further alias devices bring no further improvement.

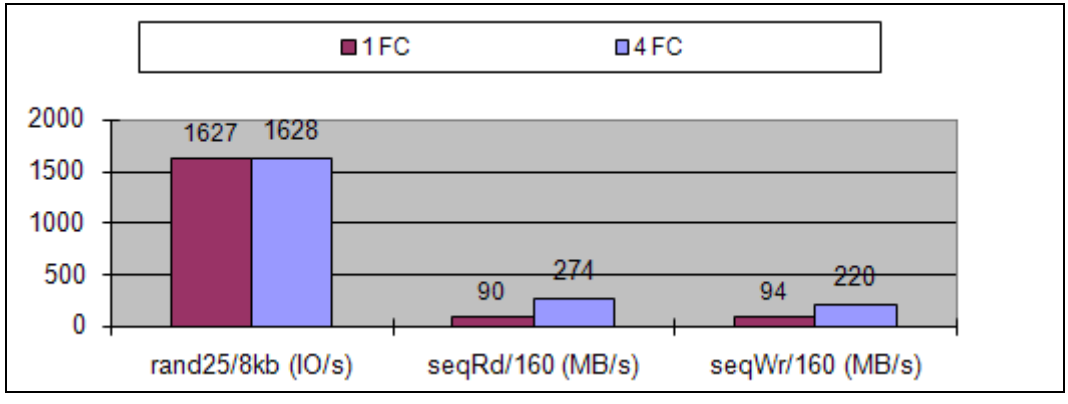
Without PAV the software service times are considerably longer than the hardware service times. The I/Os wait in the operating system. As the number of alias devices increases, the software service time is reduced and, with 4 alias devices (and 4 tasks) is equal to the hardware service times. The hardware service times increase only very slightly as a result of the increased parallelism on the disk storage system.

4.2.1.5 Limits of PAV

To achieve the desired improvements with PAV, the constraints described in this section must be complied with.

Connection via multiple type FC channels

The use of PAV requires a sufficiently high-performance connection of the ETERNUS DX disk storage system to BS2000/OSD. The figure below uses RAID1/0 with PAV variant p3 to show the reduced performance when a connection over only one channel is used compared to a connection with four channels. The load cases rand25/8kb (I/Os per second), seqRd/160kb (MB/s) and seqWr/160kb are investigated. The file size of 10 GB per task is so large that almost 100% cache misses is achieved.



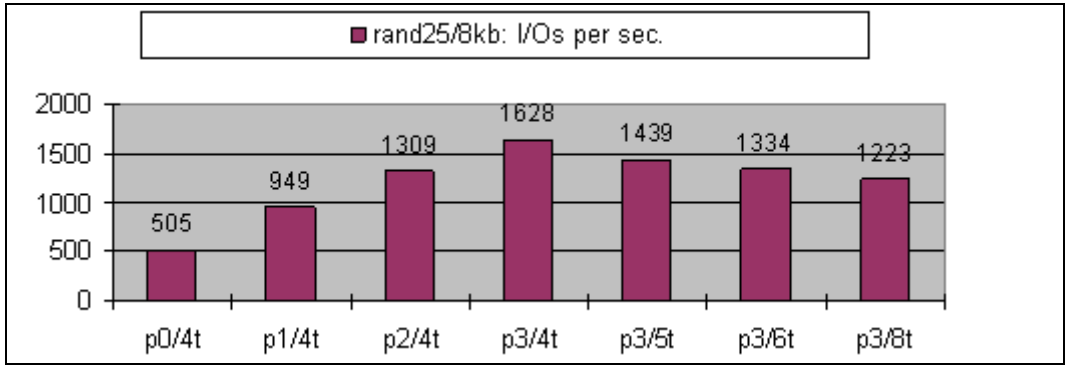
In the case of the relatively short blocks of load case rand25/8kb, a tape FC channel (1 Gbit/s) does not cause a bottleneck.

The case is different with big blocks. Here the full performance of PAV variant p3 can only be achieved with multiple FCs. In the example at least 3 connection paths are required.

Many parallel I/Os

The measurements made so far for PAV involved 4 tasks which executed parallel I/Os on one volume. Here it was apparent that as the number of alias devices grew, considerable improvements were achieved.

In this section the rand25/8kb load will be used to show that when there are too many parallel tasks (and consequently I/Os) the total I/O rate which can be achieved decreases again. The ratio of “Number of alias devices / Number of tasks” is varied. The measurements were made using large files (10 GB per task; with 5 GB and 6 to 8 tasks there are no differences) and consequently almost 100% cache misses.



When there are to many parallel tasks which access files which are too large, the total I/O which can be achieved decreases. The reason for this are the complex accesses to the physical disks.

Without PAV this decrease would already begin with 505 I/Os after “p0/4t” (no PAV, 4 tasks).

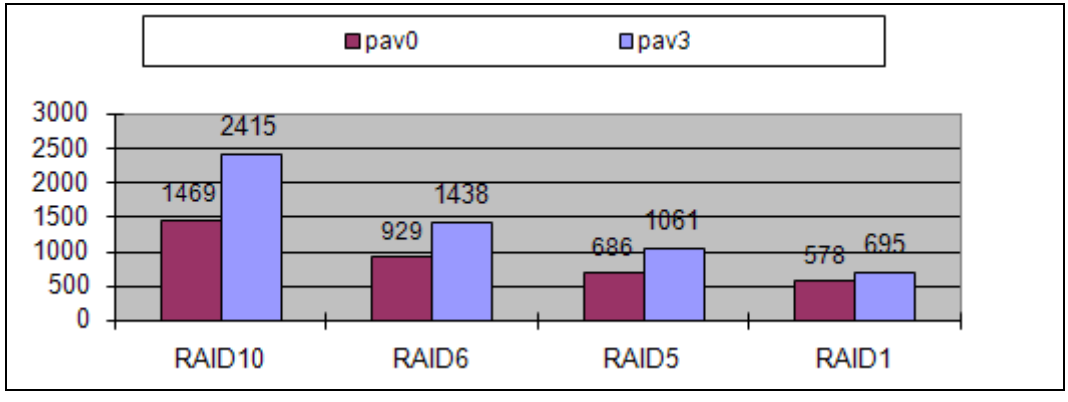
Performance of the RAID group

The performance of a volume can be enhanced with PAV only if the RAID group concerned still has reserve capacity. If the RAID group is already fully utilized, for instance because of the workload from other volumes, no further improvement is possible.

The figures below show whether improvements are still possible with PAV despite the very heavy workload on all volumes.

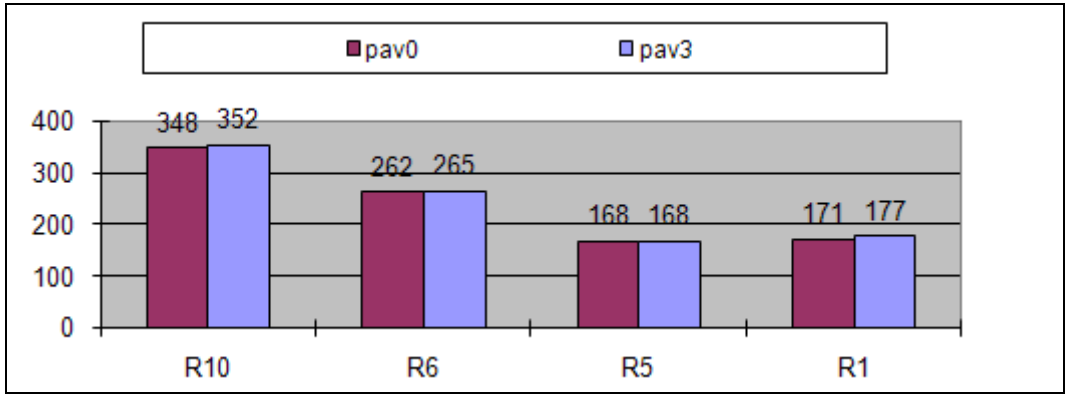
4 parallel tasks are started for each of the 16 volumes or a RAID group. Each task performs one of the load types rand25/8kb, seqRd/160kb or seqWr/160kb. The results without PAV (pav0) and with PAV with 3 alias devices (pav3) are compared. In addition to the RAID groups, the behavior in the case of “cache hit” is also specified.

Load rand25/8kb (I/Os)



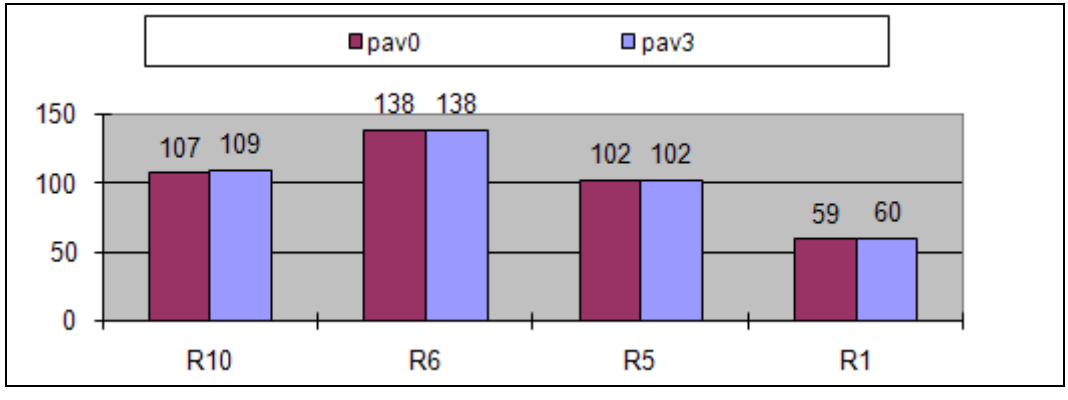
Cache hit: pav0 = 23071, pav3 = 23030 I/Os per second

Load seqRd/160kb (I/Os)



Cache hit: pav0 = 349, pav3 = 354 I/Os per second

Load seqWr/160kb (I/Os)



Cache hit: pav0 = 365, pav3 = 351 I/Os per second

With the transaction-oriented load case rand25/8kb, significant improvements can be achieved although the RAID groups also have a very heavy workload without PAV.

With the throughput-oriented load cases, no further increase is possible: the disks are fully utilized.

No improvements can be achieved by means of PAV in the case of cache hit, either.

4.2.2 Symmetrix DMX and VMAX

This section contains measurement results and conclusions for the performance of the Symmetrix disk storage systems DMX and VMAX when connected to S servers via type FC channels.

The throughput of a Symmetrix DMX-4 and VMAX disk storage system is comparable with that of DMX-3. The hardware service times in the case of read hits are also comparable. The hardware service times in the case of read misses are shorter in the case of DMX-4 and VMAX than in the case of DMX-3.

The following configuration was used to make the measurements below:

- S server with 4 CPUs under VM2000 with a uniprocessor guest system (`IO-PRIORITY` privilege) with a dedicated CPU (this corresponds to the conditions of native uniprocessor mode with approx. 320 RPF)
- Connection of the S server to the disk storage system via 4 type FC channels each with a performance of 1 Gbit.
- Disk storage systems:
  - Symmetrix DMX-3 with a 32 GB cache and physical disks each with a capacity of 146 GB and operating at 15,000 rpm
  - Symmetrix VMAX with a 96 GB cache and physical disks each with a capacity of 146 GB and operating at 15,000 rpm
- 16 volumes on disks of the type D3435 with data format NK2 and with little mutual interference (few logical volumes on a physical device)

4.2.2.1 Throughput with one task

The two tables below show the throughput (the I/O rate in I/Os per second and throughput in Mbyte/s) which a user task can achieve in the case of disk accesses to the Symmetrix DMX-3 disk storage system.

The throughput was measured for the standardized load types with different block sizes and a read hit rate of 100%:

One task: I/O operations/s	Block size (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

One task: Mbyte/s	Block size (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 Throughput with up to four type FC channels

The two tables below show the throughput (the I/O rate in I/Os per second and throughput in Mbyte/s) in the case of disk accesses to the Symmetrix VMAX disk storage system when up to four type FC channels are used.

The throughput was measured for the load type “Random25” with different block sizes and a read hit rate of 100%:

“Random25”: I/O operations/s	**Block size (KB)**					
	2	**4**	**8**	**16**	**32**	**160**
1 type FC channel	8,800	8,800	8,200	5,700	3,400	700
2 type FC channels	17,000	17,000	16,000	11,000	6,800	1,400
4 type FC channels	34,000	33,000	29,000	21,000	12,000	2,600

“Random25”: Mbyte/s	**Block size (KB)**					
	2	**4**	**8**	**16**	**32**	**160**
1 type FC channel	18	36	67	93	110	115
2 type FC channels	34	69	130	180	220	225
4 type FC channels	69	135	235	345	400	425

Type FC channel scale very well. In other words, n channels of the type FC permit an almost n-fold throughput. A prerequisite for this is that the BS2000 CPUs and the peripherals are sufficiently powerful.

4.2.2.3 Hardware service times

Hardware service times when reading

The table below shows the I/O times (hardware service times) achieved for read hits and read misses in typical configurations with typical workloads. The times for the read misses are shown for each RAID level.

The hardware service times depend not only on the I/O rate but also on the configuration of the disk storage system. In small configurations the times can be somewhat more favorable. In very large configurations the times can be worse.

I/O time when reading (ms)	Block size (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 service times in OLTP mode

The tables below show the I/O times (hardware service times) for different read hit rates in typical configurations with typical workloads in OLTP mode (load type “Random25”). For write requests it is assumed that the disk storage system’s cache is sufficiently large, i.e. the write request is completed after it has been transferred to the cache. Transfer to the physical disks is asynchronous.

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

I/O time “Random25” (ms)	Block size (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)

I/O time “Random25” (ms)	Block size (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

The I/O times apply for a “normal” load, i.e. an I/O rate which does not overload the logical volume, where RAID1/0(4+4) tolerates the highest load and RAID1(1+1) the lowest.

4.2.2.4 PAV and RAID level

For high-performance systems, not only a connection via type FC channel is recommended for S servers, but also the use of PAV/DPAV (Dynamic Parallel Access Volume) together with a suitable RAID level. This section shows that the I/O times and the throughput (I/O rates) can be enhanced by using PAV.

The diagrams below show the I/O times (software times = sum of the hardware service times and software service times) for various RAID levels and with a load which is typical for OLTP mode:

- Short blocks (2 Kbytes)
- Standardized load type “Random25”
- Read hit rate 80%

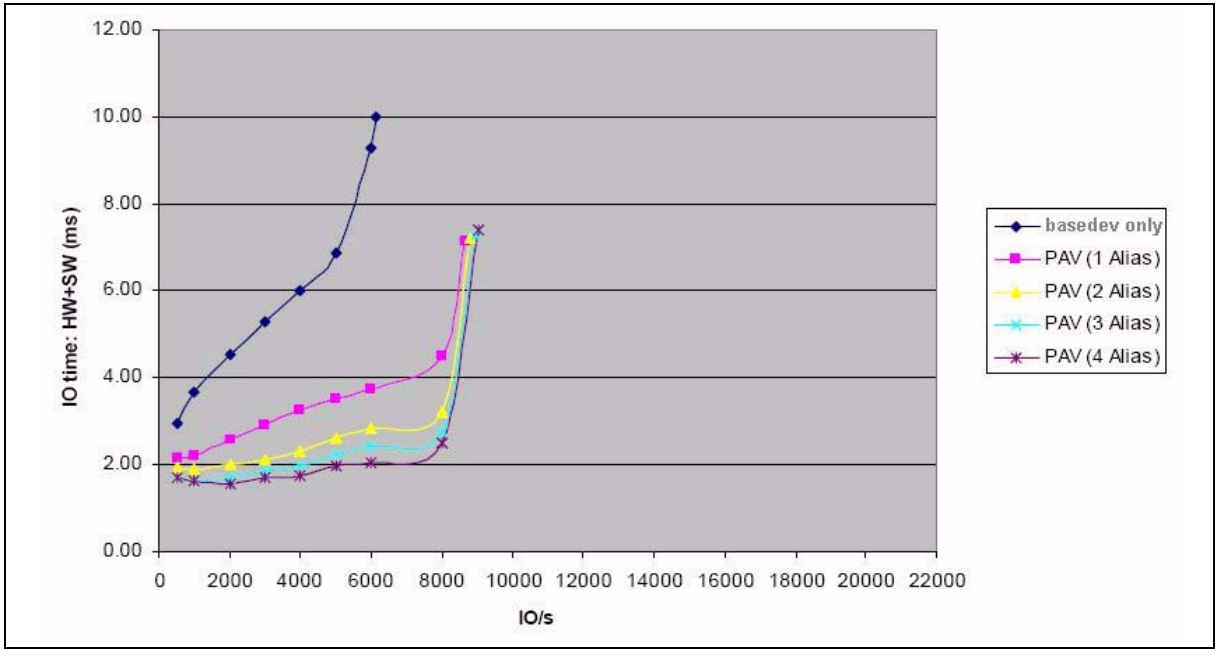
The following configuration was used to make the measurements:

- As described on page 70.
- 8 volumes each on independent physical disks. 8 tasks were active per volume.

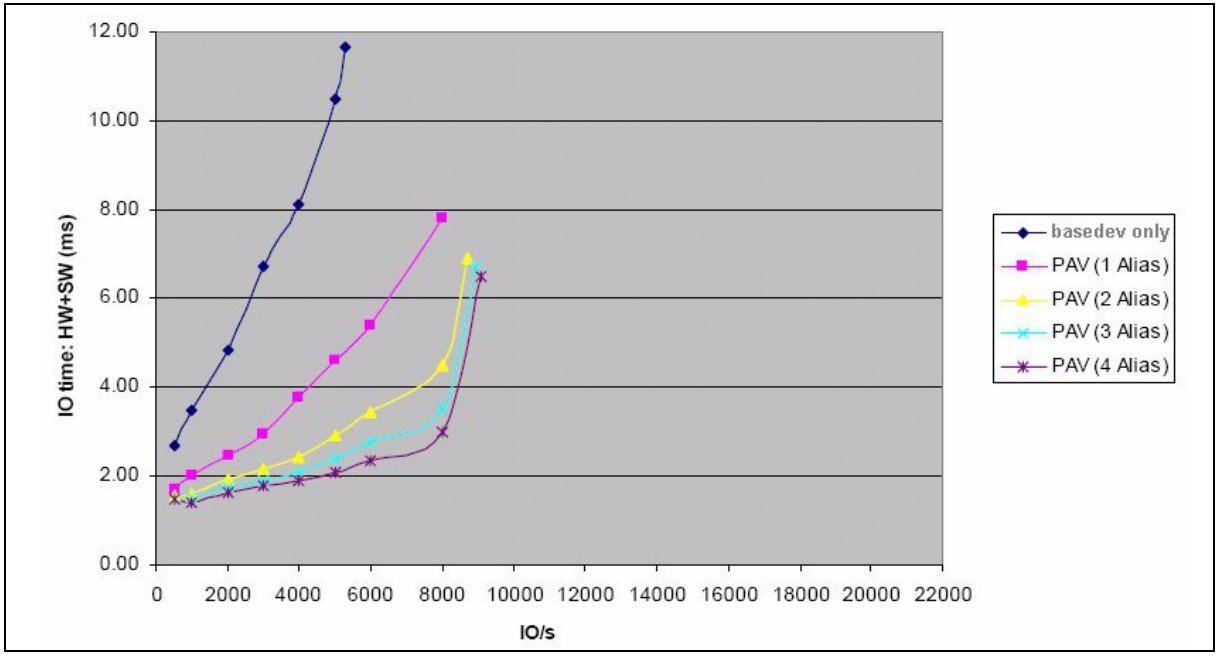
As the load increased (I/O rate) the I/O times (hardware + software) were compared at various RAID levels. Measurements were made for each RAID level using a different number of alias devices.

The measurements below were made using the Symmetrix DMX-3 disk storage system. The results for PAV also apply analogously for Symmetrix DMX-4 and VMAX.

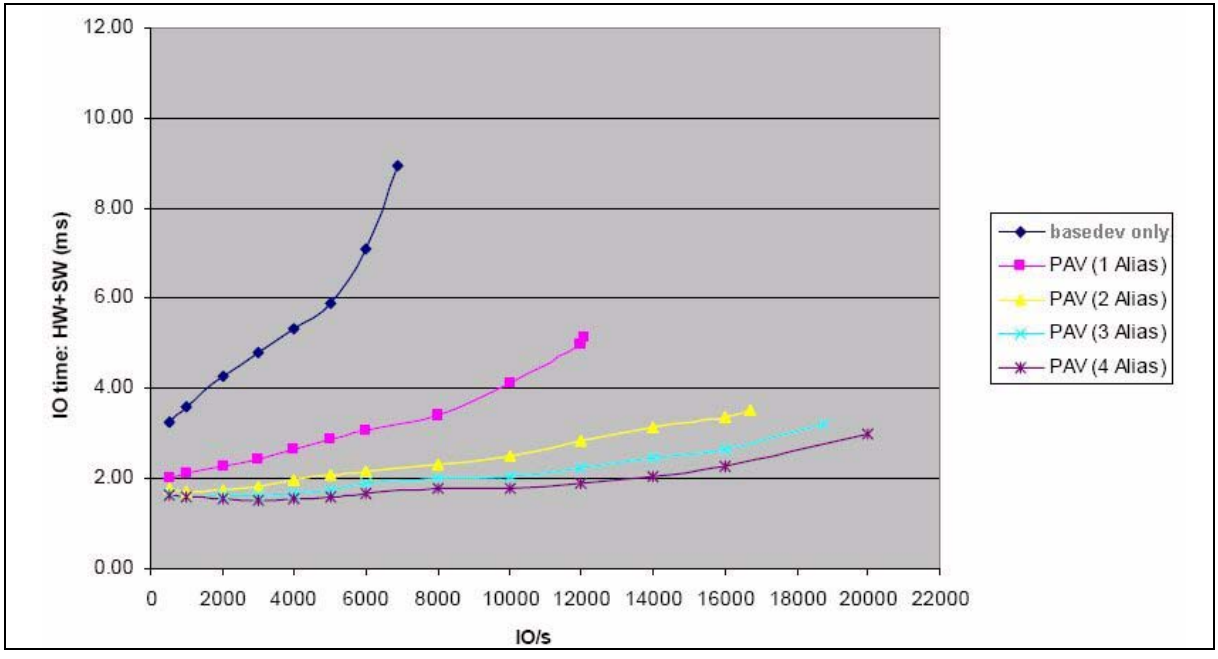
RAID1 (1+1 disks)



RAID5 (3+1 disks)



RAID1/0 (4+4 disks)



PAV with RAID1/0 enables the parallelism of the I/O load to be used effectively.

Considerable improvements are achieved with PAV. With just one alias device the I/O times decrease and the I/O rates increase significantly. 2 alias devices further enhance performance. More alias devices only result in slight improvements.
When DPAV is used, the IORM utility routine ensures that a sufficient numer of alias devices is assigned.

At the same I/O rate PAV results in considerably better I/O times (hardware and software times) at all RAID levels.

Considerably better I/O times are achieved with PAV at RAID levels RAID1 and RAID5. Because of the greater effort involved in writing (calculating and writing the parity information), the I/O rate of RAID5 is approximately the same as that of RAID1 despite “data striping” over multiple disks. The I/O times and I/O rates of RAID6 are approximately the same as those of RAID5.

RAID1/0 achieves a considerably higher I/O rate. This is due not only to “data striping”, but also to the larger number of disks.
The I/O rate achieved for RAID1/0 in this configuration with acceptable I/O times is over 20,000 I/Os per second.

RAID6 (6+2 disks)

The performance of RAID6 (6+2 disks) for S servers with the DMX-3 disk storage system is evaluated in comparison with RAID5 (3+1) and RAID1/0 (4+4). Capacity and security aspects are not examined. The behavior with PAV with up to 4 alias devices is examined.

Read

- Without PAV good throughputs of around 70 Mbytes/s are achieved with all three RAID levels when sequential reading takes place.
- With PAV an excellent throughput is achieved on one logical volume when multiple alias devices are used. In the case of multiple tasks, throughput is 250 Mbytes/s for all three RAID levels.

Write

- Without PAV the throughput on all RAID levels is around 60 Mbytes/s when sequential writing takes place. However, when random writing takes place, the throughput with RAID5 and RAID6 drops to 20-30 Mbytes/s, in other words considerably below the RAID1/0 value (60 Mbytes/s).
- When writing with PAV and multiple alias devices, RAID1/0 also provides higher performance than RAID5 or RAID6, in which the parity information also needs to be calculated and written. In the case of RAID1/0 an excellent throughput of more than 200 Mbytes/s is achieved when a sequential write takes place to one volume. A throughput of around 100 Mbytes/s is achieved with RAID5 and RAID6. The advantages of RAID1/0 are even more apparent in the case of random writing (a factor of 3-5).
- The throughput values of RAID5 and RAID6 are at the same level.

Evaluation:

- PAV enables the throughput on S servers to be significantly increased, even with 2 or 3 alias devices. (D)PAV is recommended without restriction.
- TP mode:
  When writing takes place in normal TP mode, it is assumed that sufficient space is available in the disk storage system's cache, in other words that no "Delayed Fast Write" is necessary. All three RAID levels are recommended.
- Data backup:
  In this case data is read from the disks. All three RAID levels are recommended.
- Data restoration:
  In this case data is written to the disks. RAID1/0 is advantageous here. The restrictions on the tape side must be borne in mind (see section page 112).

4.3 Disk storage systems connected to SQ servers

Input/output on FC peripherals is done via I/O processors and the PCI controllers of the SQ server.

4.3.1 ETERNUS DX

This section contains measurement results and conclusions for the performance of the ETERNUS DX440 S2 disk storage system with an FC connection to an SQ server.

The following configuration was used to make the measurements shown:

- SQ210-80F server in native mode
- Connection of the SQ server to the disk storage system via 4 FC connections each with a performance of 8 Gbit. The disks within the disk storage system could be reached via 2 paths.
- The ETERNUS DX440 S2 disk storage system with a 96 GB cache and physical disks each with a capacity of 450 GB and operating at 15,000 rpm
- 16 volumes on disks of the type D3435 with data format NK2 and with little mutual interference (few logical volumes on a physical device)

You are recommended to use ETERNUS DX disk storage systems with RAID1/0. Measurements of various RAID variants yield the best results for RAID1/0.

4.3.1.1 Throughput with one task

The tables below show the throughput (the I/O rate in I/Os per second and throughput in Mbyte/s) which a user task can achieve in the case of disk accesses to the ETERNUS DX 440 S2 disk storage system.

The throughput was measured for the standardized load types with different block sizes and a read hit rate of 100%:

One task: I/O operations/s	Block size (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

One task: throughput [Mbyte/s]	Block size (KB)			
	2	**16**	**32**	**160**
“Random25”	5	31	55	107
“Sequential read”	5	32	54	102
“Sequential write”	4	29	55	128

The throughput increases significantly with increasing block size. You are recommended to work with large blocks.

4.3.1.2 Throughput with 32 tasks

The tables below show the throughput (the I/O rate in I/Os per second and throughput in Mbyte/s) which can be achieved in the case of disk accesses to the ETERNUS DX 440 S2 disk storage system.

The throughput was measured for the standardized load types with different block sizes and a read hit rate of 100% with 32 tasks operating in parallel (2 tasks together process one volume):

32 tasks: I/O operations/s	**Block size (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: throughput [Mbyte/s]	**Block size (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

The CPUs of the SQ210-80F were fully utilized by the small blocks when the measurements were made. The maximum throughput of the disk storage system could not be achieved with the small blocks.

Because of the large cache, in the steady state (almost) all read I/Os from the cache could be handled. Most write I/Os were also acknowledged after the data had been placed in the cache. This results in shorter I/O times and greater throughput both when writing and when reading.

4.3.1.3 Hardware service times

Hardware service times when reading

The hardware service times depend not only on the I/O rate but also on the configuration of the disk storage system (above all on the cache size).

In the disk storage systems with a large cache most of the I/Os are handled via the cache. Disk accesses are performed asynchronously in the disk storage system and have hardly any influence on the hardware service times.

When small blocks (2 KB to 16 KB) are used, the hardware service times are up to 1 ms. When small blocks (32 KB to 160 KB) are used, the hardware service times are around 2 - 8 ms.

Hardware service times in OLTP mode

With high read miss rates, the I/O times for a typical OLTP load (“Random25”) are generally approx. 10% worse than the I/O times when reading (see section above).

4.3.2 Symmetrix DMX, VMAX

This section contains measurement results and conclusions for the performance of the Symmetrix DMX-3 disk storage system (146 GB disks operating at 15,000 rpm) when it is connected to SQ servers via FC.

The throughput of a Symmetrix DMX-4 or VMAX disk storage system is comparable with that of DMX-3. The hardware service times in the case of read hits are also comparable. The hardware service times in the case of read misses are shorter in the case of DMX-4 and VMAX than in the case of DMX-3.

4.3.2.1 Throughput with one task

The two tables below show the throughput (the I/O rate in I/Os per second and throughput in Mbyte/s) which a user task can achieve in the case of disk accesses to the Symmetrix DMX-3 disk storage system via FC.

A Symmetrix DMX-3 disk storage system with a 32 GB cache connected to an SQ200-80E server via 4 FC connections each with a performance of 2 Gbit, with 16 volumes on disks of the type D3435 with data format NK2 and with little mutual interference was used to make the measurements.

The throughput was measured for the standardized load types with different block sizes and a read hit rate of 100%:

One task: I/O operations/s	**Block size (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

One task: Mbyte/s	**Block size (KB)**			
	2	**16**	**32**	**160**
“Random25”	4	27	47	83
“Sequential read”	4	25	48	85
“Sequential write”	3	28	46	80

The throughput increases significantly with increasing block size. You are recommended to work with large blocks.

The NK2 data format supplies the best results. The throughput rates are lower with K data format: by around 5-10 % with short blocks, and by up to around 20-30 % with large blocks.

4.3.2.2 Hardware service times

Hardware service times when reading

The table below shows the I/O times (hardware service times) achieved for read hits and read misses in typical configurations with typical workloads. The times for the read misses are shown for each RAID level.

The hardware service times depend not only on the I/O rate but also on the configuration of the disk storage system. A Symmetrix DMX-3 disk storage system with a 32 GB cache connected to an SQ200-80E server via 4 FC connections each with a performance of 2 Gbit, with 16 volumes on disks of the type D3435 with data format NK2 and with little mutual interference was used to make the measurements.

In small configurations the times can be somewhat more favorable. In very large configurations the times can be worse.

I/O time when reading (ms)	Block size (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

The hardware service times are very much dependent on the cache size of the disk storage system. As the cache size increases, the number of cache misses and hardware service times are increasingly reduced.

Hardware service times in OLTP mode

The table below shows the I/O times (hardware service times) for different read hits with different RAID levels in typical configurations with typical workloads in OLTP mode (load type “Random25”).

I/O time “Random25” [ms]	Block size (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

The NK2 data format supplies the best results.

The hardware service times are higher with K data format: by around 5-10 % with short blocks, and by up to around 20-30 % with large blocks.

4.4 Net-Storage

BS2000/OSD V9.0 and higher permits UNIX file systems to be accessed via NFS. This enables BS2000 files to be saved and processed by file servers and Net-Storage in released directories.

From the BS2000 viewpoint, on the storage medium Net-Storage, too, data is saved to volumes which are, however, permanently assigned to a pubset. These serve as a storage extension for the pubset. It is thus possible to exceed the maximum possible storage space of an SF pubset (4 Tbyte).

Basic information on Net-Storage can be found in the “Introductory Guide to Systems Support” [10].

Net-Storage is used primarily to save (large) files which do not need to be accessed with maximum performance. Performance-critical data (e.g. database files) should still be saved to pubsets.

There are several reasons for this:

- The performance capability of state-of-the-art networks is constantly increasing, but the bandwidth must be shared with other participants in the network. Consequently, in contrast to FC peripherals, no exact forecasts of throughput and latency periods can be made.
- Read access to Net-Storage is at a higher performance level than write access. A considerably higher throughput can be achieved by reading and writing large I/O blocks than with small I/O blocks.
- The throughput increases with the number of parallel jobs and with the block size until a bottleneck is reached. Thus with sequential processing of large blocks (HSMS/ARCHIVE, COPY, File Transfer), very high throughput values can be achieved which extend to the limits of the bandwidth provided in the network.
- If the configuration and the hardware of the network permit, jumbo frames should be used (MTU 8,000 bytes). The MTU must also be configured or supported on all the network components involved (e.g. switches).

Details of the configuration of Net-Storage are provided in the “Net-Storage Installation Guide”. You will find the “Net-Storage Installation Guide” on the internet at *http://www.fujitsu.com* under the product description of BS2000/OSD-BC V9.0.

4.5 Connection to a LAN

This section analyzes the communication performance of the BS2000/OSD servers.

The following typical application types are taken into considered:

- Transaction-oriented operation:
  Sending and receiving short messages (10 bytes).
  Measured in transactions per second (TA/s).
- Throughput-oriented operation (mass data transfer similar to file transfer):
  Sending or receiving messages with a length of 8, 16, 32, 64 Kbytes.
  Measured in megabyte per second (Mbyte/s).

Under optimum conditions, the results apply for a Sockets application (TCP/IP), where the messages are not processed further. The message transfer is from main memory to main memory, so there are no disk accesses. In the case of real applications, their CPU and disk requirements must be taken into account, as must the fact that networks might not be optimized.

4.5.1 LAN connection for S servers

BS2000 systems are connected to a LAN (Local Area Network) via an HNC (High-speed Net Connect) channel adapter.

HNC

The High Speed Net Connect (HNC for short) is a high-performance network connection for BS2000 systems on S servers, see the “HNC” manual [13].

BS2000/OSD V9.0 operates the following HNC models. They are always supplied with a Linux operating system.

- HNC-IV 91853 (HNC-IV for short)
  Based on a TX300-S3 Primergy server.
  Connection via a type S channel is supported for the last time with this HNC.
- HNC-V 91854 (HNC-V for short)
  Based on a TX300-S6 Primergy server.
  Only the connection via a type FC channel is supported with HNC-V and higher.
  In addition, the following new functions are offered in HNC-V and higher:
  – Checksum offload (see page 90)
  – Link aggregation (see page 90)
  – Support of Net-Storage (see the “Introductory Guide to Systems Support” [10])
- HNC-VI 91855 (HNC-VI for short)
  Based on a TX300-S7 Primergy server.
  Hardware and software update without new functions.

The HNC supports connection to an S server via a type S channel (HNC-IV) and via a type FC channel. It supports Fast Ethernet and Gigabit Ethernet LANs. The high throughput rate of the Gigabit LAN requires the HNC to be connected to the server via a type FC channel.

The information below relates solely to the connection via the type FC channel. Connection of the HNC-IV via a type S channel which is no longer described here limits the throughput to around 12 Mbyte/s, while the FC connection (1 Gbit/s) and Gigabit Ethernet enable up to 7 times the throughput to be achieved.

The transaction rates and throughput values in the following sections were measured with the HNC-V. They are compared with the measured values of the predecessor model HNC-IV.

Gigabit Ethernet LANs (1,000 Mbit/s) are examined. A distinction is made here between the default settings with a Maximum Transmission Unit (MTU) of 1,500 bytes and jumbo frames with an MTU of 9,000 bytes.

In the case of jumbo frames, the network, switches and partner systems must also be configured for jumbo frames.

HNC at type FC channel

The table below shows the measurement values for an S server with 4 CPUs and a performance of 1,060 RPF which is connected to a Gigabit LAN with an MTU of 1,500 or 9,000 bytes via an HNC.

HNC-IV and HNC-V Gigabit Ethernet (1,000 Mbit/s)				
Channel connection	1 type FC channel			
MTU [bytes]	1,500		9,000	
Transactions [TA/s] CPU [%]	88,000 96		84,000 96	
	Send	Receive	Send	Receive
Throughput [Mbyte/s] CPU [%]	60 21	75 26	65 14	78 14

The maximum transaction rate is achieved by alternately sending and receiving short messages (10 bytes). The server on which the measurements were made was almost fully utilized solely by communication.
The maximum throughput is achieved with unilateral sending or receiving of long messages (8 KB or more).

An MTU of 9,000 bytes reduces the transaction by approx. 5% and increases the throughput by approx. 8% compared to an MTU of 1,500 bytes.

Comparison of the HNC-IV and the HNC-IV

The values for the maximum throughput and the maximum I/O rates which can be achieved are identical on HNC-V and HNC-IV.

New functions with HNC-V and openNet Server V3.4 and higher (see the “BCAM” manual [2])

- Checksum offload
  This function offloads the checksum calculation function from BS2000/OSD to the HNC (which has a low utilization rate). When working with long messages, this reduces the CPU requirement for the network workload in BS2000/OSD. The maximum throughput remains unchanged.
  This function is used by default in HNC-V and higher and openNet Server V3.4 and higher.

  When long messages are employed, the CPU requirement of BS2000/OSD can thus be significantly reduced:
  – with standard frames (MTU 1,500): by 10-15%
  – with jumbo frames (MTU 9,000): approx. 15% when sending and approx. 35% when receiving

- Link aggregation
  This function enables multiple virtual channel adapters (VCAs) to be bundled over one or more physical FC connections and operated as a single device. The aim is to increase the network performance (see the “BCAM” manual [2]).

  When this function is used, it must be borne in mind that above all when sending messages additional CPU performance of the BS2000 server is required. The higher the level of bundling, the more the CPU utilization for the network load increases.

  In a configuration in which the VCAs of two type FC channels are bundled, the Gigabit Ethernet could be fully utilized. Without this function the utilization of the Ethernet was 50 - 60%.

  By switching from bundling two type FC channels each with two VCAs to bundling two type FC channels each with four VCAs, the CPU requirement in BS2000/OSD increases by approx. 30%. You are therefore recommended only to bundle the necessary VCAs.

Throughput with one connection

The following values are possible with one (TCP) connection to a high-performance partner system:

LAN	Throughput (Mbyte/s)
Channel connection	Type FC
Gigabit Ethernet (Ethernet Frames)	60 - 75
Gigabit Ethernet (Jumbo Frames)	65 - 80

These values are not attained for communication with less powerful partner systems, nor are they possible when other components such as disks or a less than ideal network slow data transfer down.

The best possible response time between a BS2000 system and a partner system connected via Gigabit Ethernet is 0.5 milliseconds for short messages.

Formula for computing power

A rule of thumb for the computing power required for communication alone as of the socket interface (without message processing) is as follows for Fast Ethernet and Gigabit Ethernet LAN:

- MTU 1,500 bytes
  - 60 - 90 TA/s per RPF
  - 0.10 - 0.20 Mbyte/s per RPF
- MTU 9,000 bytes
  - 60 - 80 TA/s per RPF
  - 0.20 - 0.30 Mbyte/s per RPF

4.5.2 LAN connection for SQ servers

SQ servers are connected to a LAN by means of an integrated controller (PCI controller). Like all I/O operations, the I/O operations to the integrated controller are routed via the IO processor (see section “I/O processors (X2000)” on page 36).

The table below shows the measurement results for a Gigabit Ethernet with an MTU of 1,500 or 9,000 bytes. The maximum values specified are achieved using multiple parallel (TCP) connections. The values for the two SQ servers SQ200-80E and SQ210-160F, each with openNet Server V3.5, are shown.

SQ200-80E	**Gigabit Ethernet (1,000 Mbit/s)**			
MTU [bytes]	1,500		9,000	
Transactions [TA/s] CPU [%] IOP [%]	33,000 100 5		33,000 100 5	
	Send 16 KB	Receive	Send 16 KB	Receive
Throughput [Mbyte/s] CPU [%] IOP [%]	120 22 3	120 16 4	115 30 5	115 21 5

SQ210-160F	**Gigabit Ethernet (1,000 Mbit/s)**			
MTU [bytes]	1,500		9,000	
Transactions [TA/s] CPU [%] IOP [%]	88,500 98 17		87,500 98 17	
	Send	Receive	Send	Receive
Throughput [Mbyte/s] CPU [%] IOP [%]	116 20 14	116 10 13	120 13 13	121 7 13

The constraints for determining the maximum transaction rate and the maximum throughput are described in section “HNC at type FC channel” on page 89.

The maximum transaction rates are appreciably higher than all current normal requirements of, e.g. OLTP applications. The transaction rate is normally limited by full utilization of the BS2000 CPUs.

Sufficient CPU capacity is required to fully utilize high-performance networks. Provided the limits of the LAN have not been reached, the performance that can be achieved increases with the number of CPUs.

The data throughput when sending messages depends on the message length used. The following always applies: the longer a message, the higher the throughput.
The following throughput is possible with only one connection (message lengths from 8 KB through 64 KB were measured):

	Throughput (Mbyte/s)			
	SQ200-80E		SQ210-160F	
LAN	**Send**	**Receive**	**Send**	**Receive**
Gigabit Ethernet / Ethernet Frames	62 - 100	115	49 - 100	115
Gigabit Ethernet / Jumbo Frames	62 - 100	119	71 - 100	120

These values can only be achieved with powerful partner systems and an optimum network configuration.
With just one task it is possible to reach the maximum network throughput of 120 Mbyte/s when receiving messages.

Homogeneous connection

The values below are reached when two virtual machines (each with 4 CPUs) are connected to an SQ200-80E server via LOCLAN with a standard MTU of 65,512 bytes:

- The maximum transaction rate is approx. 12,000 TA/s.
  The guest systems' BS2000 CPUs are then utilized to 91 - 98%.
  The minimum response time is approx. 0.7 ms.
- The maximum throughput when sending and when receiving 64 KB messages is then over 500 Mbyte/s. In this case the BS2000 CPUs are utilized to approx. 80%.

4.5.3 Resources for performance control

BCMON

Using the BCMON command, you start cyclic BCAM monitoring and output the desired values at regular intervals. The values are displayed on the console and also written to the log file (`$SYSAUDIT.SYS.CONSLOG.<date>.<counter>`), so that the values can be evaluated later. See also the “BCAM” manual [2].

Simple example for BCMON

This example shows how the throughput of a specific connection is measured. The total throughput can be measured by not specifying `LINE=`.

`/BCMON MODE=ON,RECORD=L2,LINE=LINEFCGB,SEC=30` (command on the console)

Every 30 seconds the measured values accrued in this time are output, e.g.:

```
<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 of the line
`BYTES=(I=` : Number of bytes received
`BYTES=(O=` : Number of bytes sent
`I/O'S=(I=` : Number of inputs
`I/O'S=(O=` : Number of outputs
`PACKETING=(I=` : Number of messages per input
`PACKETING=(O=` : Number of messages per output

(2) `Line LINEFCGB` : Name of the line
`UNICASTs=(I=` : Number of UNICASTs received
`UNICASTs=(O=` : Number of UNICASTs sent
`MULTICASTs=(I=` : Number of MULTICASTs received
`MULTICASTs=(O=` : Number of MULTICASTs sent

(3) Message `BCA0B17` is merely an error counter

NETSTAT

The NETSTAT program can be used to request information about applications, connections, routing data and network interfaces. The functionality of the NETSTAT program is also available via BCAM command SHOW-NET-STATISTICS/NETSTAT. The NETSTAT program can be used under any user ID without any privileges with the SHOW-NET-STATISTICS/NETSTAT command. See also the “BCAM” manual [2].

Simple example for NETSTAT

This example shows how the total throughput is measured using NETSTAT.

/netstat interface-sum-rate=*yes,waittime=30 (user command)

After the load messages and a display of all the values accrued to date, the standardized values for the inputs and outputs are output every 30 seconds (in each case in packets and bytes):

```
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

The following SM2 monitoring programs supply important information on networking (see the “openSM2” manual [18]):

- BCAM-CONNECTION
  Monitored data on connection sets
- CHANNEL-IO
  Monitored data on the channel load and channel transfer rates.
  When an HNC-IV 91853 is connected, the utilization of the type FC channel can be measured using SM2 resources.
- PERIODIC TASK
  Monitored data on tasks: utilization of the BCAM tasks (e.g. BCAM and BCA0)
- RESPONSETIME
  Monitored data on response times, think times, transactions times, and wait times in the BCAM pool
- SAMPLING-DEVICE
  Monitored data on I/Os, data volume and utilization of devices
- TCP-IP
  Monitored data on TCP/IP connections

5 Data backup

The performance capability of online and nearline peripherals has constantly increased in recent years. As a result, Fibre Channel connections from the server to the online and nearline peripherals have become standard for open server systems.

On S servers (type FC channel) and SQ servers the Fibre Channel is the standard connection technology for online and nearline peripherals.

Balanced and combined application of online and nearline peripheral is required for optimum performance in data backup. The configuration of such systems must be geared to the scope of the backup and the time windows available for data backup. This section provides basic information on this subject.

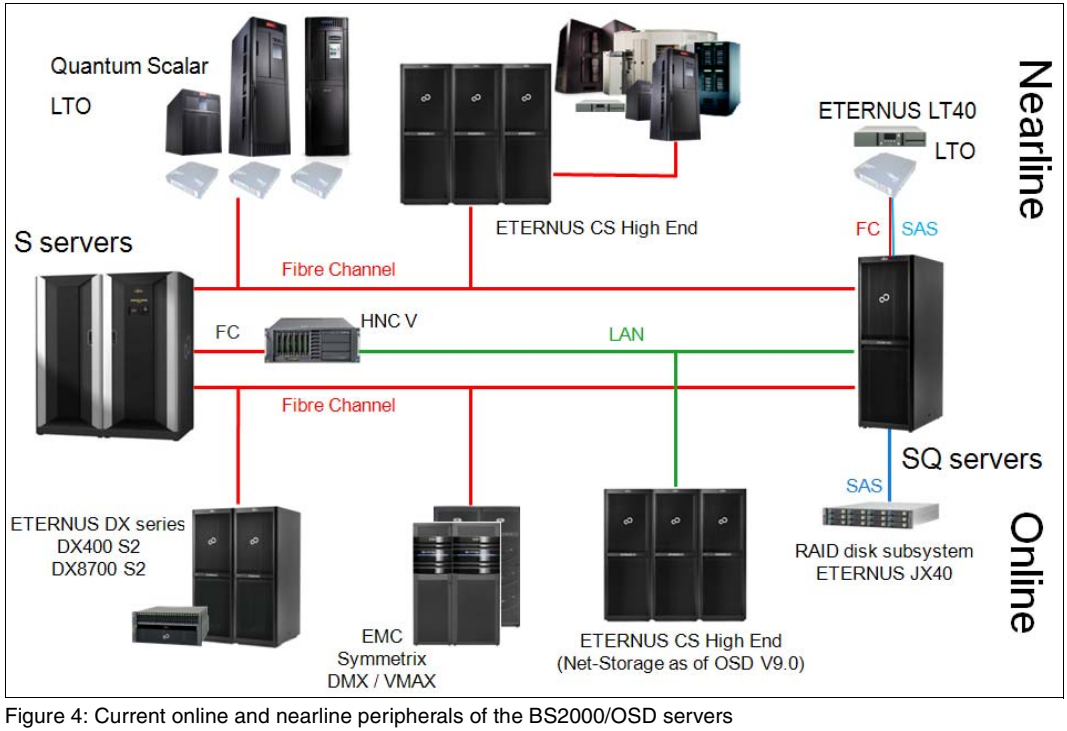


Figure 4: Current online and nearline peripherals of the BS2000/OSD servers

Forms of data backup

There are two forms of data backup:

- In the case of **logical** data backup files and job variables are read from one or more volumes and written contiguously to a backup file on one or more volumes (e.g. tape cartridges).
  Individual files and job variables are saved individually and can be restored individually.
  The software products HSMS and ARCHIVE are available in BS2000/OSD for logical data backup (see the “HSMS” [14] and the “ARCHIVE” manuals [1]).
- In the case of **physical** data backup no individual files are saved but entire volumes (individual volumes or entire pubsets). Here all the data on a data medium, including the data labels, are written block-by-block in their physical order onto a second data medium (e.g. tape cartridge).
  The software product FDDRL is available in BS2000/OSD for physical data backup (see the “FDDRL” manual [11]).

Backup volumes

The backup volume for **logical** data backup depends on the application profile in the Data Center. For example, the following values are available for the various backup types and quantities of backed-up data:

- Full backup: complete active data set
- Incremental backup:
  daily e.g. 5% of the complete data set
  weekly e.g. 15% of the complete data set
  In an incremental backup not the difference to the last full backup is formed but, for each file, the difference to all previous backups in.

The number of backup media required depends on the volume of the full and incremental backups and must be determined separately for each application.

In the case of **physical** data backup of a volume FDDRL by default transfers (DUMP function) only the areas marked as occupied in the F5 label. As a rule, therefore, in a physical backup with FDDRL no more data is transferred than in a logical backup with HSMS. In the case of physical backups, however, neither incremental backups can be created nor can individual files be restored.

5.1 Logical data backup with HSMS/ARCHIVE

BS2000 files, POSIX files and job variables of the local BS2000 system are saved using HSMS. BS2000 files and job variables can be saved with ARCHIVE.

In addition to data backup, data restoration must also be borne in mind. With restoration of the saved data (rare in comparison with backup), the throughput values are a little different than those for backup. However, they are usually only critical if a full backup is to be loaded. With partial restoration, which runs in parallel with normal operation, the throughput plays a subordinate role.

Parallelism

HSMS enables multiple jobs to be performed in parallel. The degree of parallelism can be specified by the number of HSMS server tasks.

In turn, parallel data streams (to multiple tape devices) can be used in a job. This is controlled using the `PARALLEL-RUNS` operand in the action statement or an archive attribute. Under ARCHIVE this is controlled using the `DRIVES` operand in the `SAVE` statement.

HSMS/ARCHIVE also generates asynchronous I/Os when reading from disk (save) or writing to disk (restore).
These I/O operations are executed in parallel when the disk storage system's Parallel Access Volume (PAV) function is used together with a suitable RAID level (see page 55).

Parallel data streams and multiplexing result in shorter backup times.

HSMS/ARCHIVE directory

The HSMS or ARCHIVE directory is an ISAM file in which all information on the data sets saved and the associated volumes is stored.

When a large number of files are to be saved and there are a lot of backups, the directory is correspondingly large. When they are backed up the file sets that have been saved are written to the directory, and read from there when they are restored. Furthermore, in the case of an incremental backup, the system must also ascertain whether the current file set is already contained in the backup copy. Accesses to the directory take place in the ARCHIVE main task parallel to the actual file backup in the subtasks.

Backup jobs which relate to the same directory or same HSMS archive cannot run in parallel.

File sizes

Logical data backup is based on files. A certain basic effort is required for each file. The file sizes consequently have a major influence on throughput when HSMS/ARCHIVE is used for logical data backup. The throughput can be significantly lower with a large number of small files than in the ideal case with a small number of very large files.

Large tape blocks

Using large tape blocks increases the performance and tape utilization. By default, 256 KB tape blocks are used when saving to tape cartridge using HSMS and ARCHIVE. To permit this, the value `BIG` is predefined in the ARCHIVE parameter file `BLOCK-SIZE-T-C`. It may then be necessary to specify `BLOCKING-FACTOR=*STD / *MAX` in the HSMS statements and archive definitions.

LTO drives are always written to using a block size of 256 KB irrespective of predefined parameters and archive definitions.

5.2 Performance measures in backup operation (HSMS)

Various options which the user can use to achieve better performance for logical data backup using HSMS/ARCHIVE are described here. The main objective is to reduce the backup time.

The measures presented can be used individually and independently of each other. The measures are always beneficial in all situations. No situation analysis is required.

It is not possible to specify the improvement in performance for each measure precisely. The general situation and usage type (e.g. file size) must be taken into consideration. All measures (with the exception of 4.) can be tested simply with a regular or time-critical backup job. The improvement in performance is shown by comparing the backup times with and without the measure.

The measures are primarily examined in terms of their effect on backup operation. The disturbance to applications run in parallel is generally negligible. In the case of some measures (e.g. 3. and 5.) the backup time is reduced when the total number of inputs/outputs for the backup remains unchanged. This results in an increased I/O load per time unit.

1. Limiting the scope of the backup through file selection and the incremental process

   Using the FROM-/EXCEPT-FILE parameters (positive and negative specification with partial qualification) is recommended.

   A maximum backup class can be predefined: only files whose class is less than or equal to the specified backup class are saved (as a file attribute with the default backup class = A).

   Temporary files and paging files are never saved.

   Management classes in SM pubsets (freely definable file attribute for group formation in SM pubsets) can be used for file selection when backup and migration take place.

   You can select whether migrated files should also be backed up (backup operand SAVE-DATA). By default only catalog entries of migrated files are also backed up.

   Two processes are available for incremental backups between the periodical full backups:

   - MODIFIED: Save files only when these do not already exist in the archive
   - BY-CATALOG-MODIFICATION: Save files if these were modified only after a specified date or after the date of the last backup for the same archive

2. Sensible distribution of archives or archive directories
   - Decentralized organization (archives per pubset) also for large SF pubsets
3. Reducing the backup time through parallel operation (increased I/O load!)

   Executing multiple HSMS tasks (for various archives) simultaneously. The maximum number of parallel tasks can be set using the HSMS statement `//MODIFY-HSMS-PARAMETERS NUMBER-OF-SUBTASKS=`

   Several tape devices can be used by ARCHIVE subtasking for each task (specification of `PARALLEL-RUNS` either in the task or as an archive attribute).

   The cartridges with a high capacity should always use the Several-SVID-Format for continuing (CONTINUE) tapes. This is practically a prerequisite for parallel operation, otherwise the tape consumption is high.

   Increasing parallel operation reduces the runtime, and the number of inputs/outputs per second is increased.

   When a large number of tape devices are used (e.g. with ETERNUS CS HE), these can be used more in parallel. HSMS/ARCHIVE ensures efficient use of these devices.
4. Increasing throughput of disks and tapes

   A type FC channel connection should be preferred to a type S channel connection for both disk and tape storage systems because of the higher data rates.

   Disks on the type FC channel are always in FBA format. They can be operated with a block size of up to 480 KB (CKD format permits up to 72 KB).

   In order to reduce the number of tape I/Os the use of large tape blocks (256 KB instead of 32 KB to date) is recommended (`BLOCKING-FACTOR = *MAX` parameter in statements, as archive attribute or as global ARCHIVE parameter `MBK-SIZE = *BIG`)

   In ARCHIVE hardware compression from/to MTC devices is always used. Software compression can optionally be used for DMS backup to disk. However, compression as a software function has a detrimental effect on the CPU performance.

   In addition, K formatting should be dispensed with and NK2 format used as the maximum I/O length for FBA disks in K format is 160 KB and in NK2 format 480 KB.

   In practice parallel operation is restricted by the channel configuration rather than by the CPU performance.

   HSMS/ARCHIVE also places a load on the disk side through catalog accesses and the wastage of small files.

5. Multiplexing (generating parallel data streams)

   In BS2000 backup operation HSMS/ARCHIVE and the PAV (Parallel Access Volume) function enable multiple I/O operations to/from disk to be started in parallel. This permits the data streams between the tape and the disk to be multiplexed.

6. Reducing the downtime for online operation (application restart before the end of the actual backup)

   HSMS and its CCOPY (Concurrent Copy) function enable consistent backups to be created simultaneously and the data which is to be backed up to be processed in any required way. To permit this, the application which processes the files to be backed up must be terminated briefly before backup begins or be guided to a synchronization point. The backup job can then be started. After an initialization phase which can be monitored with a job variable, the application can be enabled again. Backup then runs parallel to file processing.

   Backup with CCOPY uses either a temporary scratch file or a replication function of the disk storage system. Further information on this subject is provided in the “HSMS” manual [14].

   Use of a replication function of the disk storage system permits load splitting between the application on the original disk and the backup on the mirror disk, and as a result operation of the application is disturbed considerably less by the backup I/Os.

   In these cases a task job variable shows the status for the application restart before the task is completed.

7. Performance of the catalog accesses in the case of DMS backups

   In shared pubset operation the normal backup requests are sent to the master side in HSMS to utilize the quicker local catalog accesses without MSCF traffic.

   Backup requests with CCOPY of mirror disks of the disk storage systems are always performed locally in shared pubset mode because the catalog accesses to the split mirror disks of the disk storage systems are performed locally. This enables the I/O load to be split (e.g. with two systems for application and backup).

   When a very large number of files are to be backed up, summary reports (only in the event of errors are entries made on a file-by-file basis) should be used instead of full reports (entries for all files) in order to keep the trailer time short.

8. Database backups in ongoing database operation

   Files which can be backed up but are open for writing can be backed up using the backup option `SAVE-ONLINE-FILES = *YES`, e.g. for UDS databases with log files and for Oracle database files or single tablespaces.

   SESAM database backup by SESAM/SQL via the HSMS program interface is performed in the same way, also in combination with CCOPY of mirror disks of the disk storage systems. This results in minimal refreshment phases for the database.

5.3 Physical data backup with FDDRL

With FDDRL, volumes are backed up on the basis of their physical block structure and therefore without catalog access. FDDRL supports public volumes (SF and SM pubsets) and private disks, as well as (under special conditions) volumes of third-party systems.

Various options are available for backing up pubsets:

- Backing up an exported pubset
- Backing up a split disk mirror (when using replication functions, see page 50)
- Backing up the home pubset
  (this can also be backed up online using FDDRL V18.0 or higher)

Parallelism

FDDRL allows the backup and restoration of multiple volumes in parallel and independently of each other under separate tasks (FDDRL subtasks) to a corresponding number of (possibly virtual) tapes. The degree of parallelization can be specified with the FDDRL parameter `TASK-LIMIT`.

With FDDRL asynchronous I/O operations are also generated when reading from disk (DUMP) or writing to disk (RELOAD).
These I/O operations are executed in parallel when the disk storage system's Parallel Access Volume (PAV, see page 55) function is used together with a suitable RAID level.

RUN-PRIORITY

The task priority for the FDDRL subtasks is specified with the FDDRL parameter `RUN-PRIORITY` in the context of the job class. This permits the priority of FDDRL processing to be controlled and the throughput to be influenced:

- Aim: during low-load BS2000 operation an FDDRL run is to be performed as quickly as possible and without regard for the response time behavior.
  In this case the `TASK-LIMIT` parameter should have the value of the available tape devices (up to 16) in order to achieve a good throughput due to the high degree of parallelization. The `RUN-PRIORITY` parameter should be set to “high”.
- Aim: during normal BS2000 operation an FDDRL run is to be performed without impairing the performance of the main application.
  In this case the `TASK-LIMIT` parameter should be set to a relatively low value to take into account the other applications. In the event of multiple parallel runs a noticeable worsening of the response time behavior of the other applications must be expected.
  The `RUN-PRIORITY` parameter should be set to “low”.

Tape format

In order to make allowance for the fact that new tape storage systems are continually being developed, by default FDDRL works with the largest possible I/Os and “large tape blocks” of the same size (e.g. 160 KB for FBA).

In the case of very high-speed tape storage systems, FDDRL can read blocks from up to 4 disks in parallel and write alternately to tape. This process, known as “multiplexing”, makes sense particularly when disk processing is slower than tape processing. Depending on the tape storage systems available and the size of the tape volume used, the following backup strategies can be employed:

- Large number of tape drives and small tape volumes

  With ETERNUS CS HE a large number of tape drives are as a rule available. The volume size can be configured and is normally set significantly lower than when LTO cartridges are used. Here it is recommendable to use the FDDRL statement `//DUMP-PUBSET ..., SAVE-ENTITY=*SINGLE-DISK` to back up pubsets.

  One tape volume is written for each disk volume.

  This ensures that maximum performance is achieved. The tape and disk sizes should be coordinated to achieve a good tape occupancy level.

- Few tape drives and large tape volumes

  When, for example, LTO drives are used, the following FDDRL statement is recommended: `//DUMP-PUBSET ..., SAVE-ENTITY=*DISK-SET(NUMBER-OF-DISK-SETS=n)`

  Multiplexing enables multiple disk volumes to be backed up on a small number of tape volumes and tape drives. The number of drives used in parallel can be set using the `NUMBER-OF-DISK-SETS` parameter. The `TASK-LIMIT` must be coordinated with this using `//MODIFY-FDDRL-PARAMETERS`. Since multiple disk volumes are backed up on one tape volume, large tape volumes can be used effectively.

  The creation of disk sets using `//DUMP-PUBSET` also has the advantage that complete and consistent restoration of the pubset is possible with `//RELOAD-PUBSET`. If disk sets are not used, the pubset must be restored with `//RELOAD-DISK`. In this case the system administrator has the task of guaranteeing the completeness and consistency of the various restored volumes.

  In addition to the operand values mentioned, `TAPE-FORMAT=*STD` should also be set (`//MODIFY-FDDRL-PARAMETERS`). FDDRL then determines the ideal tape format itself. A different format specification is only provided for data interchange with systems using earlier FDDRL versions.

5.4 Components for data backup

The following components are relevant to performance when local data backup takes place:

- Software: HSMS, ARCHIVE or FDDRL
- BS2000 CPU
- Tape storage system including the connection to the server
- Disk storage system including the connection to the server

The performance of some subcomponents is examined in this section. This enables an approximate estimate to be made of the anticipated throughput rates for data backup and restoration in concrete configurations.

The backup performance is determined by the lowest throughput value of the subcomponents involved.

The throughput rates specified relate to one task during the actual backup phase, in other words without mounting the tape and without generating the report files or starting and stopping the programs.

The measurement results in this section assume favorable conditions, e.g. no parallel load on the measured configuration, sufficient CPU performance.

5.4.1 BS2000 CPU

In the case of local data backup the following approximate values must be assumed for the CPU requirement in BS2000/OSD:

- Data backup with HSMS depending on the file size:
  - on S servers: 1.3 (large/medium-sized files) - 2.3 (small files) RPF per Mbyte/s
  - on SQ servers: 0.7 - 1.5 RPF per Mbyte/s
- Data backup with FDDRL: below 0.2 RPF per Mbyte/s

Particularly in multiprocessor systems it must be borne in mind that the CPU performance required for a backup task must be provided by just one BS2000 CPU.

The I/O processors on S and SQ servers generally have no influence which restricts the performance.

5.4.2 Connection to the server

The following theoretical performance rates apply for the connections to the server:

- Type FC channel (S server): 1 Gbit/s
- FC connection (SQ server): 1 to 8 Gbit/s

Disk storage systems are normally connected via more than one path. If the options of parallelization (suitable RAID groups, PAV, multiplexing) are used in the disk storage system, the degree of connection utilization can be extremely high in the case of throughput-oriented data backup loads.

High-speed tape storage systems should be connected to high-speed type FC channels or FC connections. These may not be used together with disk storage systems.
The channels of the S servers to the tape storage systems can be almost fully utilized by just a few devices when data backup takes place.

5.4.3 Tape storage systems

Virtual archive system ETERNUS CS HE

With ETERNUS CS HE a virtual archive system is placed in front of a real archive system with the physical drives and cartridges. While the data backup application on the server continues to assume that it is working with MTC drives, the virtual archive system stores the data temporarily on magnetic disk (Tape Volume Cache). This enables the parallelism of the data backup to be increased without increasing the number of real MTC drives. Only when all the data has been transferred to the virtual MTC device and after the cartridge has been dismounted is the data stored physically on an MTC. Given these advantages, ETERNUS CS HE is urgently recommended for nearline operation in the BS2000 environment.

The performance data specified below relates to CentricStor V4.0 (CentricStor is the predecessor of ETERNUS CS HE). However, the recommendations derived from this also apply for the current versions of ETERNUS CS HE.

Scalar MTC archive systems with LTO technology

LTO devices of the real Scalar MTC archive systems from Quantum Corp. can be connected directly to the BS2000/OSD server either parallel or as an alternative to ETERNUS CS HE. Information on the MTC archive systems is available in the “ROBAR” manual ([24]).

In contrast to ETERNUS CS HE, the parallelism is limited by the number of MTC drives.

MTC archive systems to SQ servers

An ETERNUS LT40 (SQ200) or ETERNUS LT40 S2 (SQ210) MTC archive system with one or two drives is optionally available for tape automation on SQ servers.

Throughput and compression with LTO drives

The information below relates to the direct connection of LTO devices to BS2000/OSD servers (without ETERNUS CS HE).

The data written from the server (e.g. a BS2000 system) is initially transferred to the LTO device's cache. Here it is compressed as far as possible by the LTO device. The compressed data is then written to the physical tape in the LTO device. An analogous procedure is used for reading data.

The following performance values apply for the LTO devices:

Device Type	Maximum Throughput	Capacity
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

These performance values relate to the device / the physical cartridges.

A distinction must be made to the following data here:

Device Type	Throughput (compressed)	Capacity (compressed)
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

This data relates to the data transfer between the server and the LTO device, i.e. to the time before compression (writing) or after decompression (reading). A compression factor of 2 is assumed. In BS2000/OSD, compression factors of 2 to 3 are typical. This consequently results in the following throughput for BS2000/OSD:

- LTO-2: 70 to over 100 Mbytes/s
- LTO-3: 160 to over 240 Mbytes/s
- LTO-4: 240 to over 360 Mbytes/s
- LTO-5: 280 to over 420 Mbytes/s

Such a throughput is only possible in BS2000/OSD to some extent and is only rarely required.

In the case of S servers the throughput with LTO-3 or higher is limited by the type FC channel (1 Gbit/s). Consequently there are no major changes on S servers either with regard to the throughput or the streaming behavior. Considerably better values have been achieved here on SQ servers.

On the hardware side LTO-4 and LTO-5 drives support “tape encryption”. According to vendor information, when tape encryption is enabled the data rate is reduced by only approx. 1%.

Streaming limit for LTO drives

LTO devices work best when the data is transferred to the tape as evenly as possible at the maximum throughput rate.

When data is written from the server (taking into account the compression rate) at a data rate which is below the maximum throughput rate, the LTO device adapts to this lower data rate (rate adaption). This enables load fluctuations to be compensated for. The lower limit for rate adaption is the “streaming limit” of the LTO device. The streaming limit is device- and vendor-dependent and is around

- 15 Mbyte/s for LTO-2
- 30 Mbyte/s for LTO-3, LTO-4 and LTO-5

If the data rate falls below the streaming limit, this results in “start/stop operation”. In this case the tape and drive are subject to increased mechanical loads which can reduce their service lives. Lengthy start/stop operation should therefore be avoided. Brief periods of start/stop operation when mounting or unmounting tapes cannot be avoided.

Choosing the correct cartridge determines the quality and durability: the cartridges and the drive must suit each other. You are urgently recommended only to use qualified cartridges labeled “Preferred Quality”.

When the compression factor is 2 the server must therefore transfer 30 Mbyte/s with LTO-2, and 60 Mbyte/s with LTO-3, LTO-4 and LTO-5 in order to exceed the streaming limit. If this is not achieved with slow servers, in BS2000/OSD the IORM utility routine can be used to disable compression on the LTO drive in order to prevent possible damage to the tape. Depending on the current transfer rate, IORM also enables compression to be enabled and disabled dynamically. When compression is disabled, the throughput from BS2000/OSD to the LTO device and on to the drive is the same.

Performance of the tape device systems

For data backup the data is written onto one or more tape storage systems and read from these when the data is restored.
This section examines the performance of tape storage systems in isolation, in other words without the other components for data backup.

The performance of newer tape storage systems can only be fully utilized with type FC channels (S server) or FC connections. When the tape storage systems are connected via a type S channel, the maximum throughput which can be achieved is 15 Mbyte/s.

In BS2000/OSD the “large tape blocks” (256 KB instead of 32 KB per I/O) enable the performance to be increased by a factor of 2.

The tables below show the possible throughput of various tape storage systems. The load corresponds to data backup using HSMS/ARCHIVE or FDDRL with large tape blocks. The I/Os were performed only from the BS2000 main memory (not from disk) to the MTC device.

The throughput specified is for data which is difficult to compress (compression factor 1) and for data which can be compressed normally (factor 2-3).

S server, connection via type FC channel (1 Gbit/s):

		Throughput (Mbyte/s)			
Compression factor		1		2 - 3	
System	**Devices**	Write	Read	Write	Read
Scalar i2000	LTO-3	80	60	80	60
Scalar i500	LTO-3	80	80	95	95
Scalar i6000	as of LTO-3	80	80	95	95
ETERNUS CS HE	VTA	90	75	90	75

The normal limit of the type FC channel is almost 100 Mbyte/s.

Use of LTO-4 or LTO-5 leads to no further enhancement of the performance.

With their faster PCI controllers, SQ servers achieve a throughput of 200 Mbyte/s or more.

5.4.4 Disk storage systems

Characteristics of the data backup load

For data backup the data is read and written sequentially. With HSMS/ARCHIVE and FDDRL this is done in multiple parallel, asynchronous data streams from and to the disk storage system. Normally very large quantities of data are transferred. The caches of the disk storage systems reach the limits of their capacity here. Consequently read ahead strategies must be used to read from the physical disks, and delayed fast writes to write to the physical disks. The performance of the physical disks and of the RAID groups is therefore decisive for the throughput which can be achieved. With a data backup load, today's disk storage systems also achieve a very good throughput rate with read-ahead or Delayed Fast Write.

Performance of disk storage systems

For data backup the data is read from one or more disk storage systems, and written to these when the data is restored.
This section examines the performance of disk storage systems in isolation, in other words without the other components for data backup.

Newer disk storage systems are connected to the server via type FC channels or an FC connection. This permits full use to be made of their performance capability.
When the disk storage systems are connected via a type S channel, the maximum throughput which can be achieved is 8 - 12 Mbyte/s.

In order to achieve high throughput rates, not only a multipath connection via a type FC channel must be provided, but also suitable RAID levels (see section “Properties of external disk storage systems” on page 45). RAID level RAID1/0, RAID5 or RAID6 offers the best throughput for data backup here.
RAID1 is particularly ill-suited to achieving a high throughput in the case of loads with medium to high parallelism.

The table below shows the possible throughput with different settings. The measured values reflect both the ideal case with a small number of large files (1000 Mbyte). They also show the reduction in throughput when medium-sized (10 Mbyte) or small (1 Mbyte) files are used instead of the optimum very large size.

Reading takes place here only from the disk storage system to BS2000 main memory without writing to tape. The disk storage system has a multipath connection via a type FC channel or an FC connection.

S server, ETERNUS DX or Symmetrix DMX/VMAX disk storage system, connection via type FC channel (1 Gbit/s, 4-path

RAID level	**Read throughput (in Mbyte/s)**					
File size	1000 Mbyte		10 Mbyte		1 Mbyte	
PAV (1 alias)	w/o PAV	with PAV	w/o PAV	with PAV	w/o PAV	with PAV
RAID1	70	75	50	70	35	40
RAID1/0	75	140	70	130	60	90
RAID5	75	130	70	125	50	90

In the case of data backup, RAID groups with “data striping” (RAID1/0, RAID5, RAID6) have a considerably higher throughput than RAID1. With large files it generally significantly exceeds the throughput achieved on the tape side.

The throughput loss with medium-sized files (10 Mbyte) is low. It only becomes significant below the limit of approx. 1 Mbyte.

5.5 Recommendations

Whenever data is backed up you should ensure that the performance of the tape and disk storage units used is balanced. This section provides recommendations for achieving as high a throughput as possible during data backup.

The general recommendations for the peripherals (see chapter “Performance behavior of the peripherals” on page 45) also apply for data backup.

BS2000/OSD

- S servers: use “Dynamic PAV” (DPAV)
- Use of Fibre Channel connection technology (on S servers: type FC channel, not type S channel)

Tape storage systems

- Use of ETERNUS CS HE
- If high-speed LTO tape devices (LTO-3 and higher) are to be connected directly to the server:
  - tape material: use only qualified cartridges (“preferred quality”)
  - ideal operation above the streaming limit in order to prevent start/stop operation; occasionally falling below the streaming limit for brief periods can be tolerated

Disk storage systems

- Use of state-of-the-art disk storage systems
- Use of high-speed disks
- As large a cache as possible
  This only plays a minor role in data backup, but is very important in normal operation. The size selected should ensure that the read hit rates in OLTP mode are 60 - 80%.
- Use of RAID level RAID1/0, RAID5 or RAID6
- Take into account the requirements for configuration from the BS2000/OSD viewpoint
- PAV and RAID1/0, RAID5 or RAID6 also enable “large” volumes (more than 32 Gbyte in size) to be used
- Use of FBA disks of the type D3435 with data format NK2

6 Performance aspects in VM2000 operation

VM2000 permits simultaneous execution of multiple, self-contained BS2000 guest systems on a real S or SQ server. The individual guest systems run on virtual machines (VMs) and, as far as their functions are concerned, operate in the same way as in native mode.

The resources CPU, main memory and devices are assigned to the virtual machines and thus to the guest systems by VM2000.

A hypervisor controls the execution of the guest systems on the VMs. In particular it virtualizes the global resources CPU and main memory and starts execution of the operational guest system's CPUs on the real CPUs (scheduling).

- On S servers the VM2000 hypervisor is a separate load module of VM2000 which is loaded (automatically) when VM2000 operation is initialized.
- On SQ servers the Xen hypervisor performs this role. Some of the hypervisor tasks are performed by the carrier system X2000.

VM2000 on S servers distinguishes two processor statuses:

- Hypervisor mode (the VM2000 hypervisor runs in this status)
- VM mode (the guest systems run in this status)

A context (a set of registers containing the states of the CPU and information specific to VM) is associated with each of these statuses. This context is loaded when the status is activated.

In contrast to native mode VM2000 overhead arises:

- Hypervisor overhead (program execution in hypervisor mode)
- Indirect overhead (reduced server performance through changing VM contexts)

With careful configuration the VM2000 overhead is between 5% and 12% of the server performance. How high the VM2000 overhead on a server will actually be depends on the VM2000 configuration. Refer to the information in the section "Optimum configuration of VM2000" on page 132 to keep the VM2000 overhead to a minimum.

The following sections provide information on the ideal structure for VM2000 operation. Refer to the "VM2000" manual [33] for detailed information on VM2000.

6.1 Managing the CPU performance with VM2000

Examples illustrating the topics in this section are provided in the section “CPU management” of the “VM2000” manual [33].

6.1.1 VM2000 scheduling (S servers)

When a VM is scheduled under VM2000 the VM2000 hypervisor starts a loadable virtual CPU of a VM on a free real CPU. VM2000 uses two different processes for scheduling at VM runtime:

- CPU assignment in time slices (see below)
- Fixed CPU assignment (dedicated CPUs, see page 120)

Here the CPU performance of the server is distributed ideally across the loadable virtual machines in accordance with the settings for the VMs.

On SQ servers the Xen hypervisor performs scheduling. To do this it uses a time slicing procedure, but this is not influenced by VM2000. The VM-specific control factors of multiprocessor level, CPU quota and limiting of the CPU performance are also available here.

CPU assignment in time-slicing mode

Normally the number of attached real CPUs in a CPU pool is **less** than the sum of the attached virtual CPUs of all active VMs which are assigned to this CPU pool. VM2000 starts a virtual CPU on a real CPU from the pool using time slicing.

The maximum size of the time slice is defined dynamically for each virtual CPU of a VM in VM2000 in the range from 0.1 to 8.0 milliseconds. VMs with a “very small” CPU quota are then also assigned a smaller time slice. The maximum size of the time slice is only reached in few cases on the current BS2000/OSD servers. Generally a time slice is terminated by the virtual CPU of the VM entering the interruptible idle status.

The VM2000 administrator can influence the distribution of the CPU performance using VM-specific control parameters, see page 119. VM2000 also enhances the performance, e.g. by means of CPU affinity, see page 120.

The scheduling priority of the individual guest systems is derived from the specified CPU quotas and the CPU time used most recently.

VM-specific control parameters

The control parameters which are relevant to performance for a VM in time slicing are:

- Multiprocessor level of the VMs/guest systems (see page 129)
- CPU quota of the VM group, CPU quotas/member CPU quotas of the VMs (see "CPU share of a VM" below)
- Performance limit of the VM or VM group (MAX-CPU-UTILIZATION, see page 120)
- VM Privilege IO-PRIORITY (see page 127)

The time period in which a guest system can accept the CPU quota assigned to it depends to a very large extent on the degree of utilization of the VM2000 time slice, both that of the home guest system and of the other guest systems. This degree of utilization is high when the workload is CPU-intensive and low when the workload is I/O-intensive.

If a guest system does not fully utilize the VM2000 time slice assigned to it, this improves the I/O behavior of the other guest systems.
In the case of a multiprocessor VM, VM2000 attempts to distribute the unused share of a virtual CPU's CPU performance primarily to the other virtual CPUs of the VM.

However, if the time slice is used fully by a guest system, the I/O times can be considerably increased, above all for guest systems whose load consists almost exclusively of I/O-intensive tasks. The measures that can then be taken are:

- restriction of the CPU-intensive guest system by MAX-CPU-UTILIZATION (see page 120) and/or
- assignment of the privilege IO-PRIORITY (see page 127)

/SHOW-VM-STATUS INFORMATION=*SCHEDULE provides information on the mean value of a guest system's used time slice. Here the wait time up to activation of the virtual CPU is output as the measurement for the "increase of the VM".

CPU share of a VM

In the information commands VM2000 outputs various CPU quotas for a VM or VM group which enable the VM2000 administrator to observe the planned and current distribution of the CPU performance:

- CPU-Q (CPU quota /SHOW-VM-ATTRIBUTES/-RESOURCES)
  is the (MEMBER-)CPU-QUOTA of the VM set with /CREATE-VM or /MODIFY-VM-ATTRIBUTES

- EFF-Q (effective CPU quota, /SHOW-VM-ATTRIBUTES/-RESOURCES INFORMATION=*CPU)
  is the standardized CPU quota of the VM taking into account the constraints multiprocessor level, performance limit of the VM and CPU pools
  (sum of the standardized CPU quotas of all **configured** VMs = 100%).
  EFF-Q shows the CPU share of the VM or VM group to be expected in the long term in the event of a CPU-intensive load in all guest systems and when all real and virtual CPUs are connected.
- CUR-Q (current CPU quota, /SHOW-VM-STATUS)
  is the standardized CPU quota of the VM taking into account the constraints multiprocessor level, performance limit of the VM and CPU pools
  (sum of the standardized CPU quotas of all **active** VMs = 100%).
  CUR-Q shows the CPU share of the VM or VM group to be expected at present in the event of a CPU-intensive load in all guest systems and in relation to the real and virtual CPUs currently connected.

Performance limit of a VM

If a VM or VM group is never to be assigned more than a particular percentage of the server's CPU performance (not if CPUs are free, either), this can be defined using the MAX-CPU-UTILISATION parameter. Restricted CPU performance can, for example, be required in the context of service levels. The parameter enables excess CPU utilization by a guest system with a CPU-intensive load to be restricted.

CPU affinity

In time slicing VM2000 attempts to ensure with scheduling that a virtual CPU starts on the **same** real CPU in the event of the next scheduling procedure. CPU caches and initialization data of the VM are retained and reduce the indirect overhead. However, the primary aim is to optimize the response time.

Fixed CPU assignment (dedicated CPUs)

The number of attached real CPUs in a CPU pool is **greater than or equal to** the sum of the attached virtual CPUs of all active VMs which are assigned to this CPU pool. VM2000 assigns precisely one real CPU to each virtual CPU of a VM in this CPU pool.

In this case the VM2000 administrator can use the VM attribute VM-ACTIVE-IDLE to determine whether a VM still retains control over a real CPU if the VM's virtual CPU which is running on it is inactive (interruptible wait state, "idle"), see page 123.

6.1.2 VM groups (S server)

Multiple VMs can be combined in **VM groups**.

CPU utilization is then controlled by VM2000 in two stages. This strategy enables logically associated guest systems to be examined as a unit with regard to CPU utilization. For example, multiple VMs can be provided for a customer and be assigned a service level.

The following VM group parameters enable the VM2000 administrator to influence the distribution of the CPU performance:

- A VM group is prioritized vis-à-vis other VM groups or (individual) VMs in the same way as (individual) VMs, i.e. via the `CPU-QUOTA` and `MAX-CPU-UTILIZATION` parameters of the VM group.
- The `MEMBER-CPU-QUOTA` controls prioritization among the VMs which form the VM group. For the scheduling priority the `MEMBER-CPU-QUOTA` is qualified by the `CPU-QUOTA` of the VM group.
- In addition to the CPU utilization of the VM group the CPU utilization of each VM in a VM group can be limited using `MAX-CPU-UTILIZATION`.
- With VM groups VM2000 primarily attempts to distribute the unused share of the CPU performance of a VM to the other members of the VM group.

VM groups do not generate any additional VM2000 overhead.

6.1.3 CPU pools

Under VM2000, the real CPUs of a server which are available for VM2000 operation can be split into different, disjunctive **CPU pools**. This strategy enables greater hardware efficiency and CPU affinity to be achieved for servers with a large number of CPUs. A VM is assigned to precisely one CPU pool.

VM2000 attempts to assign the CPU performance of the real CPUs in a CPU pool to the individual guest systems assigned to a CPU pool in proportion to the CPU quotas. An upper limit results from the maximum CPU utilization of a VM.

The following characteristics of CPU pools have an influence on the server's performance:

- CPU pools reduce the VM2000 overhead especially with a larger number of real CPUs
- CPU pools permit a targeted but strictly limited performance (service level) for a set of VMs ("restricted VM group") with a further reduction in VM2000 overhead
- CPU pools on S servers enable a permanent CPU assignment (dedicated CPUs) to be created for critical productive systems.

Criteria for establishing CPU pools:

- The size of the CPU pools (the number of real CPUs assigned to a CPU pool) should be planned to ensure that the CPU performance made available corresponds to the sum of the planned share of performance for the VMs assigned to the CPU pool. As with the CPU utilization of the entire server, the CPU utilization of the individual CPU pools in OLTP mode should not be greater than 75%.
- The utilization of the server's real CPUs should be as even as possible. Consequently the utilization of the CPU pools should be as even as possible.
- In the case of CPU pools with just one real CPU it is to be expected that the dilation of the hardware service time (see page 127) is greater than with pools containing multiple real CPUs.

If the performance requirement of the guest systems changes in the course of a session, `/SWITCH-VM-CPU` can be used to add/remove CPUs to/from the CPU pool dynamically.

Criteria for assigning VMs to the CPU pools:

- In the case of guest systems with applications whose response time behavior is heavily dependent on the duration of the I/O operations, the contending VMs or guest systems in the same pool should be selected carefully so that the increase in the hardware service time does not have an unfavorable effect. For example, you should prevent a large number of CPU-intensive guest systems from being assigned to one CPU pool.
- In the case of guest systems with particularly high performance requirements the CPU pool can be equipped with enough real CPUs to ensure that a real CPU is available for each active virtual CPU. With respect to scheduling, VM2000 then works with a permanent CPU assignment on S servers.
- The `VM-ACTIVE-IDLE=*AT-DEDICATED-CPUS` parameter (S server) enables a VM with fixed CPU assignment to still retain control over a real CPU if the VM's virtual CPU which is running on it is inactive (interruptible wait state, "idle").
  This also prevents the changes of context when the guest system is in the wait state (idle). On S servers this allows a performance to be achieved which is almost equal to that in native mode.

  No evaluation of the VM-ACTIVE times (`/SHOW-VM-STATUS` or VM report of openSM2) is possible for VMs with `VM-ACTIVE-IDLE`.

When required, individual VMs or entire VM groups can be assigned dynamically to other CPU pools using `/ASSIGN-VM(-GROUP)-TO-CPU-POOL`.

6.2 Main memory

Each guest system has its own exclusive share of the server's main memory. Access to the main memory is carried out as efficiently as in native mode. The access times are the same.

The main memory distribution should be carefully planned. The main memory size required for a VM is initially estimated or derived from the existing system environment. After the paging rate has been measured using SM2 it may be necessary to make corrections. The paging rate should be low, in accordance with the load profile (TP- or dialog-oriented). The current main memory size of a VM (and on S servers also the location) can be determined using `/SHOW-VM-RESOURCES INFORMATION=*MEMORY`.

Because of the code expansion and (as of SQ200) because of the (resident) memory requirements for JIT buffers, the main memory requirement for a VM on an SQ server is greater than for one on an S server. On an SQ server, 40% of the main memory is by default reserved for JIT buffers.

The main memory size for the VM2000 hypervisor (S server) is calculated by VM2000 in accordance with the size of the peripherals.

The minimum main memory size (`MIN-MEMORY-SIZE`) for a VM should only be defined when the functions for main memory configuration (`/EXPAND- /REDUCE-VM-MEMORY`) are to be used. The size selected must then be enough to satisfy at least the resident memory requirements in the guest system.

The resident memory requirement depends on whether the software product DAB is used.

The JIT memory of the SQ servers (approx. 40% of the main memory) also resides in the resident memory. The `MIN-MEMORY-SIZE` of a VM may nevertheless be defined with a lower value because a corresponding amount of the resident JIT memory is released dynamically with `/REDUCE-VM-MEMORY` in order to comply with the 40% rule for JIT. The process is reversed with `/EXTEND-VM-MEMORY`.

On SQ servers the maximum main memory size (`MAX-MEMORY-SIZE`) determines the upper limit to which the VM's main memory can be extended (`/EXTEND-VM-MEMORY`). The default value is double the VM's `MEMORY-SIZE`. If the main memory of a VM is not to be extended, the value `MEMORY-SIZE` should be selected for the maximum main memory size.

The current utilization of resident memory can be determined by SM2 from the values of the MEMORY report group: `Resident Memory = TOTAL - Pageable Frames`.

6.3 Peripherals

Channels

On S servers the channels are available to all VMs. They are shared via the firmware. The utilization of the channels can be observed with SM2 over multiple VMs.

No channels in the original sense are available on SQ servers. X2000 emulates devices with a virtual I/O path which is available on all VMs.

Devices

The devices which exist are determined dynamically when the monitor system and guest systems are started up. VM2000, the monitor system and all guest systems therefore know and monitor the same devices.

Peripheral devices must be assigned to the guest systems individually. There are two options:

- **explicitly** by the VM2000 administrator with `/ADD-VM-DEVICES` or `/SWITCH-VM-DEVICES` or
- **implicitly** by a guest system with the privilege `ASSIGN-BY-GUEST=*YES(...)` with `/ATTACH-DEVICE`. For this purpose the VM2000 administrator must permit this type of assignment for the VMs and devices concerned.

  When disks are assigned implicitly, “Shared” is entered by default as the usage mode.

Peripheral devices can be assigned exclusively to a single guest system or be available to a number of guest systems as shared devices:

- **Exclusive assignment** = The device is then permanently assigned to one guest system

  I/O operations are then initiated directly as in native mode.

  For guest systems in the wait state (“idle”), the VM2000 hypervisor (S server) monitors the receipt of I/O termination interrupts and initializes the corresponding guest system in accordance with its scheduling priority. This hypervisor overhead is minimal.

- **Shared assignment** = The device is assigned to several different guest systems

  The type of I/O processing on S servers depends on the following:

  a) If the device is only assigned to one guest system, the I/O operations are executed directly (direct I/O) as in the case of exclusive assignment without the involvement of the VM2000 hypervisor.
  In the VM2000 information commands the device has the letters SH(D) against it.

  b) If the device is assigned to a number of guest systems, on S servers the VM2000 hypervisor is interrupts all I/O operations of the guest system and serializes them (indirect I/O).
  In the VM2000 information commands the device has the letters SH(I) against it.

  In the case of shared devices the CPU requirements of the VM2000 hypervisor increase with the I/O rate. The VM2000 overhead increases by roughly 0.4% for each 100 I/O operations per second.

  On SQ servers disk I/O operations are executed as logical requests via the RSC interface. There is no hypervisor overhead.

Service times for devices (monitoring program SERVICETIME of SM2) can be ascertained by only one VM at any given time.

Privilege IO-PRIORITY (S server)

Applications generally perform synchronous I/O operations. After it has started the task waits for the end of the I/O operation. If no further task is ready to execute, the wait state (“idle”) is switched to and the real CPU is redistributed by the hypervisor. After the I/O operation has been completed the guest system must wait for a real CPU to be assigned again before it can process the end of the input/output. The delay between the actual end of the I/O operation and it being processed in the guest system is referred to as the **increase in the hardware service time** or **IO increase**.

CPU-intensive guest systems can as a result disturb the operation of I/O-intensive guest systems. Indications of this are provided by `/SHOW-VM-STATUS INF=*SCHEDULE` in the `%RUNOUT` and `%WAIT` output columns.

The quicker the CPU is reassigned to the guest systems, the lower the IO increase is. On S servers, the `IO-PRIORITY=*YES` privilege can be assigned to a VM for this purpose. If a virtual CPU of such a VM is in the wait state, it is activated immediately on a real CPU when an I/O termination interrupt arrives from the hypervisor in order to process the result of the input/output. The runtimes of I/O-critical jobs are considerably improved.

The positive effect is that the hardware service times obtained are as good as in native operation, but on the other hand the scheduling rate is higher. As a result, the VM2000 overhead also increases.

It is therefore advisable to set up the guest systems in the first place without the privilege (i.e. with `IO-PRIORITY=*NO`) and only to assign the privilege `IO-PRIORITY=*YES` where necessary.

The `IO-PRIORITY=*YES` privilege provides no benefits in the case of dedicated CPUs.

Support of the IORM utility routine (I/O Resource Manager)

The IORM utility routine (see the “Utility Routines” manual [6]) is supported by VM2000. In VM2000 mode IORM should be running on the monitor system and on all guest systems.

The following IORM functions are implemented in conjunction with VM2000:

- Dynamic I/O load distribution for disks on the FC channel using DPAV in VM2000 mode (S servers only)

  Dynamic PAV (DPAV) autonomously assigns alias devices to the volumes which profit most from this. In VM2000 mode the actual switchover of alias devices is coordinated and executed by DPAV in the monitor system.

- Optimized device selection in ETERNUS CS HE operation under VM2000 with DDAL (Dynamic Device Allocation)

  As an alternative to using the devices in accordance with the way they are generated (as in the previous versions), the operating system can ensure even utilization of all the available ICPs (Integrated Channel Processors).

  Under VM2000 the IORM function DDAL extends optimized (local) device selection to cover all of a server’s BS2000/OSD guest systems as of V7.0.

- I/O limit for VM2000 guest systems with IOLVM (I/O Limit for Virtual Machines, S servers)

  In VM2000 mode, less important guest systems which are I/O-intensive can hinder other, more important guest systems in I/O operations.

  The IOLVM function of the IORM utility routine can detect such conflict situations and specifically slows down I/O operations of the user’s own guest system if one of the I/O resources that are used jointly (channel, port, path, disk) exceeds the specific I/O limit.

  The I/O limit for IOLVM is defined as the maximum I/O utilization of the VM in the `MAX-IO-UTILIZATION` operand in the VM2000 command `/CREATE-VM` or `/MODIFY-VM-ATTRIBUTES`.

6.4 Planning guest systems

6.4.1 Number and multiprocessor level of the VMs

The total number and multiprocessor level of the VMs affect the VM2000 overhead:

- Total number of VMs and total number of virtual CPUs
  - The more virtual CPUs are started on a real CPU, the less ideal the effect of the CPU caches is.

    By means of CPU affinity (see page 120) VM2000 (S servers) reduces the number of CPU switchovers in which a virtual CPU is executed on a different real CPU than before.
    CPU pools enable the number of active virtual CPUs per real CPU to be restricted.
  - The VM2000 hypervisor overhead for managing the CPU queues on S servers depends directly on the number of virtual CPUs in a CPU pool.

    The relation of the number of active virtual CPUs (of the guest systems with production load) to the number of real CPUs should be less than or equal to 3 in all CPU pools.
    The number of guest systems should be correspondingly low.
    When the performance requirements for the individual guest systems are lower, a larger number of VMs (up to 15) can also be configured. In all cases it must be checked whether the available main memory is sufficient for the guest systems required.
- Multiprocessor level of a VM (= maximum utilization of the VM)

  The load of the virtual CPUs of a VM has an effect on the utilization level of the time slice and consequently on the frequency of the scheduling operations. A high utilization level of the time slice and a low frequency of scheduling operations are favorable for performance.

  The multiprocessor level of a VM should be oriented according to the peak requirement of the guest system. As low a value as possible should be selected. If the load is low, virtual CPUs in the guest system should be detached.

6.4.2 Setting the CPU quotas

The CPU quotas for the VMs should be set in several stages:

1. Rough setting of the CPU quotas

In order to make a first approximation, it is recommended that you set the CPU quotas in the same proportions as the expected CPU utilization of the individual guest systems.

Example

Two guest systems (VM1 and VM2) are planned for a biprocessor system. These are estimated to require 50% and 20% of the computer capacity respectively.

This results in the following CPU quotas:

VM1: CPU-Q=50 (standardized EFF-Q=71,43)
VM2: CPU-Q=20 (standardized EFF-Q=28,57)

Since guest system VM2 will only place a load of around 20% on the server, it should be run as a monoprocessor guest system. Guest system VM2 will certainly require both CPUs.

2. Performance limit

The limit for the CPU performance for a VM (see page 120) must be selected so that the utilization is around 75% of the performance limit.

3. Measurement

The VM and the guest system are observed using `/SHOW-VM-STATUS` and SM2 (see page 139). The measurement values must be used to check whether the actual CPU time consumption and the performance behavior are in line with expectations.

If the guest systems consume approximately the expected capacity, the settings can be left as they are.

4. Checking the performance requirement

If the guest systems do **not** consume approximately the expected capacity, you must check the actual CPU requirements of the individual guest systems. To do this the guest systems should only be started individually, i.e. without competition from another VM. The guest systems should at least be prioritized one after the other with a very high effective CPU quota.

It can be assumed that the measured VM-ACTIVE percentage quotation corresponds to the actual CPU requirement of the guest system. The CPU quotas can now be recalculated on the basis of these values.

With regard to the workload on the server from all guest systems, it is recommended that you avoid CPU workloads > 75% on S servers in OLTP mode.

5. Fine-tuning the CPU quotas

Even after the approximate CPU quotas have been set correctly (i.e. the CPU quotas correspond closely to the capacity requirements of the guest systems), it is possible that the capacity requirements of individual guest systems will deviate sharply from the expected values. Guest systems with a high proportion of I/O operations generally fall below the expected values, whereas guest systems with high CPU requirements generally produce values above those expected.

In addition, the response times of important applications sometimes fail to meet requirements. In cases like this, the CPU quotas must be tuned, i.e. individual CPU quotas must be set higher (“overplanning”) or lower than the recommended values from the approximation.

It is not unusual for the CPU quotas to be corrected by a factor of 2. The performance requirements of the guest systems determine whether corrections are necessary and the extent of such corrections. This phase requires close cooperation with the administrators of the guest systems.

6. Forming VM groups (S server)

VMs can then be combined to form VM groups.

When VM groups are used, fine-tuning of the MEMBER-CPU-QUOTA is also recommended for the group members.

Overplanning of the CPU-QUOTA for the VM group is not recommended. If overplanning is necessary, individual VMs should be planned instead of VM groups.

6.4.3 Optimum configuration of VM2000

The constantly increasing performance capabilities of the BS2000/OSD servers mean that the timing for scheduling the VM2000 guest systems has also changed:

- In the case of applications which are not CPU-intensive (e.g. OLTP mode or batch mode with a large number of I/O operations of other interruptions), there is a need for a large number of generally only very brief intervals in which CPU performance is required (“CPU time slices”).
- When systems are utilized significantly less than 100%, normally only a few of these short CPU time slices are used immediately one after the other; the CPU then switches to the wait state (idle).
- In the case of TP loads severe fragmentation of the CPU requirement into small CPU time slices must therefore be reckoned with, as well as frequent, brief wait states of the CPU.
- The length of these CPU time slices decreases as the CPU performance increases.
- As a result of this, the number of scheduling operations of VM2000 rises as the CPU performance increases (or server performance increases) and as the multiprocessor level of the VMs increases.
  The maximum size of a CPU time slice is only utilized in exceptional cases. The higher the multiprocessor level of a VM, the less the maximum size of the CPU time slices is utilized.

One result of the increasing number of VM2000 scheduling operations is consequently also a slightly higher VM2000 overhead, but hand in hand with very good throughput values and good response times.

For OLTP applications it is therefore recommended that the priority controls between the DB server and DB client tasks be set in such a way that the job log of the DB server tasks is always full. This avoids overhead through additional communication.

In addition, a range of measures are available which enable you to limit the VM2000 overhead to the necessary minimum:

- Keep the number of the VMs' virtual CPUs (the multiprocessor level) as low as possible (see section “Number and multiprocessor level of the VMs” on page 129):
  - ► Specify the multiprocessor level of a VM. Do not simply use the multiprocessor level of the server.
  - ► When selecting the multiprocessor level of a VM, ensure that it is only large enough to permit the required (peak) load to be processed.
  - ► If possible, distribute the load over more than one VM with lower multiprocessor levels.

    In measurements on an S server with 4 CPUs a load was initially distributed over 2 guest systems each with 4 CPUs, and then over 4 guest systems each with 2 CPUs. It became apparent that the load distribution over 4 guest systems each with 2 CPUs caused around half the VM2000 overhead.
  - ► Do not start any guest systems “in reserve”. Such guest systems cause VM2000 overhead even when they bear no load.
  - ► You can configure guest systems with an appropriate multiprocessor level to handle load peaks. However, detach the CPUs in the guest system in times when there is no load peak (`/DETACH-DEVICE`).
- On S servers also make use of the option of dedicated CPUs (see section “VM2000 scheduling (S servers)” on page 118):

  In VM2000 operation without additional CPU pools, all available real CPUs are used for scheduling the (virtual) CPUs of all guest systems. For systems for which particularly high performance requirements apply (e.g. productive systems in contrast to test or development systems), it is consequently recommendable to use a mode which is very similar to native mode.
  - ► Make as many real CPUs available to such systems as there are (in total) attached virtual CPUs in the guest systems concerned.

  The result of this is that each virtual CPU has its own real CPU to use exclusively (“dedicated CPU”). This prevents unnecessary overhead resulting from avoidable cache losses and wait times for a free real CPU or multiprocessor locks.

  The following configuration, for example, is suitable for this purpose:
  - ► In addition to the standard CPU pool, configure another CPU pool for critical productive systems with dedicated real CPUs.
    All other guest systems use the standard CPU pool with the remaining real CPUs.

► On S servers, also use the VM attribute `VM-ACTIVE-IDLE` for VMs with dedicated CPUs.
This function prevents virtual CPUs in the wait state (idle) from leaving the real CPU. There is hardly any hypervisor overhead.

A measurement on an S server with 4 CPUs and only one VM showed that when dedicated CPUs and the VM attribute `VM-ACTIVE-IDLE` are used, the VM2000 overhead is reduced by 75% compared to when dedicated CPUs are used **without** the VM attribute `VM-ACTIVE-IDLE`.

No evaluation of the VM-ACTIVE times (`/SHOW-VM-STATUS` or VM report of openSM2) is possible for VMs with `VM-ACTIVE-IDLE`.

- As far as possible avoid using disks which are assigned as “shared”.

  Refer to the information on exclusive and “shared” assignment in section “Peripherals” on page 125.

  Measurements have shown that with more than one guest system the VM2000 overhead increases by 0.4% for every 100 I/O operations per second on disks which are assigned as “shared”.

  ► Limit the use of disks assigned as “shared” under VM2000 to the necessary degree.

- In OLTP mode ensure that the server's total CPU utilization does **not** exceed the value of 75% which we recommend.
- Use KVP consoles. These cause less VM2000 overhead than virtual consoles.

6.5 Systems support in the guest system

VMs and guest systems with batch-oriented, CPU-intensive load are preferred by VM2000 scheduling. Dialog-oriented guest systems with frequent idle times are hampered in their response times by the IO increase. If necessary fine tuning of the CPU quotas must be performed, see page 131.

The task priorities in a dialog-oriented guest system have less effect. If required, the priorities must be set anew, see the following section.

A guest system that only makes little use of its time slice is hampered in comparison with native mode by the I/O increase on the one hand and the worse cache hit rates on the other. In particular on S servers, execution in a CPU pool with dedicated CPUs and the `VM-ACTIVE-IDLE` VM attribute can help, see page 120.

When PCS is used the increase in a category can be influenced in both directions, see section “PCS under VM2000” on page 137.

6.5.1 Setting priorities

If you distribute an IT workload across several different guest systems, the degree of concurrency of the individual components of the load changes. This means that priorities sometimes need to be changed. The following example illustrates this.

The following load components are running on a server in native operation:

- one UDS/UTM application
- several batch database applications
- a number of dialog tasks involving program development with no access to the databases

Up to now priorities and CPU utilization have been as follows:

	UDS tasks	UTM tasks	Batch	Dialog1	Dialog2
Number of tasks	1	8	2 - 6	3	10
Priority	150	150	255	180	210
Workload (%)	20	30	10	5	10

The IT load is now to be distributed over two guest systems as follows: The developer activities are to be outsourced to a separate guest system. All other components should run on another guest system on account of their shared databases

VM	Name	CPU quota	Load
VM1	TP system	80	UDS/UTM + batch
VM2	Dialog system	20	Dialog1 + Dialog2

The fact that the dialog load has been relocated means that the situation for batch tasks improves, since these can now always run when the TP load is waiting for inputs/outputs. The throughput of CPU-intensive batch tasks improves, since they tie the real CPU to the guest system.

Systems used purely for dialog tasks generally have a high proportion of idle time. The low CPU quota means that the dialog system will have to wait for a relatively long time after each idle state. This means that the response times for the dialog tasks may increase.

The preference given to dialogs with a high priority compared with those with a low priority also decreases. The following measures are necessary if the same results are to be maintained after the load has been split:

- If the dialog response times increase dramatically, the CPU quota of the dialog system must be raised. The TP response times and the batch throughput must be constantly monitored.
- The difference in priorities between Dialog1 and Dialog2 must be increased.
- The priorities of the UDS and UTM tasks must be increased in order to restore the original difference to the batch tasks, which now have a greater priority.

6.5.2 PCS under VM2000

If PCS (Performance Control Subsystem) is running under VM2000, the CPU quota of the corresponding guest system is taken into account internally. In other words, within a guest system the sum of the S-Q MAX value for response time-critical categories should be 100 (≙ 100% of the guest system capacity).
Compared to native operation, a correction generally needs to be made to the parameters of the SERVICE ranges (range between SERVICE-QUOTA MIN and MAX, controlled by the dilation parameters REQUEST-DELAY MIN and MAX).

The dilation (REQUEST-DELAY, see also section “Introduction to PCS concept” on page 222) is defined as follows:

```
Dilation = Runtime in multiprogramming mode /  Single runtime
```

Under VM2000, it is possible that both runtimes will increase. The single runtime can increase, for instance, as a result of increased I/O times caused by other guest systems or as a result of a reduction in hardware performance because of a lower cache hit rate.

Depending on the load profile of the individual categories in a guest system and the load profiles of the other guest systems, it is possible that the PCS dilation for individual categories may increase in comparison with native mode (which is to be expected) or decrease.

Checking or correction of the SERVICE-QUOTA and REQUEST-DELAY parameters should be done in the following sequence:

- SERVICE-QUOTA MIN
  This sets the service quota a category requires in normal operation, i.e. without taking workload peaks into account.
- SERVICE-QUOTA MAX
  This sets the maximum percentage of the capacity of the guest system which can be assigned to a category during peaks in the workload.
- REQUEST-DELAY MIN
  The system administrator should take a series of measurements to identify the value for REQUEST-DELAY ACTUAL as of which a bottleneck or unsatisfactory behavior in a category arises, i.e. the value at which the service rate should be increased. REQUEST-DELAY MIN is then set to this value.
- REQUEST-DELAY MAX
  This defines the degree of response to a workload peak. The nearer this value is set to that of REQUEST-DELAY MIN, i.e. the smaller the value, the stronger the reaction to peaks in the workload.

6.5.3 Operating the guest system

For performance reasons it is recommended that the guest systems should be operated via KVP consoles.

Guest systems can also be operated using virtual consoles (via $VMCONS). In this case only basic operation (startup) should be handled using a virtual console. Actual operation should be performed via logical consoles using OMNIS.

OMNIS connects directly to $CONSOLE on the corresponding guest system and thus bypasses $VMCONS. Requests to $VMCONS are synchronized via a bourse. This bourse can become a bottleneck if there is a large number of guest systems or console I/Os.

6.6 Measurement tools for VM2000

Two measurement tools are available for VM2000 operation:

- In the monitor system `/SHOW-VM-STATUS` performs measurements at VM2000 level for VMs, VM groups, CPU pools and real CPUs. Here some internal VM2000 performance values are also output. The hypervisor activities, planned and actual loads for CPU utilization and VM2000 scheduling data can be monitored.
  This command can be used to output both periodic measured values and also measured values from the immediate past.
- The openSM2 software monitor with online reports (INSPECTOR) and long-term analyses (ANALYZER).

The following ANALYZER report groups are recommended for observing the utilization of the guest systems and of the hypervisor using openSM2.

- In the guest systems

  CPU: CPU utilization
  CPU utilization in the processor states TU, TPR, SIH. These values are the utilization values determined by the guest system concerned (in relation to the number of virtual CPUs available on the guest system!).

- In the monitor system

  VM2000: VM2000 utilization/VM2000 hypervisor
  CPU utilization, current CPU quota and current maximum performance utilization per VM. CPU consumption of the hypervisor and hypervisor idle

  VM2000: CPU pool
  CPU utilization and dimensioning of the CPU pools.

  VM2000:VM groups
  CPU utilization, current CPU quota and current maximum performance utilization per VM group.

Some SM2 values must be reinterpreted for VM2000 mode. Refer to the section entitled “SM2 together with VM2000” in the “openSM2” manual [18].

7 Performance behavior of the software

This chapter deals with the performance criteria when setting up the system environment. The term “system environment” refers to the complete environment of the application programs. It comprises the system kernel and the subsystems.

System kernel

The system kernel of BS2000/OSD consists of basic control mechanisms and supporting system tasks which allow optimum use of the operating resources and communication between user and machine, as well as system routines which make programming easier. The tasks of the system kernel, which is given the highest priority, are as follows:

- Job management

  Provides the external connection:
  – Logon and Logoff handling for the terminal users
  – communication with the operator
  – processing the job input stream

- Task management

  Controls and monitors the execution of the user and system tasks with the help of the basic control mechanisms:
  – interrupt analysis
  – Creating and terminating tasks
  – managing the resources main memory and CPU
  – controlling the tasks via queues

- Data management

  Performs the measures necessary for handling the data:
  – handling the input/output between main memory and I/O units
  – volume management
  – logical access methods (SAM, ISAM, PAM, FASTPAM, DIV, BTAM)
  – supporting the HIPERFILEs together with DAB (see section “Working with HIPERFILEs” on page 176)

Subsystems are arranged around the system kernel, depending on the tasks to be performed (e.g. editing tools, compilers, DC components, databases etc.).

Subsystems

The subsystems fall into two groups:

- Privileged subsystems

  These run in the processor state TPR (e.g. SPOOL, NDM, ARCHIVE) and are so closely linked to the system kernel that they cannot be loaded with the loader programs intended for the user.

- Nonprivileged subsystems

  Vis-a-vis BS2000/OSD, these have the status of normal application programs and therefore only use the interfaces that are open for the user (e.g. compilers, editors, databases, utility routines).

Setting up the system environment

Starting from the system kernel, setting up the system environment involves all activities for creating a task-based operating system. System setup includes:

- Setting the size of the user address space on S servers
  (see section “Size of the user address space” on page 143)
- Creating the files
  (see section “Performance criteria when creating files” on page 145)
- Defining the HIPERFILEs
  (see section “Working with HIPERFILEs” on page 176)

The following notes should help you perform these tasks in such a way as to fully utilize the performance capability of BS2000/OSD.

7.1 Size of the user address space

SQ servers

The user address space for SQ servers is generated with 2 GB and cannot be modified.

S server

The total address space consists of the user address space (including the space requirement for shared products; see the SHRSIZE parameter) and the system address space. For the size of the total address space on S servers there is a choice of the values 1 or 2 GB.

On S servers the size of the user address space automatically determines the size of the system address space as the difference to the next higher value of the total address space. It is no longer necessary to specify the SYSSIZE parameter in the parameter service during system startup.

By default, the user address space for S servers is defined with a size of 1,808 MB. This size can be changed using the `SYSPRC.BS2000-EXEC.<version>` procedure (see the “System Installation” manual [31]).

With configurations which require a large system address space, the user address space must be reduced to 1,792 MB (SYSSIZE=256 MB), 1,664 MB (SYSSIZE=384 MB) or 1,536 MB (SYSSIZE=512 MB).

One of the reasons the subdivision of the total address space was introduced was to reduce the main memory requirement for the address space management (segment table). Depending on the size of the total address space (1 or 2 GB), the following main memory per task is occupied for the segment tables:

- 4 KB for an address space of 1,024 MB
- 8 KB for an address space of 2,048 MB

With a small number of tasks, selection of the largest possible address space (2 GB) is not critical. With a large number of tasks, you must check whether the extra main memory required by some large applications is justified.

Page-table management requires 3 KB main memory per megabyte of occupied address space and task. If, for example, 10 tasks completely occupy the maximum available user address space of 1,808 MB, the following main memory (resident class-3 system address space) is required for managing the address space alone (segment table and page table):

10 *	8 KB	=	80 KB	for segment table
10 *	1,808 MB * 3 KB/MB	=	54,240 KB	for page table
			54,320 KB	

With large virtual address spaces, you must make sure that the paging areas are large enough. The user-specific size (/ADD-USER, /MODIFY-USER-ATTRIBUTES) must be defined carefully. If many large address spaces are to be used, a correspondingly large-sized main memory is required.

7.2 Performance criteria when creating files

BS2000 users and system administrators can generally only influence the placement of the files with respect to logical volumes. At physical level a number of logical volumes can be located on the same physical disk. The resultant effects on performance and the performance measures are described in section “Configuration in BS2000/OSD” on page 52.

Information on the performance of Net-Storage is provided in section “Net-Storage” on page 86.

In the following sections in this chapter the (previous) term “disk” is still used synonymously for the (current) term “logical volume”.

7.2.1 Logical operating modes of disks

In BS2000/OSD, disks are addressed logically via the volume serial number (VSN). A volume can be operated as a public volume or as a private volume,
Public volumes are operated in SF pubsets (Single-Feature pubsets), SM pubsets (System-Managed pubsets) or shared pubsets.

Irrespective of which logical operating mode is in use, the time taken to perform a single data access operation is approximately the same (if the disks are of the same type).

There are, however, serious differences regarding the number of I/O operations to be performed when calling functions which require catalog access. This is due to the differing degrees of complexity with regard to memory management or the co-ordination of multiprocessor operation.

To increase availability, the following operating modes are also provided:

- Dual recording by volume (DRV)
  Data mirroring controlled by BS2000/OSD
  (software product DRV, for details see the “DRV” manual [7])
- Replication (disk mirroring)
  Disk mirroring, controlled by the ETERNUS DX and Symmetrix disk storage systems

  Data mirroring is a component of the RAID concept.
  RAID (Redundant Array of Independent Disks) is an architecture which defines how to map a number of physical disk drives to a single logical volume in order to achieve data redundancy, see page 48.

Public volumes

Memory management is performed by the system and offers optimal support for calls using catalog access:

- a high degree of user convenience by means of considerable simplification of the definitions for creating, extending and deleting files
- a lower number of management I/Os compared with private volumes (management of F5 labels in main memory, support from the software product SCA, see page 150).

The location of a file can be specified explicitly for all pubsets by systems administration. and for selected pubsets by the user as well, where the authorization for physical allocation is entered in the pubset-specific user catalog (`/MODIFY-USER-ATTRIBUTES`, `PHYSICAL-ALLOCATION` operand). This is important when creating special system files (see section “Creating system files” on page 156).

Pubsets are the strategic platform for further developments in the disk area. Private volumes do not offer full functional support of new functions, e.g. in the HIPERFILE concept.
Migration from private volumes to pubsets is thus recommended.

SF pubsets

As far as memory management is concerned, the same applies as for public volumes. In addition to an increase in availability, the following advantages can be seen as far as performance is concerned (particularly important for interactive mode):

- Splitting the file catalog TSOSCAT over separate pubsets improves catalog performance if the catalog functions are used intensively.
- It is possible for systems support to control input/output distribution by means of the user catalog entries. Users with files subject to particularly frequent access can be transferred to separate pubsets.

SM pubsets

SM pubsets consist of a number of volume sets that may have different formatting, availability and performance attributes specified by systems support.

In comparison to SF pubsets, which consist of a single volume set, SM pubsets offer better control of input/output distribution to systems support and greater convenience to users:

- By specifying direct attributes or a storage class when creating a file, a user can specify where the file is to be stored on an SM pubset.
- Systems administration assigns a volume set list to each storage class and defines the desired performance requirements for each volume set.

You will find detailed information on SMS (System Managed Storage) and SM pubsets in the “SMS” manual [32].

Shared pubsets

Two or more write-authorized servers (or guest systems under VM2000) can access shareable pubsets simultaneously (except in the case of the home and paging pubsets). The servers are interconnected via a BCAM link and work according to the master/slave principle. All catalog operations (RDCAT, WRCAT, ...) are always performed by the master server. It is thus only practical to use SCA in the master.

Taking into account the following points, the performance behavior is virtually the same as for “normal” public volumes:

- the server from which most accesses are performed is set as the master (to minimize cross-server actions)
- to keep the overhead for cross-server actions to an acceptable level, no more than two OPEN/CLOSE operations per second should be carried out on the slave server for each RPF (Relative Performance Factor).

Files on shared pubsets can be supported from any server with DAB and local caching.

Private volumes

Memory management is the responsibility of the user. The user can avoid access conflicts in the case of files which are used regularly by specifying the exact location of the files in the relevant volumes.

The number of management I/Os in the case of file processing calls with catalog access is considerably higher than with public volumes owing to additional accesses to the VTOC (Volume Table Of Contents) area. There are, for instance, 6 logical accesses to the F1 labels per OPEN/CLOSE pair, each of which requires one or more physical I/Os.

Thus efficient performance behavior can only be achieved if the number of management I/Os is small compared with the number of “productive” I/Os.

DRV: Dual Recording by Volume

DRV increases the availability of information stored on disk by recording the data on 2 physical disk drives. The duplication takes place in the IO sytem of BS2000/OSD.

For a synchronous write operation, first of all an asynchronous output to the first disk is initiated, then a synchronous output to the second disk drive.

The disk drive with the shorter queue is selected when a read operation is to be performed, thus resulting in an unsymmetrical load on the original disk and mirror disk. Selecting the least heavily used disk drive reduces the I/O time (software duration) for read accesses in comparison to SRV (Single Recording by Volume) mode.

Approximately the same performance behavior can be expected as for SRV mode.

Copy speed

The copy speed can be set dynamically in three steps for each pubset using the operand `COPY-SPEED=*LOW/*MEDIUM/*HIGH` of `/SET-DRV-PARAMETER` at both restoration and equalization. With `COPY-SPEED=*LOW`, the DRV copy operations are slowed down considerably so that the pubset for the I/O operations of applications is more accessible. The default is `*MEDIUM`.

Extent of copying

Not all the disk's blocks are copied at restoration; only the blocks identified as occupied in the F5 label are copied.
After a system failure or an abnormally terminated export of a pubset, DRV carries out equalization in order to match the disk pairs again.
In addition to the existing variants of copying either all the blocks or only the system data, DRV also allows only the system data and the open files to be equalized. This option is only possible for SF pubsets or volume sets of an SM pubset. It can only be set by `/SET-DRV-PARAMETER UNIT=*PUBSET(...)/*VOLUME-SET(...), EQUALIZE-DATA= *OPEN-FILES` before the pubset or volume set is imported.

Rapid restoration after separate processing

In the case of separate processing (e.g. backup of the original data, batch processing of the mirror data), DRV remembers the changed areas. This allows the restoration time to be dramatically reduced when only the changes since the beginning of separate processing have to be copied (since this can be measured in terms of Mbytes) rather than all the files (Gbytes of data).

Replication (disk mirroring)

In the ETERNUS DX and Symmetrix disk storage systems, the RAID1 level is available on the hardware level. Dual data recording takes place without the need for any further software support. See also page 48.

7.2.2 Accelerating catalog accesses with SCA

SCA (Speed Catalog Access) is a recommended software product for accelerating catalog services.In searches for file entries and job variable entries, SCA replaces sequential searches for catalog blocks with direct accessing.
This is accomplished using two tables. One table provides information about free space in the catalog blocks, and a second (reference table) assigns the logical block number of the catalog block to the file names or job variable names.
Both tables are created by the SCA task.

SCA can only be called for SF pubsets. In the case of SM pubsets, accelerated access to catalog entries is already implemented internally.

Two versions of SCA are available: with and without task switching. The use of SCA without task switching is recommended.

- SCA without task switching

  The reference table is created in class 4 memory and is thus available to all tasks. All SCA routines (except for table creation) run under the accessing user task. There is no overhead for task switching, and the overhead for serialization measures is minimal.

- SCA with task switching

  The reference table is managed by the SCA task in its class 5 memory.
  Each catalog service request from a task results in a task switch.

You can store the variants that are to be used for a pubset or specify whether SCA is to be started automatically (standard for BS2000/OSD V8.0 and higher) when the pubset is imported in the MASTER catalog MRSCAT (`START-SPEEDCAT` operand in `/ADD-` and `/MASTER-CATALOG-ENTRY`).

The “SCA without task switching” version may be expected to require approx. 1 - 2 % less CPU time, but somewhat more class 4 memory than does the “SCA with task switching” version.

Automatic startup of SCA

Since BS2000/OSD V8.0, if SCA is installed SCA has by default been started in `*SPEEDCAT-TASK` mode when a pubset is imported in order to guarantee high-performance catalog access with SF pubsets.

Since BS2000/OSD V8.0 the new default value `START-SPEEDCAT=*AUTOMATIC` has therefore been used in the `ADD-MASTER-CATALOG-ENTRY`command.

Advantages of using SCA

1. Increased total throughput rate for the system when the workload involves an intensive use of the catalog.

   The reduction in the number of physical accesses of TSOSCAT per logical catalog call (OPEN, CLOSE, ...) results in a lowering of the system workload and a marked reduction in the catalog access time.

   In addition, the burden on the disk drives containing the catalog files is reduced. If SF pubsets are used, these are generally the xxxx.0 volumes. As far as the whole system is concerned, this means that the throughput of the accessing user tasks can be increased.
   Runtime for programs with a high frequency of catalog access (e.g. ARCHIVE) as well as response times for commands which need to access the catalog regularly (e.g. `/CREATE-FILE`, `/MODIFY-FILE-ATTRIBUTES`, `/IMPORT-FILE`) can be improved considerably.

   SCA is indispensable for fast systems operating in interactive mode. Its use leads to a marked increase in throughput.

2. Greater flexibility in computer center organization

   The use of SCA makes one of the previous organizational methods of improving catalog performance redundant. It is therefore no longer necessary to place frequently used catalog entries at the beginning of a catalog block chain.

3. Full compatibility

   SCA does not change the structure of the catalog. As a result, both the user interfaces and the internal system interfaces to the catalog management system are retained.

Significant improvements in performance can only be expected when the user has a large number of catalog entries (more than 60-100 file entries per user ID).

SCA can be started and/or terminated for different catalogs simultaneously, either automatically, when the pubset is imported (determined by the MRSCAT entry), or by procedure call (`/ENTER-JOB FROM-FILE=SPEEDCAT.ENTER.START`) after the pubset has been imported.

7.2.3 Use of SF pubsets

When running applications with certain input/output requirements, it makes sense to group similar public volumes together in pubsets.

One pubset should comprise 4 or less public volumes. The advantage of this is that, in the event of a failure, only the pubset concerned is blocked, and operation can continue to a limited degree with the remaining pubsets. Creation of a standby pubset (a copy of the home pubset) for the purpose of a quick restart is recommended (for a description of handling see the “Introductory Guide to Systems Support” [10]).

Using a number of SF pubsets improves performance by increasing the range of distribution options for files.

Notes on the distribution of the TSOSCAT system files, paging areas and SYSEAM are given in section “Creating system files” on page 156.

In order to distribute the input/output requirements of users who work under different user IDs by means of system specifications, the following parameters must be taken into account when /ADD-USER or /MODIFY-USER-ATTRIBUTES is used:

```
/ADD-USER USER-IDENTIFICATION=xx, DEFAULT-PUBSET = *HOME / <catid>      (1)
                                  PUBSET = *HOME / <catid>              (2)
                                  PUBLIC-SPACE-LIMIT = <max>/<integer>  (3)
```

(1) <catid> identifies the pubset in which the user can access files without the specification of a catalog ID, e.g. in /CREATE-FILE, /ADD-FILE-LINK, /MODIFY-FILE-ATTRIBUTES.

(2) <catid> defines the pubset which can be accessed by the user. It is only possible for a user to access a particular pubset if an entry exists for him/her in the user catalog of the relevant pubset.

(3) <max> specifies the maximum public space available to this user ID for the pubset specified in (2).

One exception is when the pubspace value for a pubset is specified as zero for a given user ID. This user then has access to all the files whose access rights are set appropriately. He/she cannot, however, create any files in this pubset.

7.2.4 Use of SM pubsets

Whereas the formatting attributes of a volume set, which are defined at initialization, remain unchanged for as long as it exists, the availability and performance attributes can be changed in the ongoing operation. Only the performance attributes are dealt with below.

SM pubsets from the point of view of the user

So called storage services can be requested for a file using /CREATE-FILE or /MODIFY-FILE-ATTRIBUTES (STORAGE-CLASS operand) or the FILE macro (STOCLAS operand):

Operand	Meaning
STORAGE-CLASS	File attributes which are relevant for the storage location.
= *STD	Assigns the default storage class from the user catalog of the SM pubset.
= <name>	Assigns a specific storage class.
= *NONE	There is no explicit assignment of a storage class. The user defines the performance attributes (*STD / *HIGH / *VERY-HIGH / *USER-MAX).

The user can obtain information on the names and attributes of the available storage classes (i.e. access must be permitted when GUARDS protection is set) with /SHOW-STORAGE-CLASS.

The user can get information on the storage services of the existing volume sets with /SHOW-PUBSET-FILE-SERVICES and thus check whether there is a volume set with the desired attributes for his/her file.

The storage services can also be requested by means of direct attributing.

SM pubsets from the point of view of systems support

In order for the user's performance requirements for the volume sets of an SM pubset to become effective, corresponding authorizations and storage space quotas must be entered in the user catalog (/ADD-USER, /MODIFY-USER-ATTRIBUTES or /MODIFY-USER-PUBSET-ATTRIBUTES):

Operand	Meaning
DEF-STORAGE-CLASS = <name>	Default storage class when the user specifies STORAGE-CLASS=*STD

Operand	Meaning
DMS-TUNING-RESOURCES	For an explicit request via direct operand or implicitly via a storage class:
= *NONE	caching not permitted
= *CONCURRENT-USE	for PERFORMANCE=*HIGH
= *EXCLUSIVE-USE	for PERFORMANCE=*VERY-HIGH
PERM-/TEMP-/WORK-SPACE-LIMIT = *PARAMETERS(...)	Specifies storage space quotas of the users (for permanent, temporary and work files, respectively):
HIGH-PERF-SPACE= ...	for high-performance storage space
VERY-HIGH-PERF-SPACE= ...	for very high-performance storage space

Systems support uses /CREATE- or /MODIFY-STORAGE-CLASS to create or modify storage classes.

When the user assigns a storage class to a file, this file implicitly receives all the performance attributes set in the storage class.

Operand	Meaning
PERFORMANCE	Defines the caching behavior (main memory, GS):
= *STD	No caching
= *HIGH	Caching with preemption on the basis of LRU
= *VERY-HIGH	caching without preemption
USAGE	Increased performance requirements apply:
= *READ-WRITE	to read and write operations
= *READ	to read operations only
= *WRITE	to write operations only
DISK-WRITE	Specifies when data consistency is restored:
= *IMMEDIATE	Data consistency after every write operation (prerequisite: non-volatile cache medium)
= *BY-CLOSE	Data consistency after CLOSE processing
= *STD	*IMMEDIATE applies for permanent files, *BY-CLOSE applies for temporary files
VOLUME-SET-LIST	Assigns a volume set list
= <name>	Assigns a specific volume set list
= *NONE	No volume set list is assigned

To meet higher performance requirements (*HIGH / *VERY-HIGH), systems support must assign main memory or global storage as the cache medium (/MODIFY-PUBSET-CACHE-ATTRIBUTES ; see section “User PFA Caching” on page 184).

When a volume set list is specified for each storage class, computer center-specific strategies for controlling input/output distribution can be implemented. If a user assigns a storage class with an associated volume set list to a file, the file is created, if possible, on a volume set contained in the list. A volume set list is created using /CREATE-VOLUME-SET-LIST.

The performance attributes of a volume set are defined by means of /MODIFY-PUBSET-DEFINITION-FILE:

Operand	Meaning
PERFORMANCE	Defines the caching behavior (main memory, GS):
= *STD	No caching
= *HIGH	Caching with preemption on the basis of LRU
= *VERY-HIGH	caching without preemption
= list-poss(3) ...	List of values permits a performance range
WRITE-CONSISTENCY	Specifies when data consistency is restored:
= *IMMEDIATE	Data consistency after every write operation (prerequisite: non-volatile cache medium)
= *BY-CLOSE	Data consistency after CLOSE processing

Volume set selection will not function properly unless the performance profile is described adequately. The performance profile is not determined automatically on the basis of the physical configuration. It is the job of systems support to describe the real conditions accurately (taking into account a volatile or non-volatile cache medium when assigning WRITE-CONSISTENCY).

7.2.5 Creating system files

System files are required by the operating system to complete its tasks. Because of their crucial importance, these files are amongst those that have the greatest effect on the performance of an IT system. Of the many system files, the following are the ones accessed most regularly:

Function	DMS name
File catalog (SF pubset) File catalog (SM pubset)	$TSOS.TSOSCAT $TSOS.TSOSCAT.<volset-id>
Paging areas	$TSOS.SYS.PAGING.<vsn>
Background storage for access method EAM	$TSOS.SYSEAM

Priority must be given to **avoiding access conflicts** when creating these files.

The usual way of reducing the probability of access conflicts is to distribute the files over several volumes. If, for reasons of economy (e.g. utilization of disk capacity), user files and system files are placed on the same volume, then it is desirable to select user files which are not frequently used.

The characteristic features of system files are the **file size** and the **access frequency**.

If a system has already been set up, the location of the paging areas can only be changed with great difficulty. Therefore particular attention must be paid to the location and size of this system file at configuration.

If the system is being used for program development, then cataloged module libraries (TASKLIBs) and macro libraries are often accessed so regularly that their location must be taken into consideration at configuration.

7.2.5.1 File catalog (SF pubset)

The file catagog (this is the file TSOSCAT) is only accessed from CMS (Catalog Management System) routines. The PAM blocks in this file contain:

- file entries or
- job variable entries or
- global management data
  (MRS catalog and entries describing the current state of TSOSCAT)

The file is accessed in blocks. The system parameter BMTNUM (see page 337) or `/IMPORT-PUBSET` or `/ADD-` and `/MODIFY-MASTER-CATALOG-ENTRY` are used to specify how much I/O buffer is available to the CMS. Simultaneous user requests to the same catalog block are synchronized with the aid of a Buffer Management Table (BMT) and a bourse (one BMT and one bourse per catalog I/O buffer).

The catalog file can be divided into several extents. For technical reasons, the first extent of the file must be on the volume xxxx.0. If there are several pubsets, there is a catalog on each pubset. Placing parts of TSOSCAT on heavily used disk drives must be avoided. All volumes with paging areas and SYSEAM area fall into this category.

Catalog formats

There are three catalog formats:
NORMAL and LARGE (implicitly for pubsets with objects > 32 GB) and EXTRA LARGE.

TSOSCAT type	Max. size (catalog blocks)	Max. size (PAM pages)	Compatible with
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

The catalog formats LARGE and EXTRA LARGE are incompatible:
i.e. if they are used for a pubset, a downgrade to an older BS2000 version is no longer possible for this pubset.

Up to 100 subcatalogs each with 64,016 PAM pages can be created for SM pubsets with catalogs in EXTRA LARGE type for each of the special catalogs #MIN, #JVC and #PTV (`/ADD-CATALOG-FILE`).

CMS recognizes when a catalog is 90% full and automatically extends it provided this is possible without changing the catalog format.

For special catalogs in EXTRA LARGE type a new subcatalog is then created if none of the existing subcatalogs can be extended.

The file can also be expanded manually using the /MODIFY-FILE-ATTRIBUTES command (SPACE operand). The change becomes effective immediately and not only after the next time the pubset has been imported.

Creating TSOSCAT

The size of TSOSCAT is determined by

- the number of users permitted to use it
  (represented by the user IDs entered by means of /ADD-USER)
- the number of files and job variables in the system
- the size of the catalog entries; with files this is dependent on the number of extents they have, and with job variables on the size of the JV value.
- the splintering of catalog entries, which is caused by selective deletion
  The space which becomes free by deleting “individual” entries is available for creating new entries, but is not used for anything else. Only after all entries have been deleted is a catalog block released.

The entries in TSOSCAT are stored in catalog blocks (each 4 KB in size):

- A catalog block for files, the so-called “primary block”, is reserved for each user even if a user creates no files.
- A catalog block contains either only file entries of one user or only JV entries of one user.
- 1 to 13 file entries fit into a catalog block.
  An average of 10 file entries per catalog block can be expected on a pubset which is sufficiently large and maintained using SPACEOPT.
- 8 to 18 JV entries fit into a catalog block.
  It is difficult to specify an average because the size of the JV values can be freely selected by the user. For example, in the case of MONJVs with a standard size of 128 bytes, 11 JV entries fit into a catalog block.

The average number of the catalog blocks which are occupied by “primary blocks” and file entries can be estimated using the following formula:

```
Number of catalog blocks + Number of user IDs + (Total number of files / 10)
```

The formula takes into account the splitting up of catalog entries and the decreasing capacity as the size of the entries increases. Additional catalog blocks are required for job variables.

Example

Calculation of the catalog size for 150 user IDs with 65 files per user ID:

```
TSOSCAT = 150 + (150 * 65 / 10) = 1125 catalog blocks
```

If the software product SCA (Speed Catalog Access) is not used, no more than 65 files or job variables should be cataloged per user ID . If SCA is used, then a greater number of files or job variables is possible.

Location of TSOSCAT

A TSOSCAT file of 2000 PAM blocks is automatically created on volume xxxx.0 at system generation.

Assuming that the paging area created on the system residence (xxxx.0) is **not used during subsequent operation**, TSOSCAT should be stored on volume xxxx.0 in its required size (in the case of installation using SIR).

If the load on volume xxxx.0 essentially consists only of user activities (max. 15% capacity utilization) and catalog accesses on the part of the system (also max. 15% capacity utilization), it will manage 50 catalog I/O operations per second at a 70% read hit rate. It would be advantageous to reduce this by increasing the number of CMS input/output buffers and using SCA.

If several pubsets are used, a copy of TSOSCAT is automatically placed on each pubset. The recommendations are then true for volume xxxx.0 for each pubset.

Once normal operation has been started, it is advisable to monitor the number of catalog I/O operations per second using the software monitor SM2 (reports “CMS Queues”, “CMS IO'S” in the report group CATALOG-MANAGEMENT).

As well as the frequency of catalog I/O operations, it is also necessary to consider the address space requirements for tables and buffers and delays resulting from synchronization when calculating CMS performance for productive operation. If multiple pubsets are used, catalog I/O buffers and their management tables (BMT) are created for each pubset. If there is a large number of pubsets, then the address space required by CMS can increase considerably.

It is possible to define the number of CMS buffers for each pubset with `/ADD-`, `/MODIFY-MASTER-CATALOG-ENTRY` or `/IMPORT-PUBSET`.

Example

The system parameters BMTNUM=32 and CATBUFR=N were specified in the startup parameter service. For 2 pubsets CMS needs the following amount of work area:

(33 * (4096 + 178) + 388) * 2 = 282,860 bytes, i.e. 277 KB of class 4 memory

- Number of pubsets
- Central CMS work table
- BMT + DMS access tables
- Catalog block I/O buffer
- Number of CMS I/O buffers + 1

An additional I/O buffer for the static MRS catalog (approx. 4 KB) is required for pubsets with EXTRA LARGE catalogs.

To protect simultaneous requests, one bourse is created per catalog buffer. In addition, further bourses are used as protection against competing calls from CMS routines.

From time to time, it is necessary to revise the organization of files on disks and of file entries and job variable entries in catalog blocks. This is done in order to keep path lengths as short as possible when searching for file entries and job variable entries, and to ensure that as few catalog buffers as possible are locked for simultaneous requests from other users during this search. During this process, all files and job variables must be saved, deleted and recreated. When recreating, two requirements - which may possibly be contradictory - must be fulfilled as far as possible.

1. On the one hand, large files should be transferred into the system first, as this allows them to be distributed according to considerations of practicability. Small files can be placed in gaps.
2. On the other hand, CMS capacity is used efficiently if file entries for heavily used files are as near as possible to the beginning of the series of catalog blocks for each user ID. CMS starts searching for file entries or job variable entries in the first block of the series of user catalog blocks in each case.

The second requirement should be given priority.
If there are too many user IDs and file and job variable entries per user ID, SCA should be used (for details, see section “Accelerating catalog accesses with SCA” on page 150).

7.2.5.2 File catalog (SM pubset)

An SM pubset consists of a control volume set and a number of “normal” volume sets.

The file catalog of an SM pubset is distributed to the individual volume sets in such a way that every subcatalog contains the information for the files on the corresponding volume set.

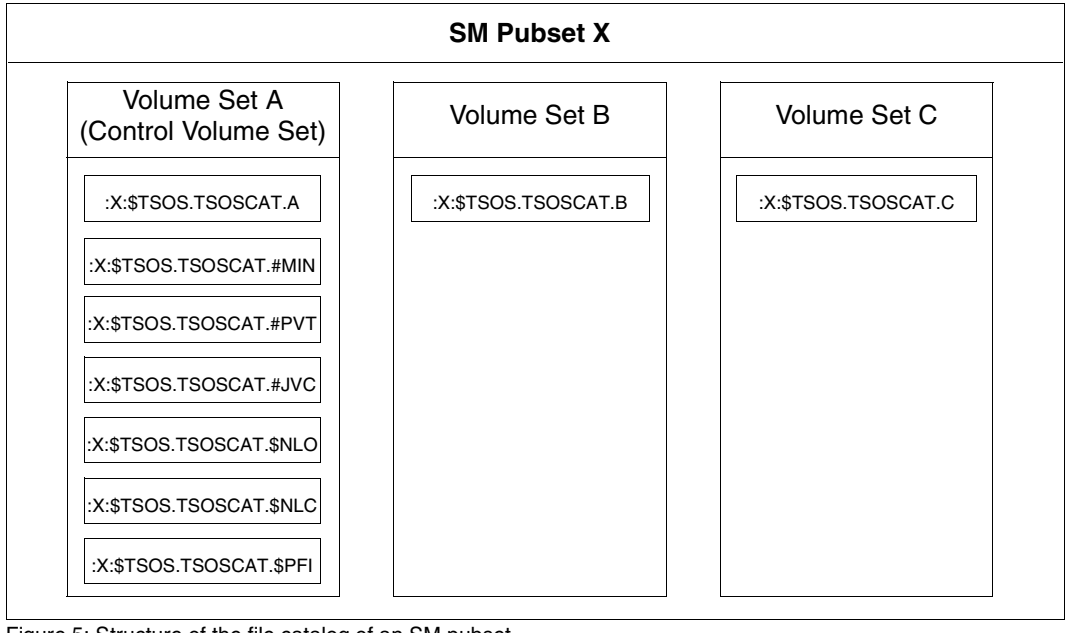


Figure 5: Structure of the file catalog of an SM pubset

When several SF pubsets are merged to form an SM pubset, you must remember that the values for the CMS buffers are adapted in the MRSCAT .
When the values are simply added, better catalog performance generally results on account of the better equalization of load fluctuations on the catalog.

The following global subcatalogs are on the control volume set and should be divided up there among several volumes:

- global subcatalog for migrated and empty files
- global subcatalog forfiles on private volumes
- global subcatalogs for job variables

Other subcatalogs on the control volume set ($NLO, $NLC, $PFI) are used exclusively for data security.

The central user catalog is also on the control volume set.

The structure of the individual subcatalogs corresponds to that of the TSOSCAT file of an SF pubset. Therefore, when creating these subcatalogs the same recommendations apply as for creating TSOSCAT on SF pubsets.

Use of SCA is not possible; accelerated catalog access is already implemented internally.

7.2.5.3 Paging areas

Paging areas are cataloged as files with the name:
`:<catid>: $TSOS.SYS.PAGING.<vsn>`

Using the parameter service, a choice can be made from all available paging files every time the system is run. This enables modifications made necessary by various workload conditions to be carried out in each session.

All pubsets containing at least one paging area for the imminent session must be online before system initialization (before startup). These pubsets are used exclusively by the startup procedure, but are not implicitly imported (i.e. the paging areas are used even if the pubset is not imported). During the session, these pubsets cannot be used by other BS2000 systems (as “shared pubsets”).

Paging areas may also be contained on SM pubsets.

Creating paging areas

When a pubset is created, the paging files are reserved on each of the volumes of this pubset intended for paging, and an entry is made for each one in the file catalog of this pubset. A volume can only contain one paging file (although the file can have several extents).

The size of a paging area depends on the number of programs executing simultaneously and their virtual size. To this must be added the system address space currently generated.

The following formula is recommended for creating paging areas:

```
Size = (Number of users * Virt. program size + System address space) * 2
```

After normal operation has started, a check should be carried out (report “Paging Area Utilization” in the report group MEMORY of SM2) in order to ascertain the actual file size required.

For programs executing simultaneously, the size of the paging area should be several times that of average requirements, but should not total more than twice the maximum requirement.

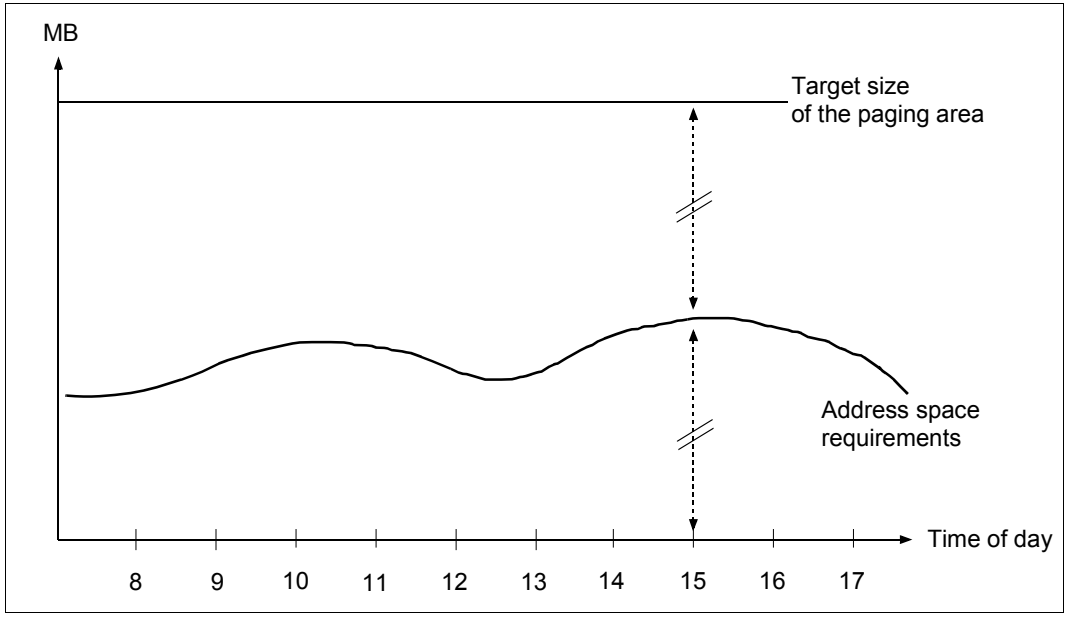


Figure 6: Allocation of the paging area

The **access frequency** of the paging area depends on:

- the operating mode (TP, interactive, batch mode)
- the size of main memory
- the speed of the server

Interactive mode

Interactive mode is characterized by a large number of tasks which, however, are relatively small as regards working set requirements (50 - 150 4 KB pages).
Since few page requests are normally made to the individual user tasks per dialog step (5 - 10), there is no significant drop in performance visible to the user, even when the total paging rate is high.

However, a paging operation not only involves a time delay, it also places considerable demands on the system.

Therefore, for reasons of efficiency, paging should take up no more than 3% of the performance capacity of the server. With high-powered servers even this percentage is too high, and requires an uneconomically large number of volumes with paging areas.

TP mode

TP mode is characterized by a small number of tasks with relatively large working set requirements (several MB in size).

In order to fulfill a particular user function, a large number of pages are addressed, often over several tasks (especially in the case of data base/data communication applications).

Increasing the paging rate usually leads to a severe drop in performance as viewed by the user. Due to the heavy page requirements for fulfilling a function, the paging rate climbs dramatically with even a slight shortage of main memory. At this point, an insignificant additional increase in workload is sufficient to cause a drop in performance (exponential increase in paging rate: the system “capsizes”).

TP mode is considerably more sensitive to main memory shortages than the other operating modes. Consequently, the reference values for the upper limit of the paging rate must be correspondingly lower (see section “Standard values for BS2000/OSD servers” on page 331).

From an economic point of view, it is always desirable to reduce the paging rate to values less than those suggested above.

In order to be better able to react to sudden overloads, utilization of a volume which is used exclusively for paging activities should not exceed 10%. With a read hit rate of 70% this volume will handle approx. 30 paging I/O operations per second. As normally no more than 10 - 20 paging I/O operations should be performed (see the section “Standard values for BS2000/OSD servers” on page 331), it is generally not necessary to distribute the paging area over more than 3 to 6 physical devices.

Location of the paging areas

Paging files should be the same size on all volumes (if possible only one extent) so that a constant I/O workload can be achieved on all volumes. As a rule the I/O rate correlates to the size of the paging file.

If possible, paging areas should never be activated on volumes containing the file catalog or a SYSEAM extent. This also applies to volumes which are heavily utilized by user files.

Further details can be found in the “Introductory Guide to Systems Support” [10] (under “Memory management”) and in the “System Installation” manual [31] (under “Handling important system files”).

Sample calculation of size and distribution of the paging areas

Assumptions

TP mode:	1200 user in over 30 UTM tasks and 5 database tasks each with an average program size of 120 MB
System address space:	240 MB
CPU:	S server (mono, 240 RPF)
Disk storage system:	Symmetrix DMX-3
Paging rate:	Max. 100 I/O operations per second As explained on page 164, a paging volume should not perform more than 30 paging I/O operations, i.e. (3) - 4 volumes with paging areas are required

Calculation

```
(35 * 120 MB + 240 MB) * 2 = 8880 MB
```

In order to achieve an even distribution of workload over the volumes, the paging area must be divided into extents of equal size, i.e. 2220 MB per volume.

Selection of specific volumes for paging during startup

If the paging areas are not selected via the parameter service, the areas created on the home pubset are used.

Specifications in the system parameter file determine which paging areas are actually to be used for the following session. If, after planning the workload of an IT system, the load profile is already known before system initialization, the paging parameters can be set in a realistic fashion according to this profile using the parameter file.
As a rule, no paging area should be reserved on the first volume of each pubset (VSN=xxxx.0), as the file catalog relevant to that pubset is always stored here and, if applicable, a high I/O rate can be expected with regard to CMS.

Extending and reducing the paging area

The paging area can be extended as well as reduced in size. This means that it is possible to relocate the paging area from the performance-critical home pubset to a pubset with a lower workload. For more information, refer to the section entitled “Paging area” in the “Introductory Guide to Systems Support” [10].
You will find information on increasing the paging area or reducing the paging activities to selected volumes in the descriptions of `/EXTEND-PAGING-AREA` and `/MODIFY-PAGING-AREA-ATTRIBUTES` in the “Commands” manual [15]).

7.2.5.4 SYSEAM file

The SYSEAM file is used to store temporary data (access method EAM). SYSEAM can be created on any pubset.

Users access the SYSEAM file which resides on their user default pubset. If there is no SYSEAM file available on the user default pubset, the SYSEAM file of the home pubset is accessed.

The optimum size of SYSEAM can only be determined by observing the mean occupancy over a sufficiently long period of time (for information on making the primary allocation, see appendix, “Performance relevant system parameters”, parameters EAMMEM, EAMMIN, EAMSEC, EAMSIZ, page 344ff).
Observation is carried out with the software product SPCCNTRL (Space Control). Heavy demands are placed on the SYSEAM file, especially in the case of interactive program development in interactive mode: frequent compilation runs have a considerable effect on both size and access frequency.

For monitoring access frequency, use of SM2 is recommended (file monitoring).

Location of SYSEAM

To avoid SYSEAM becoming a bottleneck when there is a large number of users in conjunction with powerful CPUs, it makes sense to distribute it across several pubsets.

If possible, no SYSEAM extents should be stored on volumes with paging areas.

Since all I/O requirements are subject to caching (write hit = 100%) in disk storage systems, it is not necessary to distribute the SYSEAM file across several volumes in a pubset.

Any increase in the size of the SYSEAM file workload is generally reflected by a corresponding increase in the size of the file. If this is the case, it is recommended that the volumes required are reserved exclusively for SYSEAM and that, with regard to the I/O rate, the upper workload limit of 30% is approached. In this case, a volume manages around 300 I/O operations per second.

SYSEAM peak loads

The main users of SYSEAM are the compilers, the software product LMS and the linkage editors (binders) BINDER and DBL.

Systems support should know the times when these program services are called. Then decisions can be made whether additional pubsets need to be created and when they should be imported.

If a high proportion of the workload consists of program services, the TSOS macro libraries, the user macro libraries and the MODLIB files should not be placed on volumes which also contain SYSEAM extents.

If sufficient main memory is available, it makes sense to place part of the SYSEAM area in class 4 memory (this applies only for the home pubset) using the system parameter EAMMEM (see appendix, “Performance relevant system parameters”, page 344), thus saving on SYSEAM I/O operations. This permits compilation and binder runs to be accelerated correspondingly.

Measures which can be taken after SYSEAM secondary allocation:

When the console message `DMS0852 (&00): (&01) PAM SEITEN AND (&02) EXTENTS` (where `&00` is the name of the SYSEAM file) is issued in a session, the file size exceeds the value of EAMMIN or the pubset-specific entry for MINIMAL-SIZE in the MRSCAT.

This message, which is issued whenever the file is extended or shortened, should appear only during peak workloads. If it is issued frequently, it means that the value selected for the system parameter EAMMIN or for MINIMAL-SIZE in the MRSCAT entry is too small. In order to avoid secondary allocations, the size of EAMMIN must be adjusted and the file extended accordingly. After system initialization, the files can be erased and then newly created again. During this time, no users may access the SYSEAM files. A renewed startup is not necessary afterwards.

The SYSEAM files are reset to the size of the system parameter EAMMIN via a system cold start.

7.2.5.5 Message files

Two areas must be taken into consideration if system messages are to be created and output efficiently in BS2000/OSD:

- message management

  All the messages which could be needed while the BS2000 system is in use are stored in a number of ISAM files. Message files which are essential for the system run and ones which have been specified by means of parameters are made available at startup (system parameters `MSGFILii` and `MSGNOFL`, see appendix, “Performance relevant system parameters”, page 343). Further message files can be specified in the MIP parameter file (`SYSPAR.MIP.<version>`). However, they are enabled only after the MIP subsystem has been loaded. Further message files may be added or removed during the session. See the “Introductory Guide to Systems Support” [10].

- system-internal message handling

  Each message file created with the utility routine MSGMAKER contains the message text and a number of attributes (for searching for the message text) as well as the meaning and the response text (for `/HELP-MSG-INFORMATION`).

When the system is being set up, the files is opened and the management section is read into a work area in class 4 memory.
If users make requests for system messages to be output in the course of processing, the following actions are taken when handling these message outputs:

- on the basis of the accompanying message code, a check is made as to whether the message is already in the message buffer or whether it needs to be read using the ISAM access GETKY
- on the basis of the weight code, a check is made as to whether this message is to be output on the output medium console or whether it is to be suppressed
- variable sections of the text are updated and output is initiated (if suppression has not been specified). At least one change of task is necessary for this.

Output media can be any of the following:

- the main console and/or the subconsole
- and/or SYSOUT/SYSLST of the user task
- and/or a user area (buffer)
- or a specific BCAM application (e.g. OMNIS).

Every request for output of a message to the master console and/or subconsole (even if output is to be suppressed) is also written as a record to the CONSLOG file.

Saving message files

Message files which are only required for special applications should not be placed on the home pubset disks (where they merely occupy space unnecessarily), but rather on the pubset belonging to the particular application. For current requirements, disks can be attached with /IMPORT-PUBSET or the files can be restored from backup data (e.g. using HSMS). The message files are then transferred into the active system using /MODIFY-MSG-FILE-ASSIGNMENT or /MODIFY-MIP-PARAMETERS.

Individual messages (or the contents of complete message files) are integrated into the message files required for the session.

Number of message files

The messages of special software products can remain in the software product's message file. They are automatically installed with IMON.

The number of message files managed as an integral part of the overall system, and which of these should be made available in which session and perhaps removed after they are no longer needed, is left to the discretion of systems support.

A smaller number of additional message files facilitates management and requires less system address space. A larger number of message files allows the system to be tailored more exactly to the workload conditions.

Suggestions for limiting or speeding up the output of system messages

1. Suppression of certain system messages

   Preparing, updating and outputting messages is necessary for the system to run properly, but also simultaneously increases its workload. The way in which messages are handled should be tailored to the particular requirement. For example, the request to output a message can be revoked if the loading of a program (/START-EXECUTABLE-PROGRAM) has no relevance for the operator's activities and no log of the event is evaluated (set system parameter EACTETYP to 0, see page 343).

   If different applications are processed sequentially during data processing, the number of outputs to the consoles which are to be suppressed should be chosen to suit each individual application.

   This means that:
   All messages which need to be handled differently with regard to output suppression are given different weight codes in the message files. /ADD-CONSOLE-FILTER is used to set the filter level most suitable for each individual type of workload. Productive use, program development, test operation, diagnosis following faults or after version changeover etc. and any mixture of these are all different types of workload.

The following recommendations should be observed:

Unnecessary outputs to the consoles add to the workload on the system and complicate the log. The time spent designing a plan for allocating weight codes for the individual message records will, in due course, prove to be time well spent.

An alternative method for suppressing output on the basis of the weight code is to use the message subscription or message suppression function (`/MODIFY-MSG-SUBSCRIPTION`, `/SET-MSG-SUPPRESSION`). This makes it possible for a message-receiving entity to subscribe to or suppress individual messages. Both possibilities can also be used in combination:

a) First you roughly define the message set by allocating routing codes and specifying weight codes.

b) Then you carry out the fine-tuning via subscription/suppression.

2. Selective output of system messages

   The following applies to the assignment of routing codes (parameter service, statement `SET-CODE`):

   – With the appropriate routing codes, consoles can receive operating messages.

   – The passing of messages is performed by a system-internal communication mechanism (with the aid of bourses) and incurs a noticeable system overhead.

   – When assigning the routing codes, you should always check which messages or commands must be available on the consoles, so as to avoid superfluous messages by restricting assignment.

   – The message delivery can be optimized by means of the message subscription/suppression mechanism.

3. Speeding up message output with DLAM (Dynamic Loadable Access Method)

   Systems administration can set the DLAM flag for any message.These flags are stored in the directory of each message file. At startup, the directory of all message files which, according to the start parameters, should be available at “system ready” (system parameter `MSGFILii`, see page 343) is evaluated and placed in the range assignment lists (in the message handler tables). Following this, all messages with the DLAM flag are read from the appropriate files.
   (MSGMAKER utility routine, statement/mask: `ADD-MSG` and `MODIFY-MSG`)

Advantage

The address space requirements for buffered message texts do not change during the system run (static format from startup). The path lengths are shortened if DLAM flags are set as it is then no longer necessary to check whether a message text which is currently required is already buffered. The physical I/O operation is brought forward from the system run to the startup phase.
Messages are created in class 5 memory at startup. No physical I/O operations are executed when a message is called during the system run.

Effort

Selecting and flagging the individual messages takes up expensive processing time. The use of DLAM is recommended on large servers with a high throughput rate.

4. Speeding up message output with LOCAL-DLAM (Local Dynamic Loadable Access Method)

   If a message has the access method LOCAL-DLAM (assigned with the MSGMAKER utility routine), it is copied into class 4 memory at activation.

   Advantage

   As with access with DLAM, no physical input/output is necessary. In addition, there is no access to the central MIP task - all actions are executed under the job-source task. This not only reduces the system overhead, but also prevents possible lock situations between the central MIP task and other job-source tasks.

5. Speeding up the message output using the system parameters MSGCENTL and MSGCENTN (see page 343)

   MSGCENTL defines the maximum length of messages and MSGCENTN the number of messages which can be buffered in class 4 memory in order to reduce the number of file accesses.

   The amount G of class 4 memory required is calculated as follows:

   ```
   G = MSGCENTL * MSGCENTN [Byte]
   ```

7.2.6 Creating user files

User files are generally created and processed by users in the course of their session. It must be noted that each execution of the following operations increases the workload on the file catalog: creating and deleting a file, opening and closing a file, changing the size of a file.

The main criteria for choosing the data volume and the logical operating mode are the size of the file, the access frequency and the frequency of accesses to the catalog during file processing.

Since BS2000/OSD V8.0 the input/output size when copying files using `/COPY-FILE` has been 128 Kbytes (previously: 64 Kbytes). When large files are copied, this results in enhancements of up to 20% in the CPU requirement and up to 15% in the runtime.

With small files the runtime can increase. When SCA is used (see page 150), the runtime improves by approx. 5%.

File size and access frequency

When data resources are planned, the frequency with which files are accessed has a considerable influence on system performance.

- To keep the load on the volumes as low and uniform as possible, (large) files which are not accessed frequently should be placed on volumes on which frequently accessed files have already been installed and vice versa.
- Frequently accessed files should always be placed on separate volumes (especially in the case of write-intensive applications). This is only practical if a number of tasks are involved.
- Input/output files which are accessed alternately within the same processing run should be on separate volumes.
- On private volumes, users can explicitly determine the position of files. On public volumes, this can be done only by systems support or by selected users (`PHYSICAL-ALLOCATION` operand in `/MODIFY-USER-ATTRIBUTES`). The position of large files should be specified explicitly, if possible.
- It makes sense to use frequently accessed files belonging to time-sensitive applications as HIPERFILEs (see section “Working with HIPERFILEs” on page 176).
- As a rule loads with normal to high write rates are managed efficiently in the disk controllers using familiar caching resources. However, bottlenecks can occur (write delay) in the case of loads with very high write rates to particular hotspots/files.

The following tuning options, which can also be combined, are provided to counteract this effect:

– The use of hiperfiles on global storage enables the shortest I/O times to be implemented with maximum throughput (see also page 176).

– Distributing the hotspots over multiple logical volumes / physical devices enables the I/O bandwidth to be multiplied. The following are provided for this purpose:

  a) Hardware measures:
  Use of RAID1/0 or RAID5 (see page 49) with additional software support through the use of PAV (see page 55).

  b) Software measures:
  BY distributing the hotspot or file over multiple files, extents or volumes. An example of this is the distribution of KDCFILE from UTM to multiple files (see page 193).

Frequency of catalog access

As already mentioned in section “Logical operating modes of disks” (starting on page 145), the internal outlay for processing calls with catalog access is at its smallest for public volumes or pubsets, followed by private volumes. It is at its highest for shareable private disks.
Throughput for the individual logical operating modes is therefore dependent on the ratio of management IO operations to “productive” I/O operations.

Public volumes or pubsets are most suited to files subject to frequent catalog access (procedures, ENTER files, a large number of OPEN/CLOSE operations in quick succession).

Files with a low frequency of catalog access can be stored on private volumes.

The high frequency of catalog accesses is only acceptable for shareable public volume sets when there is joint access by several servers.

In order to reduce the number of management I/O operations, the following points should be borne in mind:

- select the primary and secondary space allocation in `/CREATE-FILE` to suit the file size expected
- avoid erasing and creating a file unnecessarily (job command sequence: `/CREATE-FILE`, `/DELETE-FILE`, `/CREATE-FILE` for the same file name)
- use OPEN/CLOSE operations sparingly (the commands `/COPY-FILE`, `/CALL-PROCEDURE`, `/INCLUDE-PROCEDURE`, `/ENTER-JOB`, `/ENTER-PROCEDURE`, `/START-(EXECUTABLE-)PROGRAM`, `LOAD-(EXECUTABLE-)PROGRAM` and `/ASSIGN-SYSDTA` also trigger OPEN/CLOSE operations.)

Reorganization

With larger systems, it is advisable to reorganize all files on a regular basis (e.g. every three weeks). This reorganization counteracts the splitting of files into a number of small extents, which cause an increase in positioning times. The reorganization is considerably simplified by the software product SPACEOPT.

SPACEOPT clears up the fragmentation through optimum relocation of the file extents on the volumes of a pubset. Files can also be reorganized if the free space available is smaller than the size of the file, i.e. SPACEOPT also permits local enhancements on the basis of the free disk space available.

SPACEOPT can adjust the size of BS2000 volumes to the size of the LUNs (Logical Units) with which they are implemented in ETERNUS DX and Symmetrix disk storage systems. Such an adjustment can be required after disk migration using DRV, in which the source and target disks must have the same capacity. Information on disk migration, e.g. from D3475 to D3435, is provided in the “DRV” manual [7].

A further application is the expansion of a BS2000 volume following the expansion of a LUN through the inclusion of additional disks in the ETERNUS DX or Symmetrix disk storage system. This function is only available for the disk format D3435 (FBA data format).

SPACEOPT works on a volume basis, i.e. tasks can be specified for individual or all volumes of a pubset. SPACEOPT offers options for assessing the occupancy and fragmentation status and can be used during operation.

SPACEOPT can also move files if they are opened.

It is advisable to use SPACEOPT in off-peak times, as it can incur a high I/O workload depending on the degree of fragmentation of the volumes.

The performance of other BS2000 guest systems or of other servers which execute I/O operations to/from the same physical disk can be negatively influenced by reorganizing using SPACEOPT. See also section “Configuration in BS2000/OSD” on page 52.

You will find detailed information on SPACEOPT in the “SPACEOPT” manual [30].

7.3 Working with HIPERFILEs

The objective of the HIPERFILE (High Performant Files) concept is to increase the speed of file accesses and to avoid or eliminate I/O bottlenecks by buffering the data in the main memory or global storage using the software product DAB (Disk Access Buffer).

The purpose of DAB (Disk Access Buffer) is to provide faster access to disk files by means of caching on fast media, namely main memory (MM) or global storage (GS).

Main memory is a volatile storage medium that is suitable primarily for the caching of read accesses as well as for write jobs to temporary files. Unsaved write data in the buffer is lost after a system crash if it is on a volatile storage medium.
Global storage (available on S servers) is also a volatile storage medium. However, it can be upgraded to a non-volatile storage medium by means of optional add-ons (battery or uninterruptible power supply). Global storage is thus particularly suitable for failsafe fast writing.

There are two variants of the HIPERFILE concept, namely ADM PFA caching and user PFA caching:

- ADM-PFA caching (administrator-controlled Performant File Access)

  This is a variant of the HIPERFILE concept in which systems support uses commands to specify which files or volumes are to be buffered in which cache medium and which caching mode (read, write or read/write mode) is to be used.

- User PFA caching (user-controlled Performant File Access)

  In this variant of the HIPERFILE concept, the HIPERFILEs are embedded in the Data Management System (DMS). Users decide which of their files on a pubset are to be HIPERFILEs and are thus to benefit from a cache set up by systems support.

In addition, disks whose access times are considerably shorter than those of a disk storage system can be emulated in global storage. This disk emulation is managed by the GSVOL subsystem (GS Volumes in GS).

7.3.1 Caching modes

When the data requested is available in the cache, a cache hit is said to have occurred; otherwise, the attempted access is referred to as a cache miss. The ratio of cache hits to the total number of accesses is referred to as the cache hit rate. The higher the hit rate, the greater the benefit provided by the cache. The cache in the main memory/global storage is examined here; the cache integrated in the disk storage system works “invisibly” in the background.

The following caching modes are available (in the diagrams below, accesses are depicted by arrows, with read accesses on the left and write accesses on the right):

Read cache

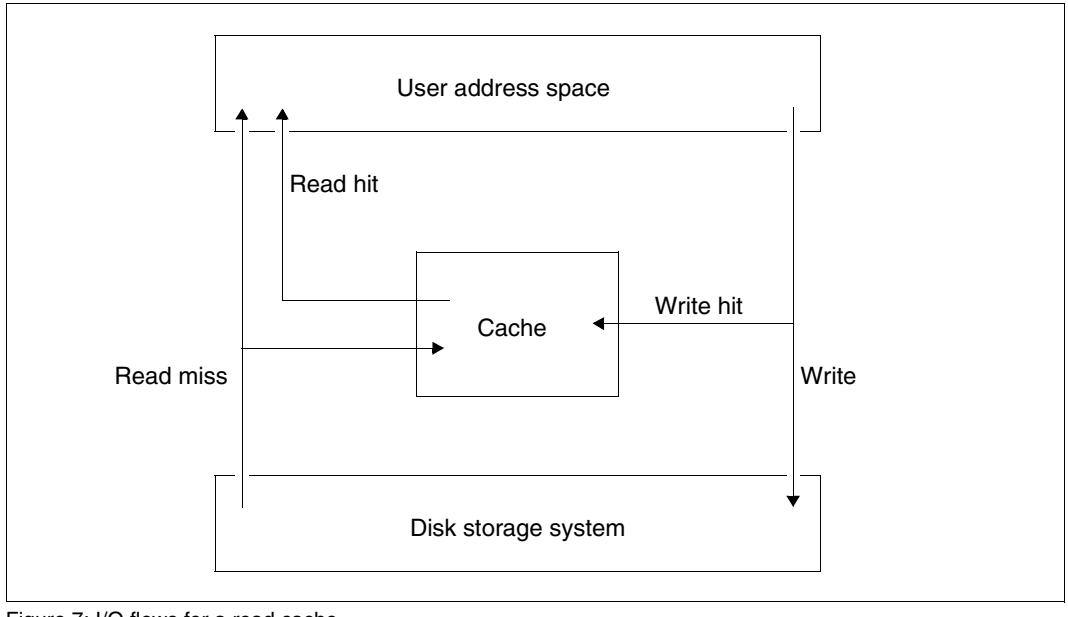


Figure 7: I/O flows for a read cache

Read hit: The data is available in the cache when accessed and is read from there.

Read miss: The data is read from the disk storage system and entered into the cache at the same time so that subsequent read requests can be handled from there.

Write: Write accesses are always to the disk storage system. If the relevant data block is already in the cache (“write hit”), the contents of the block are updated without making the write operation any faster.

Write cache

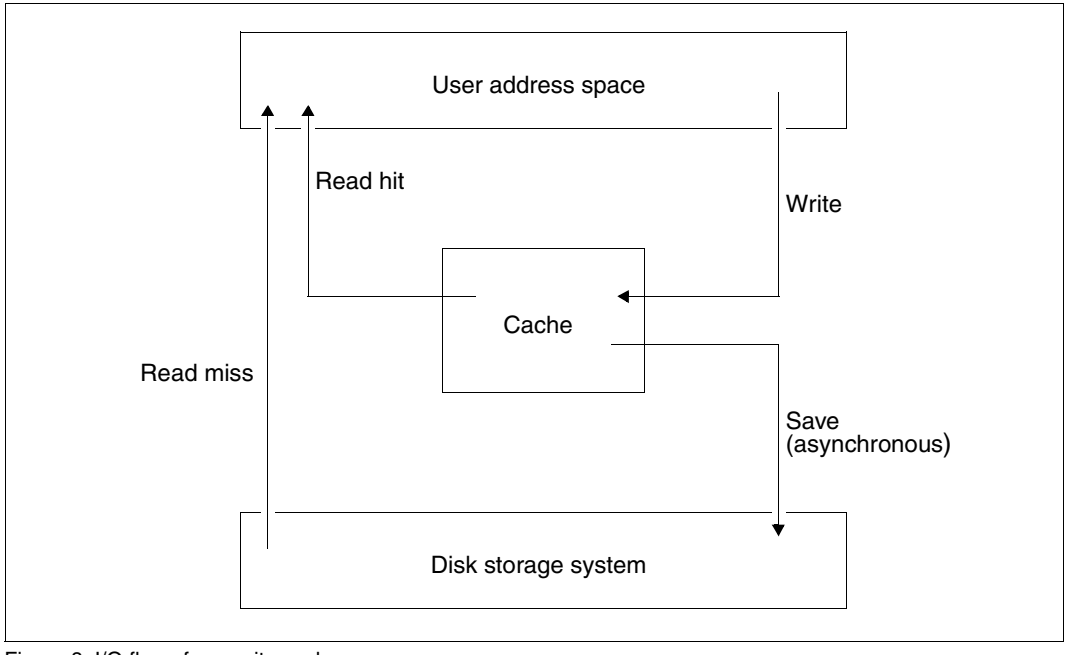


Figure 8: I/O flows for a write cache

Read hit: The requested data is read directly from the cache (as in a read cache)

Read miss: The data is read from the disk storage system but not entered in the cache.

Write: In both main memory and global storage, the data is always written to the cache, regardless of whether the data block is already there or not. The application writing the data considers the output process to have been concluded as soon as the data has entered the cache. The actual writing of the data to the disk storage system is generally initiated immediately afterwards.

Read/write cache

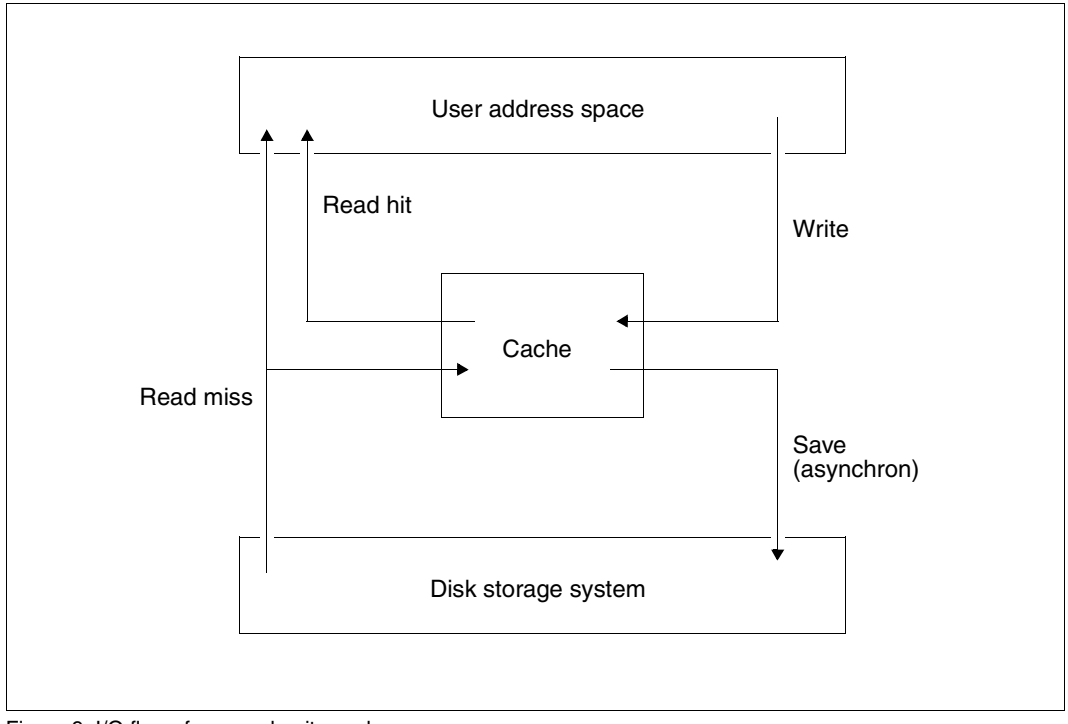


Figure 9: I/O flows for a read-write cache

Read hit: as in a read cache

Read miss: as in a read cache

Write: as in a write cache

7.3.2 ADM PFA Caching

In **ADM PFA caching**, systems support controls the creation of cache areas and the allocation of selected data areas (a cache area is always a subarea of a cache that represents a separate administration unit). Systems support can use `/START-DAB-CACHING` to create independent cache areas and thus to separate selected data belonging to different applications. A cache area can also be created for entire volumes. You will find a detailed description of this in the manual “DAB” [5].

The term **AutoDAB** refers to a variant of DAB caching. It involves automatic and intelligent procedures which, on the one hand, greatly reduce the work of systems support when administering the caching and, on the other, allow the efficiency of the DAB caching to be further increased.
A DAB cache area can be set up as either an “automatic” or a “non-automatic” cache area.

Any of the following three settings is possible when using the cache buffer:

a) The authorized user can specify which files are to use the cache area created in which cache mode.

b) Systems support can specify that all user files of a pubset are to use the cache area.

c) Systems support can create an automatic cache area in which the optimum selection of the files to be supported occurs automatically. This selection is permanently monitored and, if necessary, corrected. AutoDAB enables entire pubsets to be covered by a cache area.

DAB manages each cache area in the form of so-called cache segments whose default size is 32 KB for automatic cache areas. The default size of a cache segment in automatic cache areas is 32 KB. For non-automatic cache areas it can also be specified by systems support (4 KB, 8 KB, 16 KB or 32 KB). Automatic caching must be stopped during the modification.

The first time an area (file) supported by DAB is read-accessed, DAB reads from the disk storage system page-chained, adjacent data blocks, the number of which depends on the set cache segment size.

Spatial locality

In read accesses with good spatial locality (particularly for sequential accesses), large segments (= 32 KB) allow a high throughput of I/O jobs; in this case, there are a large number of read hits and the accesses to the disk storage system are reduced to a minimum.
In read accesses with low spatial locality (random access, the most frequently used access type in database mode) it is advisable to select small segments (= 4 KB).
AutoDAB recognizes good locality and, when it finds it, automatically performs a prefetch (even if a smaller cache segment was set).

Chronological locality

The blocks read into the cache remain there until DAB is terminated or this memory area is used for another cache segment.
Preemptions are managed on the LRU (least recently used) principle, which means that a high read hit rate can be achieved given good chronological locality of the accesses even with a relatively small cache area.
For performance-critical applications whose read accesses have low chronological locality, DAB offers the possibility of resident buffering.

In the case of a read/write (or write) cache, write jobs are always entered in the cache directly, regardless of whether or not the data block is already in the cache. As far as the application is concerned, the output operation is completed as soon as the data is written to the cache. DAB saves the data on the disk storage system later.

In order to be able to use the cache medium GS with DAB, DAB partitions must be created in GS (see the “Introductory Guide to Systems Support” [10] and the “DAB” manual [5]).

When the cache medium MM is used, it must be noted that DAB creates the system buffer in main memory (i.e. main memory must be large enough).

Systems support uses the following commands to control the use of ADM PFA caching with DAB:

Command	**Function**
START-DAB-CACHING	Creates and assigns attributes to cache areas
SHOW-DAB-CACHING	Outputs information on the current DAB configuration and access counters (hit rates)
STOP-DAB-CACHING	Saves cache areas on disk and deletes them
MODIFY-DAB-CACHING	Dynamically modifies parameters of a DAB cache area
FORCE-STOP-DAB-CACHING	Deletes a DAB cache area without backing up the write data

Recommendations on use

- General use of AutoDAB

  The selection of the data set to be buffered and the setting of the prefetch factor should occur through the automatic caching. It is therefore recommended that you operate the performance-critical volumes or pubsets with AutoDAB.

  As a rule, the following procedure is simple and effective:

  1. Determine all performance-critical applications
  2. Determine all volumes/pubsets that are accessed by these applications
  3. Combine all these volumes/pubsets to form an automatic ADM PFA cache area (useful if nothing is known of the access frequency of individual pubsets or Define an automatic PFA cache area for each individual pubset (see also section “User PFA Caching” on page 184).

- Defining the size of the cache area

  The size of the cache area should be at least 1% of the size of the data set to be supported (files processed in parallel). With predominantly sequential processing, this value can be lower, while with predominantly random processing it may be too small. If the cache size is not sufficient, this will be reported by AutoDAB as a console message.

- Speeding up of all write accesses:

  Regardless of the locality, when a read/write (or write) cache is used, DAB speeds up all write accesses as long as there are free buffers available.
  As a result, short-term peak loads can be dealt with and fast file access times achieved. However, since DAB has to save the write data buffered in the cache to the disk storage system, in the case of write applications the input/output system can only be relieved when the locality of the accesses is sufficiently high.

- Home pubset

  The home pubset cannot be buffered in the DAB cache media MM and GS by means of PFA. Although this is possible with ADM PFA caching, it should only happen with a pure read cache or in nonvolatile GS with a write cache only for files that are not accessed before System Ready.

- SM pubsets

  The data of an SM pubset cannot be buffered in a DAB cache medium with ADM PFA caching (since the storage location, i.e. the volume set, of the file can be changed dynamically).

- Database pubsets

  With pure database pubsets it may be possible to improve the “chronological locality” by selecting a smaller segment size: e.g. 8 KB instead of the default value 32 KB (`/START-DAB-CACHING`, parameter `CACHE-SEGMENT-SIZE`).

In practice the reduced I/O times usually lead to a considerable increase in throughput and thus to preemptions of the load and the CPU utilization.

In the case of applications which cause a higher I/O rate additional disks should be used to increase the throughput. (The use of DAB for write IOs is not always recommended.)

You will find more detailed information on using DAB in the “DAB” manual [5].

7.3.3 User PFA Caching

In user PFA caching (PFA: Performant File Access), systems support decide which pubsets are assigned a cache area, which properties this cache area is to have and which users are to benefit from the caching of these pubsets.
You will find a detailed description of this in the “Introductory Guide to Systems Support” [10].

PFA from the user's perspective

Depending on the definition of the cache area, the use of a PFA cache for a pubset can be either global for all files or selective for specific files:

- If the cache area is defined with the attribute `CACHED-FILES=*BY-SYSTEM` (i.e. AutoDAB is in use) or `CACHED-FILES=*ALL` (all user files of the pubset are subject to the caching), the user does not need to perform any further actions in order to benefit from caching on this pubset.
- If the cache area is defined with `CACHED-FILES=*BY-USER-SELECTED`, the caching as described below only affects files with the corresponding attribute.

The commands `/CREATE-FILE`, `/MODIFY-FILE-ATTRIBUTES` and `/ADD-FILE-LINK` and the macros `FILE` and `CATAL` can be used to assign the following file attributes:

Operand	Meaning
`PERFORMANCE`	Specifies caching behavior (main memory/global storage):
`= *STD`	no caching
`= *HIGH`	caching with preemption on the basis of LRU
`= *VERY-HIGH`	caching without preemption
`= *USER-MAX`	Highest performance attribute for which the user has authorization
`USAGE`	Improved performance requirements apply:
`= *READ-WRITE`	to read and write operations
`= *READ`	to read operations only
`= *WRITE`	to write operations only
`DISK-WRITE`	Data consistency is restored:
`= *IMMEDIATE`	after each write operation (prerequisite: nonvolatile cache medium)
`= *BY-CLOSE`	not until after CLOSE processing
`= *STD`	`*IMMEDIATE` applies for permanent files, `*BY-CLOSE` applies for temporary files

If the same files are opened and closed several times during continuation processing (generally in batch mode), the user can further improve throughput by doing the following when closing the file:

- preventing the invalidation of the read data in the cache
- suppressing the saving of the write data in the cache (only recommended for temporary files in the case of a volatile storage medium).

Another advantage is that reopened files do not have to be stored in the cache again. This type of PFA caching (HIPERBATCH = high-performance batch processing) is made possible by the operand `CLOSE-MODE= *KEEP-DATA-IN-CACHE` of `/ADD-FILE-LINK` and the `KEEP-DATA-IN-CACHE` operand of the `CLOSE` macro.

PFA from the point of view of systems support

In order for the user's performance requirements to become effective, corresponding authorizations must be entered in the user catalog (`/ADD-USER`, `/MODIFY-USER-ATTRIBUTES` or `/MODIFY-USER-PUBSET-ATTRIBUTES`):

Operand	Meaning
`DMS-TUNING-RESOURCES`	specifies the performance requirement

`= *NONE`
`= *CONCURRENT-USE`
`= *EXCLUSIVE-USE`
	caching not permitted for
	the requirement `PERFORMANCE=*HIGH`
	for the requirement `PERFORMANCE=*VERY-HIGH`

`PERM-/TEMP-/WORK-SPACE-LIMIT` `= *PARAMETERS(...)`	Specifies storage space quotas of the users (for permanent, temporary and work files, respectively):
`HIGH-PERF-SPACE= ...` `VERY-HIGH-PERF-SPACE= ...`	for high-performance storage space for very high performance storage space

You must ensure that the authorization and storage space quotas are entered in the user catalog of the pubset on which the file is located.

Systems support uses the following commands to control the use of PFA caching:

Command	Function
MODIFY-PUBSET-CACHE-ATTRIBUTES	Specifies the cache area attributes in the MRSCAT of a pubset/volume set
IMPORT-/EXPORT-PUBSET	Implicitly creates or deletes a cache area described in the MRSCAT
START-/STOP-PUBSET-CACHING	Dynamically creates or deletes a cache area described in the MRSCAT during a pubset session
SHOW-CACHE-CONFIGURATION	Outputs a PFA cache area configuration
SHOW-PUBSET-CACHE-ATTRIBUTES	Outputs a cache area description in the static and dynamic MRSCAT
FORCE-DESTROY-CACHE	Forces the deletion of a PFA cache area
MODIFY-PUBSET-DEFINITION-FILE	Specifies the (logical) performance attributes of a volume set of an SM pubset
MODIFY-PUBSET-PROCESSING	Implicitly creates or deletes cache areas when actively adding or removing volume sets to or from an SM pubset

The cache medium for the corresponding pubset is defined by means of the operands of /MODIFY-PUBSET-CACHE-ATTRIBUTES:

Operand	Meaning
CACHE-MEDIUM	Specifies the cache medium:
=*MAIN-MEMORY	Main memory
=*GLOBAL-STORAGE	Global storage
CACHE-SIZE=n	Size of the cache area
FORCE-OUT	Specifies how full the cache has to be before the contents are saved on disk:
=*AT-HIGH-FILLING	75%
=*AT-LOW-FILLING	25%
=*NO	No saving at intervals

In addition, the following information is important for the CACHE-MEDIUM operand:

Operand	Meaning
CACHE-SEGMENT-SIZE =*4KB/*8KB/*16KB/ *32KB	Size of a cache segment (i.e. the minimum size of the data area stored if there is a read miss): 4, 8, 16 or 32 KB (only relevant for non-automatic cache areas)
CACHED-FILES =*BY-USER-SELECTED =*ALL =*BY-SYSTEM	Specifies which files should use the cache. The user uses performance attributes to specify which of his/her files should use the cache. All user files use the cache. Automatic cache area: the automatic, intelligent caching of the AutoDAB means that the files relevant to performance are determined automatically; the prefetch factor matching the access profile of the selected files is set, and the files are monitored cyclically to guarantee optimum cache performance.

User PFA caching is required for a shared pubset when local write caching is to be supported for each server (with ADM-PFA caching, only read caching is possible).

7.4 Transaction mode and BCAM

This section describes the measures for response time optimization in transaction mode and with BCAM.

In many cases the network quality has the greatest influence on the response time behavior. Lost and corrupted packets and packet transposition force the transport system to take error recovery measures, which result in packet repetitions and reduced transmission performance. A network analysis must include not only the end systems, but also all the other network components, such a switches, routers, firewalls, gateways, etc. SNMP agents can integrate the measuring components of BS2000/OSD into such an infrastructure.

7.4.1 Phases of an OLTP transaction

The phases of an OLTP transaction which are relevant to performance (see section “Online application” on page 16) are:

1. Network runtime up to the server
2. Waiting for acceptance of the transaction by BCAM
3. Processing by BCAM and forwarding to the TP application (e.g. openUTM)
4. Waiting for acceptance by openUTM
5. Processing time of the transaction, controlled by openUTM with the following detail phases:
   a) CPU requirement of the application / of openUTM (with wait for CPU allocation)
   b) I/O processing in the application / in openUTM (with wait for the I/O operation to end)
   c) Waiting for job acceptance by database tasks
   d) CPU requirement of database activities (with wait for CPU allocation)
   e) Processing the I/O operation in the database task

   These detail phases can occur several times
6. Sending the response to BCAM
7. Waiting for auf acceptance of the response by BCAM
8. Processing by BCAM

9. Transfer of the output data for network transfer
10. Network runtime up to the client

Only phases 3 - 9 can be measured using BS2000 resources (principally using openSM2, see the “openSM2” manual [18]).

For phase 2, BCAM supplies an indicator of possible delays. openSM2 presents the times collected by BCAM in graphical form in the monitoring routine BCAM-Connection:

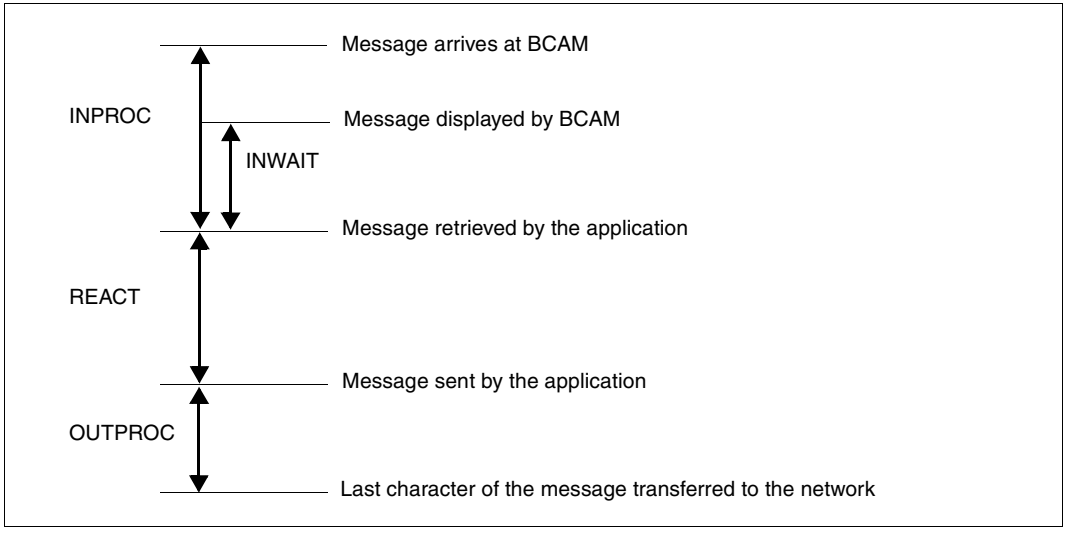


Figure 10: BCAM times

The following times are correlated:

INPROC	Phases 3 and 4
INWAIT	Phase 4
REACT	Phases 5 - 7
OUTPROC	Phases 8 and 9

7.4.2 Optimizing the various phases

This section deals with bottlenecks in the openUTM and BCAM environments which can lead to longer response times in the various phases. It indicates how these bottlenecks can be detected by means of measurements and how they can be eradicated or alleviated.

Phases 1 and 10: Network runtimes

General information on the LAN connection is provided in section “Connection to a LAN” on page 87.

Phase 2: Waiting until the incoming message has been accepted by BCAM (Inbound Flow Control):

Mechanisms for controlling the data flow protect the transport system from being flooded by data from the partners. These ensure, on a connection-specific basis, that a (high-speed) sender is matched to the processing speed of a (low-speed) recipient.

In this way BCAM protects its memory pool, which is available to all applications and connections. During phases in which heavy use is made of the BCAM pool, BCAM prevents its partners from sending messages into the BS2000 system. This can result in increased response times.

Longer or persistently high utilization of the BCAM pool can be caused by applications which fetch their messages slowly. A pool size which is too small for the processing performance required can also result in a heavy workload on the BCAM pool.

openSM2 provides messages for analyzing the utilization of the BCAM pool, see the `RESPONSETIME` monitoring routine in the “openSM2” manual [18].

BCAM itself enables you to see whether a bottleneck has occurred by entering the console command `/BCMON` and by analyzing the values of the console message `BCA0B21`.

Brief description of the procedure:

- Determine the maximum values set (optional) using `/SHOW-BCAM-PARAMETERS PARAMETER=*LIMITS`
- Activate BCAM monitoring (output of the measurement values every <n> seconds): `/BCMON MODE=ON,RECORD=RES-MEM,SEC=<n>`
- Describe the main output values in console message `BCA0B21`:

`USED-BCAM-MIN-I`	Minimum BCAM buffer allocation for incoming messages in KB
`LIMIT-ACT`	Current upper limit for the buffer allocation in KB

- Indicator for high utilization of the BCAM buffer by incoming messages: `(USED-BCAM-MIN-I) * 4 > LIMIT-ACT`

- As an alternative to or to complement the above-mentioned points: /BCSET THRESHOLD-MSG=ON can be used to request a warning message.

  Console message BCAB021 is then output if BCAM holds up the incoming messages for more than 5 seconds on account of a pool bottleneck (or if it cannot accept any send requests from the application, see phase 7). This warning message is disabled using /BCSET THRESHOLD-MSG=OFF.

- Optimization measure in the event of a bottleneck in the BCAM pool: The maximum permissible threshold value should be significantly increased using the /BCMOD RESMEM=<n> command (<n>=3*LIMIT-ACT would be ideal).

Phases 3 and 8: Processing the incoming/outgoing message in BCAM

The time values for these phases should be in the low single-digit range (in milliseconds).

Phase 4: Waiting for openUTM

The time which elapses between a job being issued to openUTM and acceptance of the job is referred to as the INWAIT time. It is determined for each application and is included in the INPROC time.

openSM2 records the INWAIT times (and also the INPROC , REACT and OUTPROC times) in 5 time intervals which are known as buckets. Buckets can only be set up globally, not on an application-specific basis. To simplify monitoring, intervals should be classified in such a manner that all times which are not acceptable are assigned to the last interval (overrun value).

Example

The normal response time is 100 - 200 ms (typical for large S servers). Short-term fluctuations up to a factor of 2 should be tolerated. However, no lengthy doubling of the response time should be tolerated.

The buckets in openSM2 should consequently be defined so that they ensure that all INWAIT times which are equal to or greater than 400 ms are counted in the overrun interval, e.g. with
/SET-BCAM-CONNECTION-PARAMETER INWAIT-BUCKETS=(50,100,200,400)

This statement defines the buckets in such a way that all wait times < 50 ms are counted in the first interval, all wait times between 50 and 100 ms in the second interval, all wait times between 100 und 200 ms in the third interval, all wait times between 200 and 400 ms in the fourth interval, and all wait times > 400 ms in the overrun interval.

The INSPECTOR or ANALYZER component of openSM2 must then be used (for every application) to monitor the measurement values of the overrun interval. The values are output as a percentage (in relation to the time values recorded in the monitoring cycle) of this application. A percentage of 10 or higher indicates that bottlenecks occurred when the jobs were accepted by the UTM tasks.

Measuring option in openUTM

The UTM command `KDCINF STAT` is used to output a number of useful UTM measurement values, see the openUTM manual “Administering Applications” [20]. The output value `%Load` provides important information for the analysis of whether the number of UTM tasks could constitute a bottleneck. This value specifies the average utilization of all UTM tasks in the last monitoring cycle. At least a short-term bottleneck is indicated by a value greater than 90 (%).

Threshold value monitoring of the number of UTM tasks using openSM2

An imminent UTM task bottleneck can be recognized from the `UTM-Applikation` monitoring report of openSM2. For this purpose the values for the duration of a transaction in seconds (`DT`), the number of transactions per second (`TX`), and the number of UTM tasks for the application (`UT`) must be ascertained from this report.

This enables the average utilization of the UTM tasks of an application to be calculated: `Load (in %) = 100 * DT * TX / UT`.

In INSPECTOR of openSM2 this calculated value can be subjected to threshold value monitoring. If a value of 90 (%) is exceeded, an email to this effect can, for example, be generated for systems support.

Optimization measure

The UTM command `KDCAPPL TASKS=<n>` enables the number of UTM tasks to be increased on an application-specific and step-by-step basis until the INWAIT times are acceptable. The optimum number of tasks ascertained in this way should be entered in openUTM’s start parameter file. It will then be effective the next time the application is started. If `<n>` exceeds the maximum value defined in the KDCDEF run, this maximum value must be increased and a new KDCDEF run must be started.

When the number of UTM tasks changes, the number of TAC classes and the number of tasks in these TAC classes must also be taken into account. Allowance must also be made for a certain buffer for load peaks.

Note

Only if appreciably more UTM tasks than required are started can this lead to slight performance losses as a result of slightly higher utilization of the main memory and CPU.

Phases 5 and 6: Processing the transaction

The time accumulated in this phase (and in phase 7) is specified as the REACT time in the BCAM Connection Report.

Tuning I/O performance when accessing the KDCFILE

In addition to the measures relating to the hardware and in BS2000/OSD which are described in this manual, the performance of applications with high transaction rates can also be enhanced in openUTM by means of optimized write accesses to the KDCFILE. To do this, the page pools and/or the restart area can be exported from the KDCFILE in openUTM. These exported areas are then distributed over a number of volumes.

Example:

The page pool is to be distributed to 2 public volumes, the restart area to 4 public volumes:

```
/CREATE-FILE FILE-NAME=<filebase>.PO1A, -
SUPPORT=*PUBLIC-DISK(VOLUME=<v1>,SPACE=*RELATIVE(PRIMARY-ALLOCATION=666))

/CREATE-FILE FILE-NAME=<filebase>.PO2A, -
SUPPORT=*PUBLIC-DISK(VOLUME=<v2>,SPACE=*RELATIVE(PRIMARY-ALLOCATION=666))

/CREATE-FILE FILE-NAME=<filebase>.RO1A, -
SUPPORT=*PUBLIC-DISK(VOLUME=<v3>,SPACE=*RELATIVE(PRIMARY-ALLOCATION=300))

/CREATE-FILE FILE-NAME=<filebase>.RO2A, -
SUPPORT=*PUBLIC-DISK(VOLUME=<v4>,SPACE=*RELATIVE(PRIMARY-ALLOCATION=300))

/CREATE-FILE FILE-NAME=<filebase>.RO3A, -
SUPPORT=*PUBLIC-DISK(VOLUME=<v5>,SPACE=*RELATIVE(PRIMARY-ALLOCATION=300))

/CREATE-FILE    FILE-NAME=<filebase>.RO4A, -
SUPPORT=*PUBLIC-DISK(VOLUME=<v6>,SPACE=*RELATIVE(PRIMARY-ALLOCATION=300))
```

In addition, the following parameters must be modified in the MAX statement in KDCDEF: `PGPOOLSFS=2`, `RECBUFFS=4`.
The files defined above are then used in the KDCDEF run. In this case the KDCDEF program may modify the values for `PRIMARY-` and `SECONDARY-ALLOCATION`. Without the aforementioned commands, KDCDEF would create the files itself (without volume assignment).

The new files are used after openUTM has been restarted.

Controlling the UTM jobs by means of TAC classes

Similarly to category control using PCS (see section “Introduction to PCS concept” on page 222), transactions in openUTM can be assigned to so-called “TAC classes”. These TAC classes can be controlled using two methods which cannot be combined:

- Priority control
- Process limitation

Details of job control in openUTM are provided in the openUTM manual “Generating Applications” [21].

Recommendations for use:

- When the TACs are distributed to TAC classes, it must be ensured that higher-priority TAC classes are not held up by TACs from lower-priority TAC classes (use of blocking calls).
- You are recommended not to define more than 3 or 4 TAC classes.
- Priority control is used above all to ensure that long-running TACs from low-priority classes do not hinder short-running TACs from higher-priority TAC classes in the event of (short-term) overloads. As a result, high-priority transactions are given preference using all the started processes.
- The process limitation is used to ensure that long-running TACs cannot hinder short-running TACs. The advantage of this variant is that it always guarantees that enough free UTM tasks are available for new (and important) TACs.
- In addition, TACs can be controlled by specifying the `RUNPRIO` in the TAC statement. However, this is only recommended for highest-priority TACs and must be coordinated with the priority structure of all tasks which run in BS2000/OSD.

Further performance information on openUTM:

- Monitoring the cache hit rate

  The UTM command `KDCINF STAT` (see page 192) should be used (at least occasionally) to check whether the hit rate in the UTM cache’s page pool is high enough. When the value is below 90 (%), the UTM cache may need to be enlarged (see the openUTM manual “Generating Applications” [21]).

- Use of UTM-F

  This UTM variant is suitable mainly for applications with “retrieval” accesses and no or only a few updates. It is used to avoid I/O operations in the case of reduced functionality (with regard to failsafe performance and restart).

Phase 7: Waiting before BCAM

This phase is included in the REACT time in the BCAM Connection Report (see phase 5).

As in phase 2, BCAM can be subject to the partner's control mechanisms for controlling the data flow. This can be recognized from ZWR values > 0 (ZWR=Zero Window Receive) in the openSM2 report BCAM. In such situations BCAM accepts a certain volume of data to be transferred from the application, but refuses to accept further data when a threshold value is exceeded.

BCAM also delays the acceptance of data in phases in which the BCAM pool is heavily utilized. Lengthy or permanently high utilization of the BCAM pool can be caused by connections which are subject to data flow control. A pool size which is too small for the processing performance required can also result in a heavy workload on the BCAM pool.

As in phase 2, BCAM offers a monitoring option.
If the `BCAOB21` message displays high values for `USED-BCAM-MIN-O`, the BCAM pool should be enlarged, as in phase 2.

Phases 8 and 9: Delay in transfer to the network

The time in phase 8 (100% BCAM processing) is in the single-digit millisecond range. The times for phases 8 and 9 are combined in the OUTPROC time. You are recommended to monitor the OUTPROC time like the INPROC time (see page 191). High values in the overrun interval indicate possible network performance problems (e.g. due to malfunctions in routers or switches).

8 System control

8.1 Job management

The job management system (JMS) provides a number of control options in the selection of jobs for processing.

Job classes serve to define certain job characteristics. In addition to the parameters for the selection of jobs, job classes also include statements important for subsequent processing. These parameters can furthermore be used to control the execution of other applications, such as TP or interactive applications. Therefore specification of a job class is not just important for batch processing.

Selection strategies for jobs (job scheduling strategies) selected by the user are implemented with the aid of job streams.

8.1.1 Job classes

The parameter JOB-TYPE=*BATCH of a job class specifies that this class is subject to job scheduling. Jobs of a job class to which the parameter JOB-TYPE=*DIALOG applies are started immediately, provided that this is permitted by the CLASS-LIMIT parameter.

Selection parameters for batch jobs

CLASS-WEIGHT

Designates the urgency of the jobs waiting in a job class as compared to another class (important only after system saturation occurs).

Value range <integer 1..9>; 1 is the lowest urgency, 9 is the highest urgency.

JOB-PRIORITY

Defines the importance (priority) of the waiting jobs within a job class.

Value range <integer 1..9>; 1 is the highest priority, 9 is the lowest priority.

CPU-LIMIT

Determines the CPU time which a job in this job class may use.
The parameter NO-CPU-LIMIT=*YES has the same effect as the entry PRIVILEGE=*NO-CPU-LIMIT in the user catalog.

START

Specifies the possible start times (*SOON, *WITHIN,...).
The additional condition ALLOWED=*IMMEDIATE has the same effect as the entry PRIVILEGE=*START-IMMEDIATE in the user catalog. This means that a job may be started immediately, even if the job class limit has already been reached.

If, in an XCS network, the HOST=*ANY operand is specified in /ENTER-JOB or /ENTER-PROCEDURE, the jobs are automatically distributed to the individual computers in the network (for details, see the “HIPLEX MSCF” manual [12]).

To select the target system, the criteria are considered in the following order:

1. Status of the job class

   The desired job class must be defined and active (it must not have been suspended by means of /HOLD-JOB-CLASS, for example).

2. The system must not be in a state of saturation.

3. Utilization of the job class

   As long as the job class limit (CLASS-LIMIT) has not been reached, for jobs that are to be executed immediately the difference between the job class optimum (CLASS-OPTIMUM) and the number of jobs already started is taken.

   If the job class limit has been reached or exceeded, the difference between the job class limit and the number of accepted (i.e. already started and pending) jobs is taken.

Parameters for executing batch jobs, TP applications and interactive applications

The following parameters are important when it comes to executing applications in the desired category, taking into account the relevant task attribute TP, DIALOG or BATCH (see also section “Task management” on page 203) and the desired priority.

TP-ALLOWED = *NO / *YES(CATEGORY=<name>)

DIALOG-ALLOWED = *NO / *YES(CATEGORY=<name>)

BATCH-ALLOWED= *NO / *YES(CATEGORY=<name>)

The above parameters define the task attribute for the job class. This serves to assign special internal execution parameters, optimized for the operating mode TP, interactive or batch.

If no category name has been specified, the associated default category name is assumed (e.g. for TP-ALLOWED=*YES the default category name is TP).

A category name must not be assigned to more than one task attribute.

START-ATTRIBUTE = *BATCH / *DIALOG / *TP

This parameter defines the task attribute used to start tasks of this job class. The parameter values for START-ATTRIBUTE and TP-ALLOWED, DIALOG-ALLOWED and BATCH-ALLOWED must be mutually compatible.

RUN-PRIORITY

This parameter defines the execution priority (= task priority).

The parameter RUN-PRIORITY=*PARAMETERS (DEFAULT=n, MAXIMUM=n) an be used to define the maximum task priority a user may assign.

Example

An important production application (job handling) is to be run as a TP application with high priority and in a separate category.

With the aid of the JMU utility routine (see the “Utility Routines” manual [6]) a job class is defined (//DEFINE-JOB-CLASS), the access right is specified (//GRANT-JOB-CLASS-ACCESS) and storage in the file SJMSFILE is performed.

```
//DEFINE-JOB-CLASS NAME=PRIO1,
  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=...)
```

//SET-MODIFICATION-MODE is used to specify whether the JMU utility routine is to put the modification into effect immediately, in the current session, or/and write it in the SJMSFILE file, which is not evaluated until the next system startup.
The production application can be started simply by issuing /ENTER-JOB with JOB-CLASS=PRIO1.

Note

For reasons of compatibility, a logical OR link between the entries for the job class and the user ID is provided, the respective “higher” authorization having priority in each case.

Example

Job class	**User catalog**	**Status**
TP-ALLOWED=*NO	MAX-ALLOWED-CATEGORY=TP	TP permitted
RUN-PRIORITY= *PARAMETERS(DEFAULT=150, MAXIMUM=128)	MAXIMUM-RUN-PRIORITY=180	maximum priority=128

8.1.2 Job streams

BS2000/OSD is supplied with predefined job classes. If no job classes are defined in a BS2000 system, all jobs are assigned to class $SYSJC (to which users have no access) and managed by the **system job scheduler $SYSJS**, which is permanently coupled to the system.

The system job scheduler is a system task (see section “Creating and terminating tasks” on page 243) which executes as a privileged task in CPU state TPR. Its selection strategy is based on the LIFO (Last In First Out) principle and cannot be influenced.

Independent of the job classes defined and any selection strategies which may have been implemented, the system job scheduler permits jobs to be started under the system administrator ID TSOS at any time during the session.

Job classes defined with the parameter `JOB-TYPE=*BATCH` are assigned (`STREAM=name`) to a job stream (also known as a job scheduler). A job class can only belong to **one** job stream, but a job stream can manage more than one job class.

The jobs in the stream are sorted by the **standard job scheduler** according to a prescribed selection strategy. The standard job scheduler is an application program that executes in CPU state TU under the system administrator ID TSOS. The selection strategy is specified by means of the stream parameter (`STREAM -PARAMETER`). It is set by means of the JMU utility (`//DEFINE-JOB-STREAM`).

In order to implement various selection strategies, the scheduling parameters specified for the stream definition can be dynamically updated (using `//MODIFY-JOB-STREAM` of the JMU utility routine).

Up to 16 job streams can be installed at the same time.

Selection strategies (STREAM-PARAMETER='...')

- Selection according to arrival time:

  `JOB-PRIORITY=NO,CPU-TIME=NO,WAIT-TIME=NO`

  `WAIT-TIME` defines the waiting time of the job after acceptance.
  This strategy is advisable if the CPU time requirements for the different jobs are more or less the same.

- Selection according to priority:

  `JOB-PRIORITY=YES,CPU-TIME=NO,WAIT-TIME=NO`

  This ensures that important jobs will be given priority.

- Selection according to CPU time required:

  `JOB-PRIORITY=NO,CPU-TIME=YES,WAIT-TIME=NO`

  Jobs with lower CPU time requirements are given priority.

- Selection according to priority and CPU time requirements:

  `JOB-PRIORITY=YES,CPU-TIME=YES,WAIT-TIME=NO`

  When different jobs have the same job scheduling priority, those requiring less CPU time are given priority. When the CPU time required is the same, job scheduling priority is the criterion for selection.

- Selection according to priority and waiting time:

  `JOB-PRIORITY=YES,CPU-TIME=NO,WAIT-TIME=YES`

  By including the waiting time after job acceptance, jobs with a lower priority also have a chance of being selected.

- Selection according to throughput:

  `JOB-PRIORITY=NO,CPU-TIME=YES,WAIT-TIME=YES`

  Jobs with low CPU time requirements are given priority, taking into account the waiting time following acceptance.

- Selection according to throughput and priority:

  `JOB-PRIORITY=YES,CPU-TIME=YES,WAIT-TIME=YES`

  In addition to selection according to throughput, the job scheduling priority is taken into account.

The job scheduler passes the jobs it has selected and sorted to the **class scheduler**.

The class scheduler is not a separate task, but belongs to the core of the job management system. It ensures that the job mix desired by systems support (via the `CLASS-LIMIT` parameter of the individual job classes) will be adhered to.

When clearing bottleneck situations resulting from system saturation (saturation of the paging area), the `CLASS-WEIGHT` parameter determines the urgency the individual job classes will have.

8.2 Task management

In BS2000, logically interrelated requests to the system and their execution are managed in the form of tasks.

If these requests to the system are UTM applications, the various transaction codes within UTM can be controlled by means of so-called TAC classes. Similar resources are available here as are used for tasks and categories under PRIOR or PCS.
An introductory summary on this subject is provided on page 194.
Detailed information can be found in the openUTM manuals.

8.2.1 Introduction to the PRIOR concept

The name PRIOR is understood to encompass those task management routines permitting

- the control of tasks with the aid of categories and priorities
- the monitoring and control of the system workload by means of internal mechanisms.

In effect this implies

- management of the resources main memory (MM) and CPU, including execution of the corresponding control functions
  (see section “Managing the resources main memory and CPU” starting on page 244)
- controlling the tasks via wiat queues
  (see section “Controlling tasks via queues” starting on page 261)

The concept is based on priority control without changing the range of resources allocated. This means that the influence of systems support is restricted merely to determining which one of a number of users will first obtain a requested resource. Their influence on what share of a resource's capacity will be available to any applicant within a given time period is negligible.

8.2.1.1 Task categories

To improve the individual possibilities of control - depending on the respective user requirements - the tasks are subdivided into categories. There are 16 categories.

Four of these are always provided and are given the following default category names:

SYS	For system tasks only
TP	Transaction tasks
DIALOG	Interactive tasks
BATCH	Batch tasks

Further categories can be set up by systems administration under freely selectable names. Each task is assigned to a category.

In addition, there are four task attributes whose names are the same as the default category names SYS, TP, DIALOG and BATCH. The task attributes are assigned special execution parameters important to task scheduling.

The task attribute TP is different from the other task attributes in that it includes optimized main memory management tailored to fit the requirements of transaction mode (small number of functionally related user tasks, usually with relatively large working set requirements, serving a large number of terminals).
The task attribute TP can be provided by defining it in the job class or by calling the TINF macro (in the latter case the required authorization must have been entered in the user catalog).

The most important attribute of each category is the multiprogramming level (MPL). This refers to the number of tasks per category which are authorized to use main memory, i.e. are in the active state.

Systems support uses the category attributes MIN MPL, MAX MPL and WEIGHT (`/MODIFY-TASK-CATEGORIES`) to specify the relative importance of the categories with respect to activation (i.e. assignment of authorization to use main memory).

Whether or not a task actually becomes active or is preempted depends on the category attributes, as well as on the system workload and the priority assigned to the task in question (see section “Managing the resources main memory and CPU” on page 244).

Specifying **MIN MPL** guarantees a certain amount of minimum support for a category. The system sets out to reach the specified MIN-MPL value.

Specifying **MAX MPL** does not signify a fixed limit, i.e. activation will take place beyond the MPL value as long as there are no resource bottlenecks.

By means of the **WEIGHT** parameter, the activation or deactivation sequence is controlled. In addition, this parameter has a slight influence on which CPU is allocated.

Modifying the assignment of a task to a category

Systems support can use /MOVE-TASK-TO-CATEGORY to modify the assignment of a task to a category if, for example, different (better) support of this task or a reduction of the load on a category is required (with or without the use of PCS).

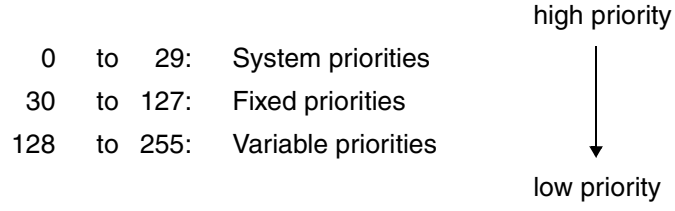
Effects of category control

Owing to the above-named attributes, category control has a negligible effect in the lower or normal workload range.

In the **full workload range** or **overload range** (i.e. in the event of resource bottlenecks) category control has a major influence and is therefore used for **load limitation**. Less significant categories are pushed into the background.
This full workload range is only reached in exceptional cases on systems with a very large main memory. Overloading of the CPU is not enough by itself to activate the load limitation facility. For this reason category control is largely ineffective where these systems are concerned.

8.2.1.2 Task priorities

By means of a priority assignment, the user specifies the urgency with which his/her requests are to be executed. There are 3 priority classes:



The variable and fixed priorities are available to the user. In this form they are referred to as external priorities.

Priorities can be assigned:

- by the user at command level with /SET-LOGON-PARAMETERS
  /ENTER-JOB
  /CHANGE-TASK-PRIORITY
- by the user at program level using the TINF macro
- by the system administrator or operator as a default value
  in the job class (RUN-PRIORITY) and with /CHANGE-TASK-PRIORITY

The maximum permissible priority must be entered in the job class or in the user catalog.

Variable priorities

Variable priorities are characterized by the dynamic setting of the internal priority using the HRN algorithm (Highest-Response-ratio Next), based on the ratio `Elapsed time / CPU time used` and taking into account a “start priority.” The start priority is computed on the basis of the specified external priority, the category weights, the performance power of the system, and the current workload.

This makes for expedient operation, even in the case of tasks with a low start priority.

The variable priority is calculated or upgraded at the following times:

- at activation time (see section “Main memory management” on page 246)
- during micro time slice runout
  (see section “CPU management control function” on page 260)
- periodically (every second).

The period check is performed for all active tasks as well as all inactive tasks ready to execute (i.e. those waiting in the “inactive ready” state for authorization to use main memory).

Fixed priorities

In the case of fixed priorities, the predefined external priority remains unchanged. It is converted without change to the internal priority.

Effects of priority

Since the priority is taken into account during both activation and initiation, issuing a priority which deviates from the standard external priority of 255, i.e. a better start priority, has practically the same effect no matter what the workload is, i.e. high, low or excessive.

When an I/O request is placed in the device queue, the priority is not taken into account. The effects of priority increase in direct proportion to the recourse made to the CPU by competing tasks.

Recommendations and guidelines

PRIOR is used to control the allocation of the resources main memory and CPU to user and system tasks in accordance with the preset category and priority parameters.

In this case priority assignments play an especially important part, since they can be used to give priority to individual tasks, whereas category control can only be used to give priority to entire categories. In the case of TP applications in particular, the priority of the central tasks should be raised. Extreme caution should be exercised when working with fixed priorities. They are not subjected to the HRN algorithm and might therefore monopolize the system. Under no circumstances should priorities better than 80 to 100 be issued.

Note

The parameters for the **SYS category** cannot be changed and are set to the following values:

```
MIN MPL  = 30
MAX MPL  = 64
WEIGHT   = 512
```

System tasks have the following priorities:

Permanent system tasks		22	
Exceptions:	TSN VMM:	16	
	TSN PGE:	1	
	TSN RMM:	128	at startup
		50	after SYSTEM READY
Dynamically generated tasks (e.g. DIAA, DSSM):		60	
Exceptions (e.g. BCAM):		30	

The user should bear in mind that resource shortages cannot be compensated for by the judicious selection of value settings. Although a user with a great deal of experience may be able to alleviate a CPU bottleneck or insufficient main memory, at least for especially important tasks, albeit to the detriment of others, this is scarcely possible when the bottlenecks occur in peripheral devices.

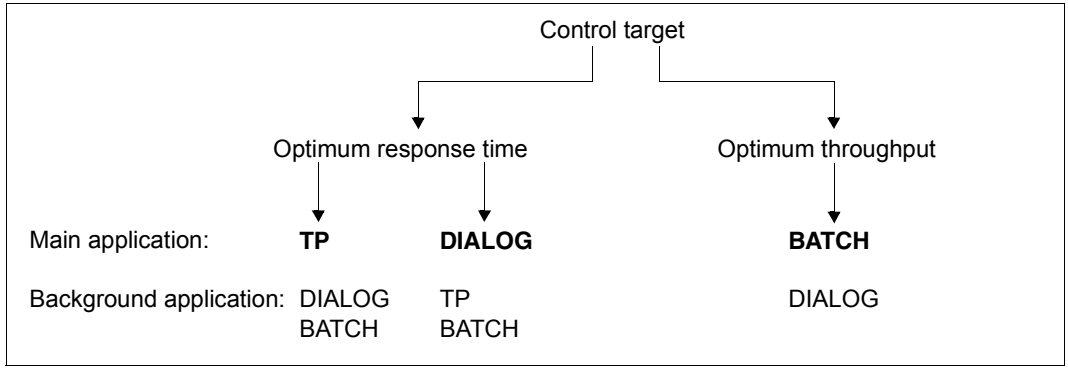
In the sections below, it is assumed that appropriate measures were taken during the capacity planning stage with regard to sizing the system to suit the problems involved. With the main objective being the optimum utilization of the resources installed.

8.2.1.3 Load analysis

In order to select the best possible values for PRIOR from the wide variety of possible settings, the user must have a clear idea of the workload. The following questions must be answered when analyzing the workload:

- Which applications demand which performance requirements?
- In the event of an overload, which applications can be pushed into the background to the advantage of more important applications?
- Which application belongs to which category?

First it is necessary to define the control target:



The distinction between optimum response time and optimum throughput is important because the two represent competing objectives: good response time behavior presupposes short waiting periods for the resources (especially channels, controllers and disk drives); thus the workload on these resources must not be excessive.

The further distinction within optimum response time as to whether the emphasis should be placed on TP or dialog applications is important for setting the category parameters.

- Dialog applications place a heavier burden on main memory with respect to paging intensity as compared to TP applications (see also section “Investigating system-oriented performance problems” on page 291).
- On the other hand, TP applications react more sensitively to background jobs, especially where paging is involved. This sensitivity is even more pronounced the fewer the number of tasks which are responsible for processing the application. The TP application is regarded as the main application if the number of TP transactions is greater than or equal to ≥ 50% of the total number of transactions.

The number of transactions per second can be calculated for TP and DIALOG applications either by using the guidelines given in section “Formulating a performance expectation” on page 20 or on the basis of measurements taken with the SM2 software monitor.

Example: Transaction rate

600 TP terminal users:
Average think time: 20 seconds (incl. network runtime)
Average response time: 1 seconds

50 dialog tasks:
Average think time: 20 seconds (incl. network runtime)
Average response time: 2 seconds

Transaction rate for the TP applications: `600 / (1 + 20) = 28.6 trans/s`

Transaction rate for the dialog applications: `50 / (2 + 20) = 2.3 trans/s`

In any event, the user should perform a measurement run using the SM2 software monitor (especially in the case of more complex applications). **Measurement of partial loads** is advisable.

The most practical approach is to start with the most important part of the workload, setting the PRIOR parameters for the main application according to the recommended values (see also “Setting the category parameters” on page 211 and “Assigning priorities” on page 219) and calculating the response time behavior, the transaction rate and the load on resources. These values are required in order to optimize the main application when operating with mixed loads, so that the user can determine how severely ancillary applications detract from the main application.

By measuring the main application in this manner, the assumptions made at the capacity planning stage can be verified or the actual resource requirements demonstrated. If an unsatisfactory level of performance should arise while the main application is operating by itself, the user should carry out a bottleneck analysis (see section “Performing a bottleneck analysis” on page 282).

The next step is to set the PRIOR parameters for the background application in accordance with the recommendations (see “Setting the category parameters” on page 211 and “Assigning priorities” on page 219) and then to increase the workload step by step.

Example: Optimizing the response time

Giving the TP application precedence over DIALOG and BATCH applications

– Step 1:

  Measure the pure TP application (modified PRIOR parameters)

  Measurement period depending on workload: 15 minutes when the load hardly fluctuates

  60 minutes when the load fluctuates greatly

  Determine response times, transaction rate, resource utilization

– Step 2:

  Add the DIALOG application (modified PRIOR parameters)
  Measurement period: 60 minutes

– Step 3:

  Add the BATCH application (modified PRIOR parameters)
  Measurement period: Several measurements, each 60 minutes

  Observe runtime behavior

As the result of the ancillary application, the main application will inevitably be slowed down somewhat (approx. 10 - 20%). This is unavoidable. By judiciously selecting and limiting the ancillary applications, the user can ensure that this delay remains within acceptable limits. The ancillary applications should, wherever possible, only make use of those resources which are scarcely required for the main application. Furthermore high sensitivity of the ancillary applications with respect to MPLs and priorities is an added advantage.

User privileges

Following the workload analysis, it is necessary to finalize user privileges via `/MODIFY-USER-ATTRIBUTES` or the `JOB-CLASS` parameters (i.e. define the highest priority a user may specify).

It is also necessary to check whether TP authorization has been entered for DCAM, UTM, UDS and SESAM applications so that it is possible to take full advantage of the TP task attribute.

8.2.1.4 Setting the category parameters

The primary goal for systems with limited memory space is to apportion **available** pageable main memory efficiently, according to the specific weight of the categories (NPP = number of pageable pages).

Memory capacity is limited if there are INACTIVE READY TASKS despite the fact that reasonable MPL values have been set (report “Active and inactive tasks for category” in the report group CATEGORY-QUEUE of SM2).
If this condition has never occurred during operation, the reader can skip this section and concentrate on the “Assigning priorities” on page 219.

For PRIOR to be able to react flexibly to slight variations in workload, the working set requirements of the tasks, as represented by the sum of the MIN MPL values for all categories should be: (between `0.3 and 0.5) * NPP`. The choice of the factor (0.3 or 0.5) depends on the emphasis desired for the application involved (see below).

Guidelines for the average working set requirements (4KB pages):

TP tasks:	WS_{TP}	=	200	to	400	pages for UTM and DCAM tasks
		=	500	to	1,000	pages for DB tasks
DIALOG tasks:	WS_{DIAL}	=	50	to	150	pages
BATCH tasks:	WS_{BATCH}	=	50	to	200	pages

The working set for system routines and common memory pool, which are called by application programs, are managed globally in the SYS category (placeholder is TSN=PGE).

Category SYS:	WS_{SYS}	=	2,000	to	4,000	pages

See the ACTIVITY report of SM2 for the NPP value.

In the following, it is assumed for the sake of simplicity that, from the user's viewpoint, only the categories with the standard names TP, DIALOG and BATCH exist.

Main application: TP

background application: dialog, batch

TP category

This category contains the tasks which place the greatest demands on performance. It is not desirable to limit the workload of this category, even in the event of overload. Therefore the value of MAX MPL must be equal to the total number of TP tasks in the system (including management tasks such as TDADM, Data Base Administrator, etc.).

The value for MIN MPL should correspond to the desired number of active TP tasks. It is generally equal to or approximately equal to the MAX MPL value. If the value is too low and an overload occurs, this will cause TP tasks to be deactivated automatically (see section “Main memory management” on page 246).

Recommended values:

$MIN\ MPL_{TP} = 0,8 * MAX\ MPL_{TP}$
$MAX\ MPL_{TP} = total\ number\ of\ TP\ tasks$
WEIGHT = 500

DIALOG category

This category is usually characterized by wide variations in workload. PRIOR attempts to reach the MIN MPL value set by systems support, if necessary by suppressing tasks in other categories (see section “Main memory management” on page 246).

Since in this case the DIALOG application is meant to run in the background, the value of MIN MPL must not be too high.

Due to the effect of think time, not all dialog tasks are active at the same time. The number of active dialog tasks can be estimated with the aid of the following formula:

```
Number of active dialog tasks =  total number of dialog tasks / N
N = (Average think time + Response time) / Average response time
```

Guidelines:

- Average think time (incl. network runtime) = 16 seconds
- average response time = 4 seconds

```
N = 20 s / 4 s = 5
```

In order to calculate the MIN MPL value it is essential to know how much pageable main memory (according to the formula `0.5 * NPP` with the emphasis on TP mode) is available for **active** dialog tasks after subtracting the working set requirement of the main application, the SYS category or, if applicable, of any necessary batch tasks for active dialog tasks (see example).

The value of MAX MPL is conceived as a workload limit for temporary overload situations. It should be set to twice the value of MIN MPL.

Recommended values:

$$\text{MIN MPL}_{DIAL} = \frac{0{,}5 \bullet NPP - (\text{MIN MPL}_{TP} \bullet WS_{TP} + \text{MIN MPL}_{BATCH} \bullet WS_{BATCH} + WS_{SYS})}{WS_{DIAL}}$$

WS_{TP}: average working set requirements per TP task

WS_{DIAL}: average working set requirements per dialog task

WS_{BATCH}: average working set requirements per batch task

WS_{SYS}: average working set requirements for system routines and common memory pools

Note

At this point it is only necessary to take the BATCH portion into account if, for operational reasons, a certain minimum number of parallel batch tasks is absolutely necessary for the corresponding IT installation.

```
MAX MPL = 2 * MIN MPL
WEIGHT = 100 (50 < WEIGHT < 200)
```

BATCH category

Normally, batch tasks are run in the background in order to make use of the free resources. In this case, the value of MIN MPL is derived from the size of pageable main memory (more precisely `0.5 * NPP`) minus the working set requirement for the main application TP, the DIALOG application and the SYS category.

The value of MAX MPL varies according to the configuration workload, and should not deviate markedly from the value of MIN MPL.

Recommended values:

$$\text{MIN MPL}_{BATCH} = \frac{0{,}5 \bullet NPP - (\text{MIN MPL}_{TP} \bullet WS_{TP} + \text{MIN MPL}_{DIAL} \bullet WS_{DIAL} + WS_{SYS})}{WS_{BATCH}}$$

```
MAX MPL = MIN MPL + 1
WEIGHT = 10 (1 < WEIGHT < 50)
```

Example for main application TP

MM = 64 MB

main application TP: 6 UTM tasks, 2 DB tasks total: 10 TP tasks

background application: 10 dialog tasks, 2 batch tasks (minimum)

```
TP category:
      MIN MPL_TP       =  8
      MAX MPL_TP       =  10
      WEIGHT           =  500
DIALOG category:
      Number of active dialog tasks =  total number of dialog tasks / N = 2
      MIN MPL_DIAL     =  2
      MAX MPL_DIAL     =  4
      WEIGHT           =  100
BATCH category:
      MIN MPL_BATCH    =  2
      MAX MPL_BATCH    =  3
      WEIGHT           =  10


64 MB = 16,000 pages, resident MM requirement = 1500 pages,
i.e. NPP = 14,500 pages.
```

$$\text{MIN MPL}_{TP} \bullet \text{WS}_{TP} + \text{MIN MPL}_{DIAL} \bullet \text{WS}_{DIAL} + \text{MIN MPL}_{BATCH} \bullet \text{WS}_{BATCH} + \text{WS}_{SYS} \leq 0{,}5 \bullet \text{NPP}$$

$$(6 \bullet 300 + 2 \bullet 800) + 2 \bullet 150 + 2 \bullet 100 + 2000 = 1800 + 1600 + 300 + 200 + 2000 =$$

$$6100 \leq 0{,}5 \bullet 14500 = 7250$$

Main application: Dialog

background application: TP, batch

TP category

The fact that the TP application appears as a background application does not necessarily imply that minimal or even no demands are placed on performance; it only means that the number of TP transactions is less than 50% of the total number of transactions.
This is always the case whenever new TP applications are introduced and the number of terminal users employing the new procedure is still relatively small. Since a **single** TP task always serves several terminal users, and also normally has greater working set requirements as compared to dialog tasks, it is generally not desirable to have TP tasks automatically deactivated in the event of overload. Therefore, the value of MAX MPL should likewise be equal to the total number of TP tasks in the system.
Similarly, the value of MIN MPL should be equal or approximately equal to the value of MAX MPL.

Recommended values:

$\texttt{MIN MPL}_{\texttt{TP2}}$ = 0,8 * $\texttt{MAX MPL}_{\texttt{TP}}$
$\texttt{MAX MPL}_{\texttt{TP2}}$ = total number of TP tasks
WEIGHT = 500

DIALOG category

If the emphasis of the application lies on the DIALOG side, this entails higher demands on main memory with respect to paging intensity (see section “Investigating system-oriented performance problems” on page 291) as compared to TP applications.

Therefore, the working set requirements of the tasks as represented by the sum of the MIN MPL values of the TP, DIALOG and BATCH categories should be ≤ 0.3 * NPP.

For calculating the MIN MPL value, it is essential to know how much pageable main memory (according to the formula 0.3 * NPP with the emphasis on DIALOG mode) is available for active dialog tasks after subtracting the working set requirements of the TP tasks, the SYS category or necessary batch tasks for active dialog tasks (see example).

Here, too, the number of active dialog tasks (initial estimation) should be set to

Number of active dialog tasks = total number of dialog tasks / 5

Recommended values:

$$\text{MIN MPL}_{\text{DIAL}} = \frac{0{,}3 \bullet \text{NPP} - (\text{MIN MPL}_{\text{TP}} \bullet \text{WS}_{\text{TP}} + \text{MIN MPL}_{\text{BATCH}} \bullet \text{WS}_{\text{BATCH}} + \text{WS}_{\text{SYS}})}{\text{WS}_{\text{DIAL}}}$$

Note

At this point, it is only necessary to take into account the BATCH portion if, for operational reasons, a certain minimum number of parallel batch tasks is required.

```
MAX MPL = 2 * MIN MPL
WEIGHT = (300 < WEIGHT < 500)
```

BATCH category

Given that batch tasks are running solely as a background workload, the following values apply:

$$\text{MIN MPL}_{\text{BATCH}} = \frac{0{,}3 \bullet \text{NPP} - (\text{MIN MPL}_{\text{TP}} \bullet \text{WS}_{\text{TP}} + \text{MIN MPL}_{\text{DIAL}} \bullet \text{WS}_{\text{DIAL}} + \text{WS}_{\text{SYS}})}{\text{WS}_{\text{BATCH}}}$$

```
MAX MPL = MIN MPL + 1
WEIGHT = 10 (1 < WEIGHT < 50)
```

Example for main application Dialog

MM = 64 MB

Main application Dialog: 45 dialog tasks

batch: 2 batch tasks (minimum)

background application: 2 UTM tasks, 1 DB task; total: 4 TP tasks

TP category:

MIN MPL_{TP} = 3
MAX MPL_{TP} = 4
WEIGHT = 500

DIALOG category:

Number of active dialog tasks = total number of dialog tasks / N = 9

MIN MPL_{DIAL} = 9
MAX MPL_{DIAL} = 18
WEIGHT = 400

BATCH category:

MIN MPL_{BATCH} = 2
MAX MPL_{BATCH} = 3
WEIGHT = 10

64 MB = 16,000 pages, resident MM requirement = 1500 pages,
i.e. NPP = 14,500 pages.

$$\text{MIN MPL}_{TP} \bullet WS_{TP} + \text{MIN MPL}_{DIAL} \bullet WS_{DIAL} + \text{MIN MPL}_{BATCH} \bullet WS_{BATCH} + WS_{SYS} \leq 0{,}3 \bullet NPP$$

$$2 \bullet 200 + 1 \bullet 500) + 9 \bullet 100 + 2 \bullet 100 + 2000 = 400 + 500 + 900 + 200 + 2000 =$$

$$4000 \leq 0{,}3 \bullet 14500 = 7250$$

Main application: Batch

Background application: Dialog

This load is typical for night-shift operation of systems whose main memory is dimensioned for daytime online operation.
As a rule, batch applications place a considerably lighter burden on main memory than online applications with regard to paging intensity; however, they put a far greater burden on the CPU.
The number of batch tasks that can be handled thus depends less on the main memory size than on the speed of the CPU.
The settings for the category parameters is of no significance here (the required load limitation can be specified using the job classes parameter `CLASS-LIMIT`).

8.2.1.5 Assigning priorities

By assigning a priority higher than the default priority of 255, it is possible to give preference to individual tasks with regard to activation and, especially, to the utilization of the resource CPU.

As a rule, assigning a variable priority of medium size is sufficient.

Variable priority

Priority should be varied in accordance with the workload in units of 5 or 10 until the desired performance target is attained. Larger priority increments are required when

- many tasks are active
- the active tasks are predominantly I/O-intensive.

Depending on the control target, the following values are recommended for the initial setting:

Main application: TP

background application: dialog, batch

TP category

If a DB/DC application with several tasks is involved, database systems which operate in multithread mode (SESAM/SQL, UDS) should be assigned a lower priority than the associated DC tasks (UTM, DCAM, ...). This allows a considerable reduction in the necessary communication between the DB and DC tasks for each database call.
If there are two or more TP applications of different weight, the priorities may be assigned incrementally.

Recommended priority range: 130 - 180

DIALOG category

The dialog application is to run in the background. As a rule, no preference is given to individual dialog tasks in this category.

Recommended priority: 210

BATCH category

For pure background batch tasks, the default priority is sufficient. To emphasize tasks within the category, it is sufficient to raise the priority to 240 (maximum).

Recommended priority range: 240 - 255

Main application: Dialog

background application: TP, batch

DIALOG category

In general, dialog tasks are treated equally as far as priority is concerned.

Recommended priority: 180

TP category

Unlike the setting of category parameters, where an attempt is made to avoid deactivation of TP tasks even when an overload occurs, the background application TP should receive lower priority with regard to the use of the resource CPU.

If, in the case of dialog tasks, loads similar to the batch category are found (compilations, programming sequences within a response time) processing should continue as for the main application TP.

Recommended priority: 210

BATCH category

When operating solely in the background, the default priority is sufficient.

Recommended priority range: 230 - 255

Under normal circumstances, batch tasks are run in the background, with the purpose of taking advantage of free resources. All tasks which are background batch tasks should be executed with the lowest priority (255), whereas all other tasks should have a priority of 210 or better. Regardless of whether the TP or DIALOG category is the main application, batch tasks which execute in the background must consist of compute-intensive programs, so that the system can regulate the use of available resources.
These tasks are then made to handle their I/O operations via channels and disk drives which are not being used for foreground tasks.

Fixed priorities

Fixed priorities are provided for special applications with extremely high real-time requirements. The following points should be borne in mind in this connection:

Fixed priorities severely restrict the system's freedom to make decisions. Tasks awaiting activation or initiation cause other tasks to be immediately suppressed when resources are being used to full capacity.

Therefore, to ensure improved system behavior, fixed priorities should be chosen only after a thorough analysis of the workload and the allocation of resources has been made and in conjunction with steps to limit the workload (MAX MPL).

Tasks given a fixed priority should not be compute-intensive, neither in part nor as a whole, since they are subject to stringent demands with regard to freedom from errors.

8.2.1.6 I/O priority handling for tasks using IORM

Normally all tasks have the same priority when disk I/Os are executed.

An I/O-intensive application with low priority can hinder another, higher-priority application if these application execute I/Os on the same (logical) device. Operations can also be impaired if the I/Os are executed on different (logical) devices which are located on the same physical device or are connected via the same paths, are accessible via the same ports or connected to the same channels.

The IOPT (I/O Priority Handling for Tasks) function enables IORM to detect such conflict situations and to intervene in I/O operation for control purposes. Here IOPT examines both the utilization level of the I/O units (devices, paths, ports and channels) and the I/O priorities of the tasks which use them.

IOPT distinguishes three I/O priorities for tasks:

- HIGH (high I/O priority)
- MEDIUM (medium I/O priority)
- LOW (low I/O priority)

It is possible to assign the task categories (with the exception of `SYS`) corresponding I/O priorities (`IO-PRIORITY` attribute) using the following command:
`/MODIFY-TASK-CATEGORY IO-PRIORITY=*NONE/*HIGH/*MEDIUM/*LOW`

Further information on IORM and IOPT can be found in the “Utility Routines” manual [6].

8.2.2 Introduction to PCS concept

PCS (Performance Control System) is available as a performance control subsystem for regulating loads; it is based on PRIOR and provides the following functions:

- Improved preferential treatment of load components thanks to targeted SERVICE allocation; the SERVICE of a server is determined from the time the CPU and main memory (MEMORY) are reserved and from the number of I/O operations (IO).
- Increased transparency with regard to the distribution of system capacity over the various categories
- Dynamic adjustment of the service allocation to categories or tasks in accordance with fluctuations in load
- Throughput control by monitoring the capacity consumption in the categories
- Differentiation between “short-running jobs” and “long-running jobs” and thus improved control possibilities via automatic category changeover
- Improvement of the response time by means of the preferential treatment of short-running jobs.
- Differentiation in the control of a number of TP applications through allocation to several categories

PCS operation can be controlled by means of global system parameters and/or category-specific parameters:

Category-specific control parameters

With the aid of the SERVICE-QUOTA (S-Q) parameters, the performance capacity of the system is distributed among the categories. The parameters SERVICE-QUOTA-MIN and SERVICE-QUOTA-MAX define a framework within which PCS plans service requirements for the specified category (S-Q-PLN). If a category does not make full use of its share of performance, the remainder is then available to other categories.

By means of the dilation factor (REQUEST-DELAY parameter), PCS recognizes how extensively the tasks of this category are delayed by competing tasks. Dilation thus serves to indicate workload levels and specifies the factor by which the runtime of a job will be delayed if the job is forced to wait for the allocation of a resource owing to multiprogramming operation.

```
Dilation  = Runtime in multiprogramming operation / Single runtime
```

In the case of the simultaneous use of SM2 and the accompanying SM2 monitoring program SYSTEM for all disks, PCS periodically measures the current I/O time per category, which increases the accuracy of the dilation calculation accordingly.

Within a given dilation range (REQUEST-DELAY-MIN, REQUEST-DELAY-MAX), PCS dynamically adjusts the S-Q-PLN value of the category in accordance with the current load (REQUEST-DELAY-ACTUAL).

SERVICE is distributed over the various individual categories according to the following hierarchy:

1. Categories with a dilation range and max. share of performance of 100%

   These categories are allocated the performance capacity to which the current dilation entitles them before all other categories (“planned values”).

   Insufficient capacity:
   If several such categories exist and if the sum of their planned values exceeds 100%, the planned values are reduced proportionally so that they total 100%.

   Residual capacity:
   If the sum of the planned values is less than 100%, the residual capacity is planned for use by categories with and without a dilation range.
   If the tasks of the categories with a dilation range and max. share of performance of 100% are not in a position to use the planned capacity, the (current) residual capacity is made available to the other categories with and without dilation range. (Exception: full global response-time orientation).

2. Categories with a dilation range and max. share of performance < 100%

   These categories are allocated the performance capacity which is left over by categories with a dilation range and max. share of performance of 100% and to which their current dilation entitles them, before categories without a dilation range.

   Insufficient capacity:
   If several categories with a dilation range exist and if the sum of their planned values exceeds the capacity still available, the planned values are reduced proportionally so that their sum is equal to the available capacity.

   Residual capacity:
   If the sum of the planned values is less than the available capacity, the residual capacity is planned for use by categories without dilation range.
   If the tasks of the categories with a dilation range are not in a position to use the planned capacity, the (current) residual capacity is made available to the categories without dilation range. (Exception: full global response-time orientation).

3. Categories without dilation range

   Categories without dilation range are allocated the performance capacity which is left over by the categories with a dilation range. This capacity is distributed proportionally among the categories without dilation range according to their defined maximum shares of performance.

In order to take full advantage of the regulation functions of PCS, at least one category without a dilation range should always be provided.

The function “automatic category changeover” serves to improve differentiation between load requests with high resource requirements (long-running jobs) and those with low resource requirements (short-running jobs). In this case the user employs the DURATION variable to specify how many SERVICE UNITs may be used per processing step before an automatic changeover to the following category (NEXT CATEGORY) with a “weaker” performance capacity is to take place.
SERVICE UNITs are the weighted sum of the work carried out by the resources CPU, IO and MEMORY.

By means of the THROUGHPUT-QUOTA parameter, a percentage indicating the ratio between response-time and throughput optimization within the category is defined. The effects of this are as follows:

	Response time optimization		**Throughput optimization**
THROUGHPUT-QUOTA	full		full
	0%	50%	100%

For systems with TP or interactive mode, response-time optimization is of more importance; THROUGHPUT-QUOTA=0 is therefore recommended in the corresponding TP categories as well as in the DIALOG start category (short-running jobs).

For systems operating predominantly in batch mode, the most important performance goal is the greatest possible throughput together with high utilization of the resource capacity. For throughput-oriented categories, THROUGHPUT-QUOTA values greater than 50% should be provided.

Global system control parameters

Global dilation throughout the system (REQUEST-DELAY-MAX) is a threshold value for the dilation of all tasks belonging to a category with a dilation range. When this value is exceeded, load limitation goes into effect.

THROUGHPUT-QUOTA specifies a percentage value that determines the relation between response-time and throughput optimization for the entire system. The parameter has a very pronounced effect on the limit values:

- THROUGHPUT-QUOTA=0
  results in hard response-time optimization. This means that background categories are totally suppressed in the event of an overload (exceeding the value of the global system parameter REQUEST-DELAY-MAX). As a consequence, utilization of the resources and thus the throughput may be impaired.
- THROUGHPUT-QUOTA=100
  is only used for pure batch mode.

Procedure when using PCS

As a rule, the hardware configuration (is the configuration adequate to deal with load requirements?) and the software configuration (location and size of the system files TSOSCAT, paging areas and SYSEAM) should be checked before the system is used for the first time.

Depending on the particular load situation, first the predefined default options

- STD#TP (predominantly TP mode),
- STD#DIA (predominantly interactive mode) and
- STD#BATCH (predominantly batch mode)

should be used and the performance results measured (for more details see the “PCS” manual [23]). The results should be at least as good as those achieved with PRIOR, but are generally better. If the check shows that performance goals have not been achieved in individual categories, the service planning values for these categories must be increased. The current dilation value of REQUEST-DELAY-ACT constitutes the main criterion here (information can be obtained with `/SHOW-PCS-OPTION CATEGORY-NAME=*ALL` or from the report “Request delay for category” in the report group PCS of SM2).

In transaction mode and interactive mode the user is advised to proceed step by step as described below in order to increase the service values in the categories in question:

1. If the current dilation is between the values for REQUEST-DELAY-MIN and REQUEST-DELAY-MAX, first of all reduce REQUEST-DELAY-MAX for the appropriate categories. As a result, the value for SERVICE-QUOTA-PLAN will increase. The optimum work point is reached when REQUEST-DELAY-ACT is slightly below the value of REQUEST-DELAY-MAX.
2. Increase the SERVICE-QUOTA-MAX value for the appropriate category. The ratio of S-Q-MAX / S-Q-MIN £ 3 is recommended
3. Increase the value of SERVICE-QUOTA-MIN. This parameter serves the special function of reserving capacity to cater for sudden load fluctuations.

If these actions do not produce the desired results, i.e. shorter response times, further tuning measures are required; these vary depending on which mode of operation has priority.

When interactive mode has priority, long-running transactions can be suppressed by taking the following actions:

a) Reduce S-Q-MAX for the next category (Dialog 1)

a) Reduce S-Q-MIN for the next category (Dialog 1)

When TP mode has priority, first the service planning value for the DIALOG background load can likewise be reduced by reducing the value of S-Q-MAX and S-Q-MIN for the DIALOG category.
If there are a number of mutually dependent tasks in the TP category (e.g. UTM/UDS, UTM/SESAM), it is also possible to increase the priority for those tasks with greater service requirements.

If there are several applications in the TP category, the TP response times can generally be improved by increasing the service planning value. When the aim is targeted improvement of individual applications, it is advisable to keep each application in a separate TP category.

If you wish to ensure that a category is allocated whatever capacity it needs (if necessary, the entire capacity of the system), enter the value SERVICE-QUOTA-MAX=100 for that category. It does not make sense to give this value to two or more categories.

All modifications to the PCS default settings described here must be performed by systems administration with the aid of the PCSDEFINE utility routine. Once PCS has been started as a subsystem, systems support can obtain information about PCS using system commands and dynamically update the control parameters (for more details see the “PCS” manual [23]).

Enhancements to PCS V2.7 and higher:

- The BS2000 command MOVE-TASK-TO-CATEGORY was introduced in particular to provide individual tasks with a higher service allocation in exceptional cases. This enables tasks to be placed in a different category (defined in JMU).
- The MODIFY-TASK-CATEGORY command, IO-PRIORITY operand, enables all tasks in a category to be connected to the priority control of overloaded volumes by means of IORM.

Examples of the effect of the PCS parameters

Example 1

Effect of the default option STD#DIA in predominantly interactive mode for developing programs with a batch load in the background.

Default option parameter STD#DIA:

Category	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	-	-	-

All incoming interactive transactions are first accepted by the default category DIALOG. As a rule, short interactive jobs such as EDT statements require less than 200 SERVICE UNITs and thus are retained in the DIALOG category until the transaction has ended.

If the service requirements exceed 500 SERVICE UNITs (e.g. in the case of file operations), an automatic changeover to the “weaker performance” category DIALOG1 is effective.

Compiling and program testing tasks with a service requirement of more than 2000 SERVICE UNITs are switched to category DIALOG2 (no REQUEST-DELAY entry, i.e. no request for response time), where they execute until completed.

As compared to the throughput-oriented category BATCH (THROUGHPUT-QUOTA=70) for the background batch load, long-running interactive jobs in category DIALOG2 are given priority as the result of the larger SERVICE-QUOTA-MAX value.

In exceptional cases the BATCHF (batch fast) category must be used for particularly important batch runs.

Using PCS results in considerably improved response times for short-running interactive jobs when compared to PRIOR, even if PRIOR is running efficiently. The response times for interactive transactions with high service requirements are longer.

Example 2

Effect of the default option STD#TP in predominantly TP mode with an interactive and batch load in the background.

Default option parameter STD#TP:

Category	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	-	-	-

Categories with a dilation entry (REQUEST-DELAY) are given priority over categories without dilation.

If the total of the SERVICE-QUOTA-MAX values of the categories with dilation exceeds 100%, appropriate standardization measures are implemented, i.e. the competition of the categories relative to one another remains the same. By explicitly specifying SERVICE-QUOTA-MAX=100 for TP it is ensured that the background load will be fully suppressed if so required.

All incoming interactive transactions are first accepted by the DIALOG category. If the service requirements exceed 500 SERVICE UNITs, changeover to the category DIALOG1, in which no response time requirements are made, is automatic.

Prioritizing the more resource-intensive interactive transactions in DIALOG1 at the expense of the real batch tasks in the BATCH category is accomplished by means of the higher SERVICE-QUOTA-MAX value (=30).

In exceptional cases the BATCHF (batch fast) category must be used for particularly important batch runs.

When TP mode is given priority, PCS brings about a dramatic reduction of response times for TP applications in comparison to PRIOR. The short-running jobs of the background load DIALOG also benefit, while the long-running jobs are heavily suppressed. Satisfactory TP response times during heavy workload situations cannot be achieved with variable priorities under PRIOR. Under PCS control, all improvements in performance can be easily achieved using **one single control parameter**.

Example 3

Controlling more than one TP application

To control TP applications, each application, or a number of equal-ranking applications, are placed in a specific category. The next option is used to control 3 TP applications and an interactive load and batch load for the remaining capacity. The 3 TP applications are assigned to the categories TP, TP2 and TP3 according to their priority.

Categories TP2 and TP3 must be defined as separate categories using the task attribute TP in the DEFINE-JOB-CLASS statement (see the “Utility Routines” manual [6]).

PCS control parameters:

Category	SERVICE-QUOTA		REQUEST-DELAY		DURATION	NEXT	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	-	-	-

Categories with a dilation entry (REQUEST-DELAY) are given priority over categories without dilation. All TP transactions are given priority over interactive transactions owing to the more favorable values for SERVICE-QUOTA and REQUEST-DELAY.

Up to 80% of the systems performance capacity can be assigned for the TP categories (SERVICE-QUOTA-MAX for TP, TP2 and TP3).

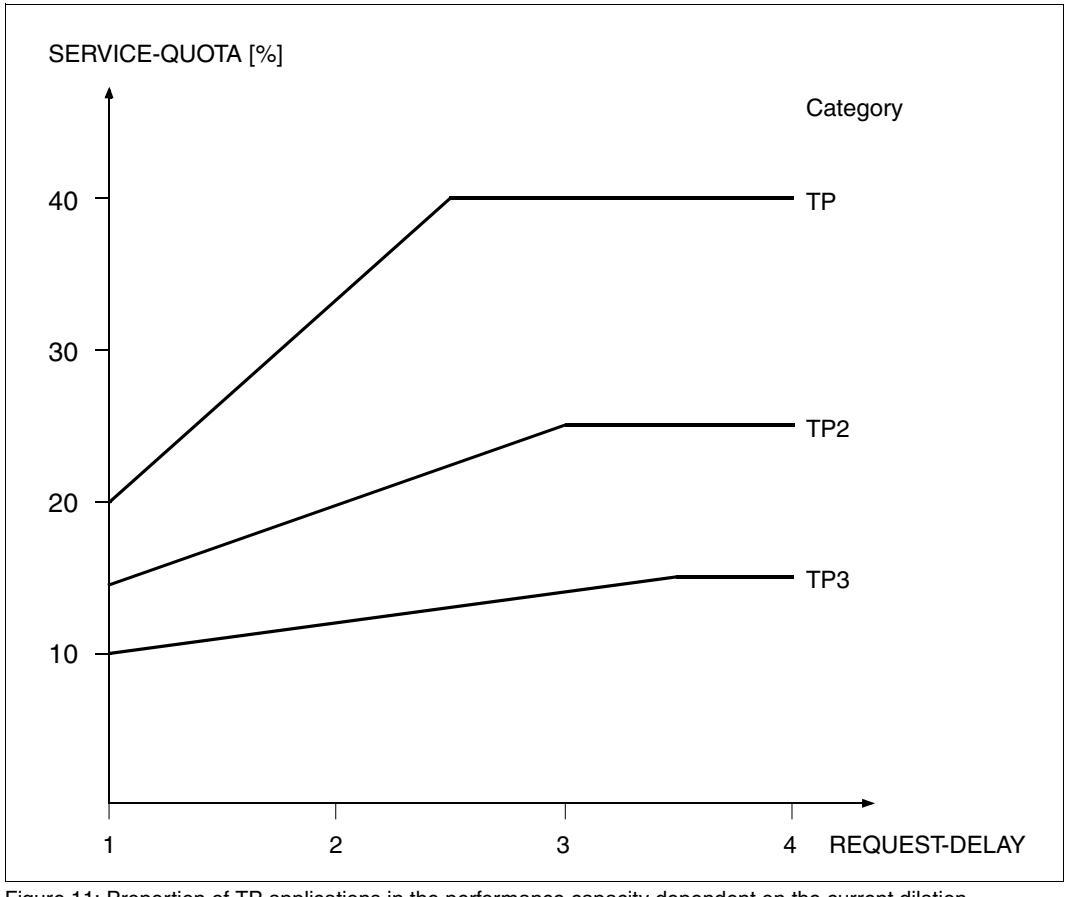


Figure 11: Proportion of TP applications in the performance capacity dependent on the current dilation

The figure 11 shows the proportion of TP applications dependent on the current dilation. When the workload fluctuates (which is expressed by the dilation) a fluctuating proportion of the performance capacity must be reckoned on.

For applications in the TP category, the value SERVICE-QUOTA MAX (40%) is planned for a dilation of 2.5 (REQUEST-DELAY MAX). If the current dilation value is between 1 and 2.5, a service planning value between 20% and 40% is assigned.

Applications in categories TP2 and TP3 are assigned a correspondingly smaller proportion of the performance capacity since the values for SERVICE-QUOTA and REQUEST-DELAY are lower.

The incoming interactive transactions are stated in the DIALOG category. If the service planning requirements exceed 300 service units, the interactive jobs are placed in the lower-performance category DIALOG1. Specifying the value 10 under THROUGHPUT-QUOTA for the DIALOG category reduces the activation priority for interactive transactions.

The more complex interactive transactions in the next category DIALOG1 are given priority over genuine batch jobs in the BATCH category by means of a higher SERVICE-QUOTA-MAX value.

In exceptional cases the BATCHF (batch fast) category must be used for particularly important batch runs.

In the case of larger systems with a very wide variety of workloads, rapidly changing performance requirements and a large number of competing tasks, desired performance goals can only be achieved by using PCS. The more heterogeneous the workload, the greater the number of tasks; the more varied the resource requirements of the individual load components, the more effective PCS will be.

8.2.3 Introduction to the TANGRAM concept

The objective of the TANGRAM (task and group affinity management) concept is to make better use of the performance available from multiprocessor hardware.

All the servers use fast caches to minimize the differential between CPU speeds and access times to main memory. Depending on the instruction profile (in particular the number of accesses to main memory required), the possible performance is to a large degree determined by the hit rate in the cache.

There are two types of cache, which are generally combined:

- Store-through caches (STC)
  The data is “passed through” as soon as possible, i.e. the data in main memory is updated as soon as possible after a write request.
- Store-in caches (SIC)
  The data is retained in the cache and is written back to main memory on the basis of the LRU (least recently used) principle. This means that the data in main memory is not up to date.

Whereas in the case of monoprocessors, full performance can be achieved using a sufficiently dimensioned cache, the situation with multiprocessors is far more complex.

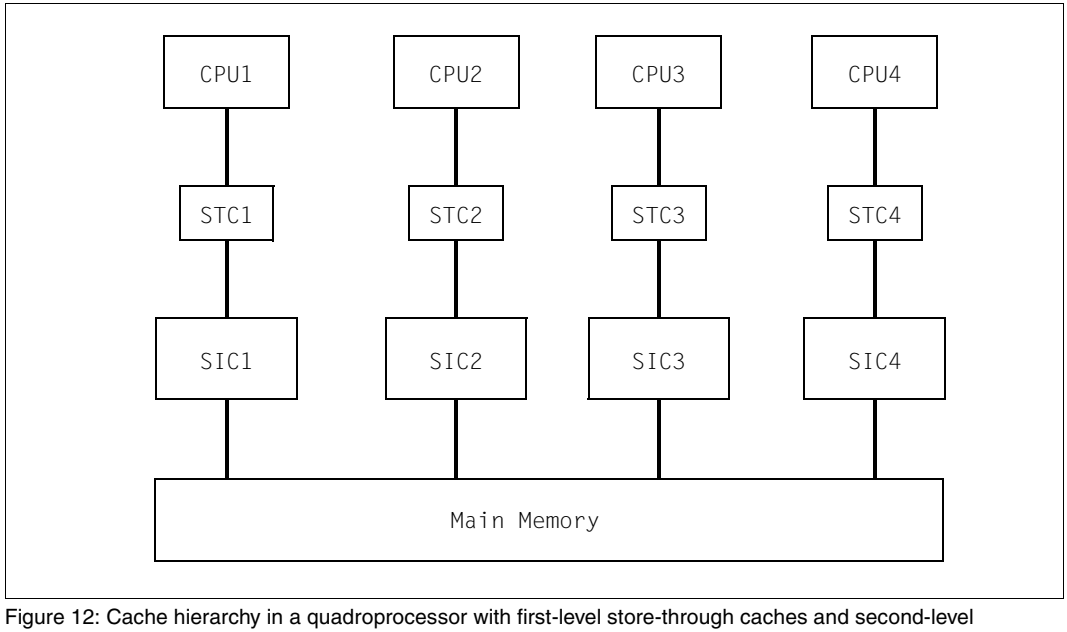


Figure 12: Cache hierarchy in a quadroprocessor with first-level store-through caches and second-level store-in-caches

The greater the number of tasks running on different CPUs which access the same memory areas (e.g. memory pools), the greater the probability that the current information is not located in the CPU's own cache (SIC), but in a neighboring cache.

The current data must first be moved to the CPU's own cache before it can continue processing, with the result that the CPU resources available to the user are reduced. This situation is exacerbated as the proportion of write accesses to the same main memory pages rises.

The TANGRAM concept attempts to counteract this loss of performance by forming task groups and assigning them to certain CPUs (preferably a subset of the CPUs available).

Formation of task groups

A task group is made up of so-called “affined tasks” which have write access to a large set of common data.

The TINF macro is used to sign onto a task group. The TANGBAS subsystem, which is started automatically when the system is ready, is responsible for managing the task groups.

The software products UTM, UDS and SESAM/SQL automatically issue this TINF call when each application or database system is started. The tasks of a single UTM application thus form a single task group.

Assignment to CPUs

Task groups are assigned to CPU groups dynamically, depending on the load, if viewed over longer periods.

For shorter periods (PERIOD parameter with a default value of 10 seconds), on the other hand, fixed assignments apply, in order to alleviate the problems with the caches described above.

The **assignment algorithm** in the TANGRAM subsystem, which must be started explicitly, measures the following values at periodic intervals:

- CPU usage of each task group
- workload of each CPU
- total CPU workload

If necessary, the task groups are reassigned to the CPU groups according to the following criteria:

- The task group must have a minimum CPU requirement (THRESHOLD parameter, with a default of 10%), before it is taken into account for individual assignment to a CPU.
- If a task group loads one or more CPUs to a great extent (CLEARANCE parameter, with a default of 20%, corresponding to a CPU workload of 80%), it is assigned one more CPU in order to ensure that any increase in load for this task group in the next time interval can be adequately handled.
- Task groups are assigned to CPUs in such a way that all the CPUs have a similar workload, as far as this is possible.

At each change of task, the **scheduling algorithm** in PRIOR checks whether a task is allowed to run on the CPU according to the assignment made by the assignment algorithm. An “idle in preference to foreign” strategy is applied, i.e. idle time in a CPU is preferred rather than initiating a foreign task (which would have to rebuild the contents of the cache entirely).

Use of TANGRAM

Improved use of the performance available from multiprocessor hardware depends on the following conditions:

- number of CPUs (greater advantages with more CPUs)
- hardware architecture (large store-in caches)
- application (proportion of write operations, options for distribution over a subset of the available CPUs)
- workload (significant advantages are only achieved with high workloads)

It is difficult to make generalized statements with regard to the advantages of using TANGRAM. Measurements with different workloads showed an increase in hardware performance of approx. 5% with biprocessor systems and approx. 10% with quadroprocessor systems.

8.3 Data management

Management of volumes and access paths

The device management facility consists essentially of the following components:

- **Resource allocation** This function controls and monitors the logical statuses (occupied/available) of the resources (devices, volumes) requested by the user.
- **Resource reservation** This function processes requests for resources. These requests are formulated by the user with the aid of /SECURE-RESOURCE-ALLOCATION.
- **Reconfiguration** This function permits the operator to add or remove hardware units and their virtual connections on the command level.
- **Volume monitoring** This function monitors the accessibility of volumes, particularly during mounting procedures.

In the case of resource allocation the following points should be noted with reference to system overhead:

Each time a volume is occupied, information to this effect is stored in its standard volume label (SVL). In the case of private volumes, system overhead varies, depending on the type of allocation (task-exclusive/task-shareable) and the time the allocation is made (ASSIGN-TIME=*USER/*OPERATOR).

- When the time the allocation is made is determined by the user and the allocation is task-exclusive /SET-DISK-PARAMETER VOLUME=vsn,ASSIGN-TIME=*USER, USER-ALLOCATION=*EXCLUSIVE the SVL on the volume is updated at the beginning and end of each file operation involving catalog access (/COPY-FILE, /DELETE-FILE, /CREATE-FILE, /MODIFY-FILE-ATTRIBUTES, OPEN, CLOSE).
- When the time the allocation is made is determined by the user and the allocation is task-shareable /SET-DISK-PARAMETER VOLUME=vsn,ASSIGN-TIME=*USER, USER-ALLOCATION=*SHARE the SVL is read when the first user attempts access, and written back when the volume is released by the last user. In the meantime, the SVL on the volume is not updated, but internal table management activities take place whenever there is a file operation involving catalog access.
- When the time the allocation is made is determined by the system (activation interrupt) SVL updating is performed once for each session, regardless of the type of allocation.

Recommendations

- The system should determine the time of allocation.
  This is set by means of `/SET-DISK-PARAMETER VOLUME=vsn,ASSIGN-TIME=*OPERATOR,USER-ALLOCATION=*EXCLUSIVE/*SHARE`

  Release of the volume (special case) can be performed by means of: `/SET-DISK-PARAMETER VOLUME=vsn,ASSIGN-TIME=*USER`

- In the case of shareable private volumes, the parameter `SYSTEM-ALLOCATION=*SHARE` in `/SET-DISK-PARAMETER` should only be used if more than one system with write authorization actually accesses this volume simultaneously.
  As already mentioned (see section “Logical operating modes of disks” on page 145), the overhead necessary to coordinate file operations involving catalog access is very high.

  As soon as only **one** system is still accessing a shareable private disk, the following command should be issued:
  `/SET-DISK-PARAMETER VOLUME=vsn,SYSTEM-ALLOCATION=*EXCLUSIVE`

9 Internal procedures

To facilitate interpretation of the measurement figures, the sections below explain in greater detail the internal procedures of the BS2000 operating system insofar as they concern throughput and response time behavior; they also present various reference values for the resource workload.

9.1 Interrupt analysis (CPU states)

There are 4 CPU states in BS2000. Of these, the TU, TPR and SIH states are essential to normal operation. The MEH state is used solely to handle hardware errors.

The following definitions apply:

TU Task Unprivileged

TPR Task PRivileged

SIH System Interrupt Handling

MEH Machine Error Handling

The user task (application program) runs in CPU state TU.

System routines which may be assigned directly to the user task (e.g. SVC: RDATA) are processed in CPU state TPR (see figure 13 on page 240).

CPU time taken up in the TU and TPR CPU states should be regarded as productive time from the user's standpoint.

The basic control mechanisms of BS2000 are run in CPU state SIH. The SIH CPU state cannot be interrupted except after MEH.

System tasks provide support for organizational tasks; they run in CPU state TPR (see figure 14 on page 240).

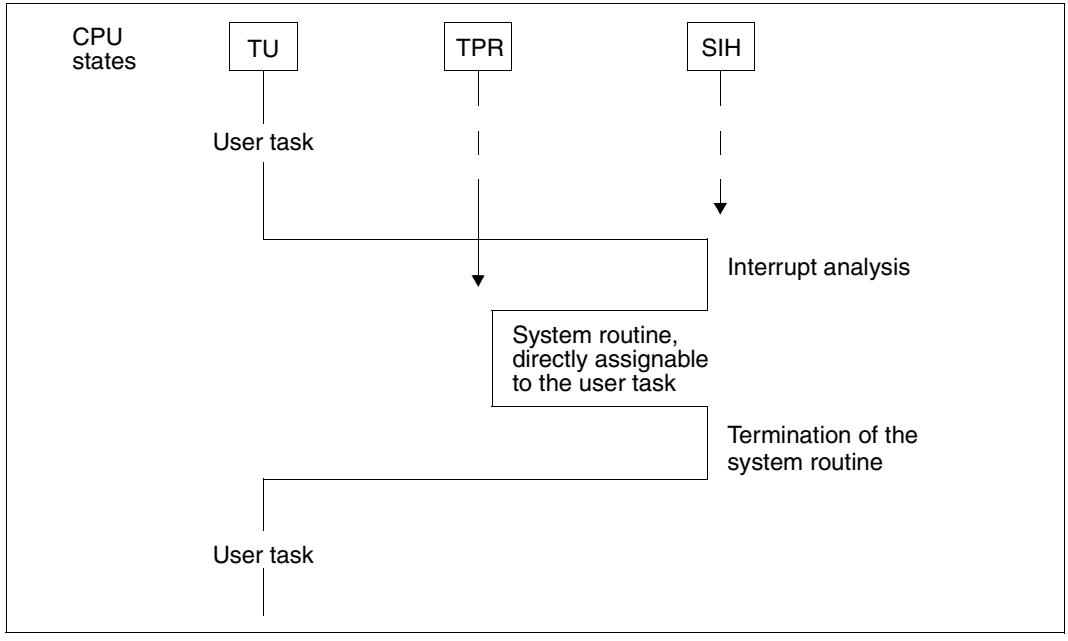


Figure 13: BS2000 CPU states (user task)

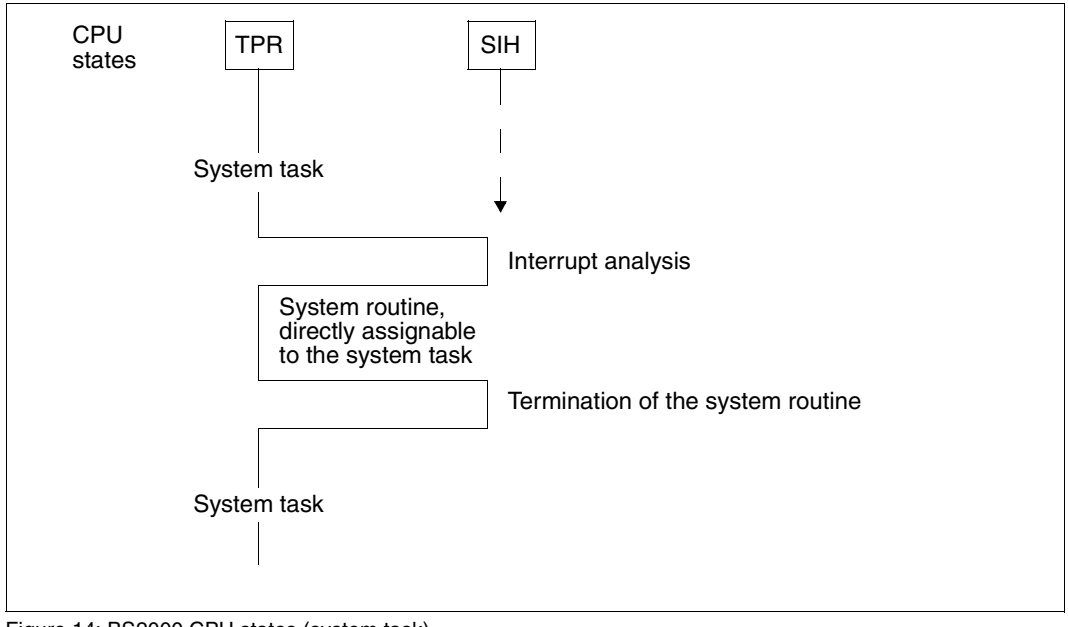


Figure 14: BS2000 CPU states (system task)

An interrupt is caused either by a user/system task which requires the support of the operating system or by an event which must be handled by the operating system.

There are five types of interrupt:

- SVC interrupt (=task-related interrupt)

  A user or system task requests certain system services. The calling task is charged not only for the time for the system routine, but also for the interrupt analysis and the termination of the system routine.

- Program interrupt (=task-related interrupt)

  This includes “genuine” errors which cause the program to abort (address error, data error, etc.). These errors may be ignored when considering throughput. Also included are two interrupt causes:

  – page not in memory (event code 4C)

  – address translation error (event code 48)

  In each case it is impossible to continue the program run for the time being since the addressed page is not immediately available.

  There are two possibilities in the case of event code **4C** (page not in memory):

  – The page is still in main memory, namely in the “free pool” (read-only queue or read/write queue):

    The page, which has not been addressed for some time, has been entered in the “free pool” by paging management. The program can be resumed following a simple page chaining procedure (PAGE RECLAIM). As a result of the global working set strategies, page reclaiming is a very rare occurrence (see section “Main memory management” on page 246). In most cases, it is the result of the first access to a page (First Page Access).

  – The page is not in main memory and must be entered there by means of a paging I/O operation (PAGE READ).

  Event code **48** involves a genuine error in address translation, which causes the program to abort.

- Timer/external interrupt (= asynchronous interrupt)

  A signal from the interval timer or ETC (Elapsed Time Clock).

- I/O interrupt (= asynchronous interrupt)

  A signal that an event has occurred in connection with input/output. (In most cases, this involves the occurrence of an end condition when performing the I/O operation.)

- Hardware error interrupt (= asynchronous interrupt)

  A signal that a hardware error has occurred.

Once the interrupt has been treated (with the exception of the SVC interrupt), the dispatcher (selection) routine of the task initiator will select and pass control to the executable user or system task with the highest priority (task in control; see also section “CPU management” on page 258).

9.2 Creating and terminating tasks

BS2000 is equipped to handle user tasks and system tasks:

User tasks

User tasks are created with the aid of system tasks at the request of the user. They are subdivided according to the type of task into

- dialog tasks
  (task type X'40', created with `/SET-LOGON-PARAMETERS` by the system task DIAA)
- batch tasks
  (task type X'20', created with `/ENTER-JOB` by the system task TSC)

The task attribute can be distinguished from the task type. The former influences capacity allocation in the current session. The following task attributes are available for the different applications:

- TP for transaction applications
- DIALOG for dialog (interactive) applications
- BATCH for batch applications

Dialog and batch tasks automatically receive the task attribute DIALOG or BATCH.

The tasks of transaction applications (DCAM, UTM, UDS, SESAM/SQL, etc.) are run under the same task type used when they were started (as a batch or interactive task).

If a job class is defined using the parameters `TP-ALLOWED=*YES(CATEGORY=xy),START-ATTRIBUTE=*TP`, these tasks are then given the task attribute TP as an additional identifier (see section “Job classes” on page 197).

The TP task attribute can also be applied with the TINF macro while simultaneously switching to the category with the default name TP.

System tasks

System tasks (task type X'80') automatically receive the task attribute SYSTEM.

The following preallocated system tasks and important, dynamically generated system tasks exist (see the “Introductory Guide to Systems Support” [10]).

9.3 Managing the resources main memory and CPU

Both user tasks and system tasks are monitored and controlled by the task management routines.

For a task to be able to run, task management must make the following decisions:

- allocation of authorization to use main memory → activation
- allocation of the resource CPU → initiation

Initiation is possible for already activated tasks only.

Criteria for managing **main memory**:

- multiprogramming level (MPL) per category
- Priority
- resource workload (CPU, main memory, paging activity)
- system services rendered (CPU time, number of I/O operations).

Criterion for managing the resource **CPU** is only the priority

If multiprocessors are in use and the TANGRAM function has been activated, the assignment algorithm can result in a different initiation sequence (see also section “Introduction to the TANGRAM concept” on page 233).

Internal category identifier

Both task categories and priorities are given a system-internal representation which differs from that at the user interface.

The default categories have fixed identifiers:

Category	Category identifier
SYS	0
DIALOG	1
BATCH	2
TP	3
xy	4
. . .	. . .

Internal priority

The specified variable priorities are weighted according to the importance of the category concerned (represented by the WEIGHT-CODE parameter). The variable internal priority is a function of:

- the external priority
- the WEIGHT-CODE (W) of the category to which the task belongs
- the sum S of the weights of all categories
  $S = W_{SYS} + W_{TP} + W_{DIAL} + W_{BATCH} + W_{xy} + ...$

```
Variable internal priority =
1 + (256 - external priority) * 0.5 * (1 + W / S)
```

Fixed external priorities assigned by the user are converted as follows:

```
Fixed internal priority = 256 - external priority
```

The internal priority serves as a basis for calculating the activation and initiation priorities.

9.3.1 Main memory management

The following concepts are employed:

- Activation (see page 247)
  Allocation of the authorization to use main memory to an executable (inactive, ready) task.
- Deactivation (see page 249)
  Withdrawal of the right to use main memory when expecting long wait periods or after providing certain system services.
- Forced deactivation (see page 250)
  Automatic deactivation of a task due to a resource overload.
- Preemption (see page 251)
  Activation of one task at the cost of deactivating another task.
- Control functions of main memory management (see page 252)
  - Activation Control Function (ACF)
  - Preemption Control Function (PCF)
  - Paging Load Control Function (PLC)
  - System Service Slice Runout Control
  - Waiting Time Runout Control
- Paging management algorithm (see page 256)
  - system working set method (SYS-WS)

The above concepts - with the exception of “Page management algorithm” - are only significant in the rare event of a main memory bottleneck.

Activation

Activation is possible only when permitted by the Activation Control Function (ACF) on the basis of the CPU and main memory workload measurement and the paging activity (see "Control functions within the framework of main memory management" on page 252).

The primary goal is **to attain the MIN MPL value** (MINIMUM-ACTIVE-TASKS) specified by the user **in each category.**

The WEIGHT-CODE parameter affects activation.

A decision regarding the index is taken (with NAT, number of active tasks in the category for which the index is calculated) if one of the following three criteria is satisfied:

- If some categories have not yet reached MIN MPL
- if MIN MPL has been exceeded in every category but MAX MPL has not yet been reached
- if MAX MPL has been exceeded in every category

then the following index decision will be made:

```
Index = (NAT + 1 – N) / WEIGHT
```

where:

```
N = 0,          if MPL < MIN MPL
N = MIN MPL,    if MIN MPL ≤ MPL < MAX MPL
N = MAX MPL,    if MAX MPL ≤ MPL
```

Activation occurs for the category with the lowest index number. From this category, the task with the highest activation priority is selected.

Exception
If the MIN MPL value has already been reached for the calculated category and there is an activation request for a fixed-priority task from another category, the latter task will be activated, provided this is permitted by the ACF control function; if not, preemption will take place.

Tasks with system priorities are always activated immediately. Here, too, the ACF control function decides whether this is possible within an activation operation or whether preemption should take place.

Activation priority

For variable external priorities, activation priority is dependent on:

- the variable internal priority
- the CPU state (TU or TPR)
- the point in time t_A when the most recent activation occurred (in each case the difference Δ is determined) and
- a configuration factor `C`

```
Variable activation priority =
(Variable internal priority + P) * Δt_A(Elapsed time) / (Δt_A(CPU time) * C)
```

Tasks waiting for authorization to use main memory and in CPU state TPR receive a higher activation priority; this is expressed by the constant P.

`P = 5` If in TPR

`P = 0` If in TU

The configuration factor `C` depends on the server type used and it is increased dynamically as the paging rate increases.

```
Fixed activation priority = Fixed internal priority
```

Deactivation

Deactivation of active tasks makes main memory available for inactive ready tasks.

Criteria for deactivating active tasks:

- further processing is impossible,
  e.g. due to wait calls in the program (PASS, VPASS) or waiting for terminal input in interactive mode (WRTRD, RDATA)
- wait periods were exceeded while waiting for events,
  e.g. waiting for terminal input in TP mode

  DCAM: YRECEIVE
  UTM: KDCWAIT

  Task communication
  ITC: REVNT
  TU eventing: SOLSIG
  Forward eventing: RSOFEI

  (See “Control functions within the framework of main memory management”: “Waiting Time Runout Control” on page 255.)
- certain system services in the form of CPU time and the number of I/O operations have been rendered
  (See “Control functions within the framework of main memory management”: “System Service Slice Runout Control” on page 254.)

Deactivation is suppressed when the function “No Deactivation” (TINF macro: DEACT parameter) is switched on.
Exception: the wait call VPASS n always causes deactivation.

Forced deactivation

Forced deactivation of tasks occurs when the Activation Control Function (ACF) detects an excessive workload on the resources CPU and main memory or an excessively high paging rate in relation to the CPU type (matrix decision).

Within an ACF measurement interval, only one task is deactivated at a time.

The following category attributes are used to determine the category from which a task is to be forcibly deactivated:

- MIN MPL
- MAX MPL
- WEIGHT

The user should ensure that only tasks from “less important” categories are deactivated by specifying correspondingly high MIN MPL and WEIGHT values for categories with important applications.

For the most part, forced deactivation applies to low-priority tasks which are in the process of building up their working set. As a result, non-productive system overhead is kept to a minimum.

Forced deactivation will not occur for:

- tasks with a fixed priority
- tasks in CPU state TPR
- tasks which have switched on the “No deactivation” function.

Preemption

Preemption can take place whenever an activation request exists but the ACF control function forbids further activation in light of the CPU and main memory workload and the size of the paging rate.

Preemptions are set out in the report group PRIOR-ACF of openSM2. They should be avoided at all costs. They may be caused when:

- The MIN MPL values are set too high.
- There are two many tasks with a fixed priority.

There are two types of preemption:

a) preemption between **different** categories

Active task A is preempted by inactive task B belonging to another category. This occurs when:

– for the category with inactive task B has MPL < MIN MPL

– at the same time that the category with active task A has MPL > MIN MPL

Preemption of tasks from different categories:

1. Activation of a task from a category with MPL < MIN MPL and the lowest category index number. The task with the highest priority from this category will be activated.
2. Forced deactivation of a task from a category with MPL > MIN MPL
(see “Forced deactivation” on page 250)

b) preemption within the **same** category

Active task A and inactive task B belong to the same category. This is only possible when:

– inactive task B has a fixed priority

– active task A has a variable priority.

Preemption of tasks within the same category:

1. Forced deactivation of a variable-priority task from this category
(see “Forced deactivation” on page 250)
2. Activation of a fixed-priority task.

Control functions within the framework of main memory management

- Activation Control Function (ACF)

  ACF measures the CPU and main memory workloads and the paging activity at periodic intervals (→ INVOCATION LONG = INVOCL, see report group PRIOR-ACF of openSM2).
  The polling interval lies between 0.25 and 2 seconds, depending on the CPU workload.

  CPU UTILIZATION

  Value calculated on the basis of the CPU workload measured during the interval and the time spent waiting for CPU allocation.

  MEMORY UTILIZATION

  Ratio of working set requirement estimated by the system for all active tasks (sum of PPC values) to the number of pages available for paging (NPP = Number of Pageable Pages).

  PAGING UTILIZATION

  Paging rate.

  The measured values are divided into the following classes:

  – H (High)
  – M (Medium)
  – L (Low)

  The reference values for utilization are given in the report group PRIOR-ACF of openSM2. The limit values for these classes are defined at system startup on the basis of the server used (CPU speed, main memory configuration).

  Depending on the resource workload and the specified category attributes or task priorities, the system decides on the basis of a matrix whether further activation is permitted or whether forced deactivation, preemption, or even no action at all should take place.

  Before a task is actually activated, ACF makes an additional measurement regarding solely the main memory workload in order to ensure that the workload on main memory will permit further activation (→ INVOCATION SHORT = INVOCS, see report group PRIOR-ACF of openSM2).

- Preemption Control Function (PCF)

  Task preemption at the activation level is time-consuming since the working set of the preempted task must be reconstructed during reactivation. In order to keep the resultant overhead within reasonable limits, the system tries to keep the preemption rate as low as possible.

  PCF periodically monitors the flattened preemption rate per interval. Depending on the hardware used, the polling interval lies between 20 and 48 seconds.
  If at least one preemption occurs during this interval, the MAX MPL OPTION will be switched on. This means that from this point on the MAX MPL value must be regarded as an inviolable boundary beyond which activation will not take place. (The MAX MPL OPTION is switched off when no preemption occurs over several intervals).

  PCF takes countermeasures to keep the preemption rate low. The strategy behind these measures is to carry out preemption requests, wherever possible, as additional activations rather than as preemptions. To do this, PCF reserves main memory areas for preemption requests whenever a certain preemption rate per interval (preemption limit) is exceeded.

  The memory area reserved for preemption requests is enlarged step by step as the preemption rate increases (5% to 15% of the real main memory configuration depending on the size of main memory). This simulates a heavier main memory workload, with the result that tasks with no preemption request will not be activated, leaving sufficient main memory to activate tasks **with** a preemption request.

  The system distinguishes between 4 different levels (preemption levels 0 to 3), which are displayed with the message `EXC0455 TASK PREEMPTION LEVEL=(&00)`:

  Level 0 Normal state (preemption rate < preemption limit)

  Level 1, 2 Short-term overloads
  The preemption level continues to climb if
  – preemption rate > preemption limit
  – preemption rate in interval n + 1 > preemption rate in interval n.

  Level 3 Long-term overload. (This can arise when the MIN MPL parameters are set too high: tasks from categories which have not yet reached MIN MPL automatically issue a preemption request.)
  The operator is told that preemption level 3 has been reached and as of this point receives notification of every change made to the preemption level. If the system discovers by means of a lowered degree of multiprogramming that the MIN MPL parameter setting is too high, the operator will be asked to check this setting and to lower the MIN MPL value of suitable categories.

  As the preemption rate decreases, reserved main memory will be released step by step. Once the normal state has been reached, the operator receives the message:
  `EXC0456 PREEMPTION CONTROL FUNCTION TERMINATED`

If preemption messages are output, steps should be taken to remedy the situation (see “Preemption” on page 251).

- Paging Load Control Function (PLC)

  PLC carries out long-term measurements of paging activities (polling interval 20 to 48 seconds, depending on the speed of the CPU).It can be seen as a supplement to ACF. While ACF measures the current paging rate and compares it to the predefined limit values in order to make short-term decisions, PLC affects the limit values defined at system startup, depending on the workload combination.

  If the workload is dialog-oriented, higher paging activities are allowed as compared to workloads oriented towards TP or TP/batch tasks. This minimizes the negative effect of paging activities on TP-oriented workloads and optimizes throughput in the case of batch-oriented workloads.

  PLC keeps statistics regarding the frequency with which the states

  – underload
  – medium load
  – overload

  occur per polling interval with respect to paging activities. If the “overload” state occurs once per polling interval, the MAX MPL OPTION will be switched on. The MAX MPL OPTION will be switched off if the “paging overload” state fails to occur in the course of several intervals, the predominant preemption level is 0 and the preemption rate is less than 1.

- System Service Slice Runout Control

  The “system service slice” is used to add up the system services received by an active task in the form of CPU time (in CPU states TU and TPR) and of I/O operations performed.

  System service is calculated using the formula `SERVICE = (A * B * C) + D`, where:

  A: Mean instruction execution time (depending on CPU speed).

  B: Mean number of instructions per I/O request.

  C: Number of I/O operations since last activation.

  D: CPU time since last activation.

  The size of the “system service slice” depends on the category and the server.

  The polling interval for “system service slice runout control” is 2 seconds.

For purposes of recovery action, a distinction is made between two types of system service slice runout, depending on whether the task is in the TU or TPR CPU state:

– system service slice runout – CPU state TU

  Deactivation will occur only when an inactive ready task with higher priority is available in the **same** category.

– system service slice runout – CPU state TPR

  Deactivation will take place when “system service slice runout” occurs for the 2. time if there is an inactive ready task with a higher priority in the same category. If the task switches to state TU after the first runout, it is immediately handled as described above.

- Waiting Time Runout Control

  The “waiting time limit” function is used to monitor tasks which are waiting for an event (bourse or inter-task communication events) and thereby occupy main memory.

  The “waiting time runout control” function is only called when more than 25% of pageable main memory is occupied by tasks waiting for bourse events (e.g. DCAM: YRECEIVE, UTM: KDCWAIT, TU eventing: SOLSIG, forward eventing: RSOFEI) or ITC events (inter-task communication: REVNT). The polling interval lasts from 1 to 7 seconds, depending on the main memory workload.

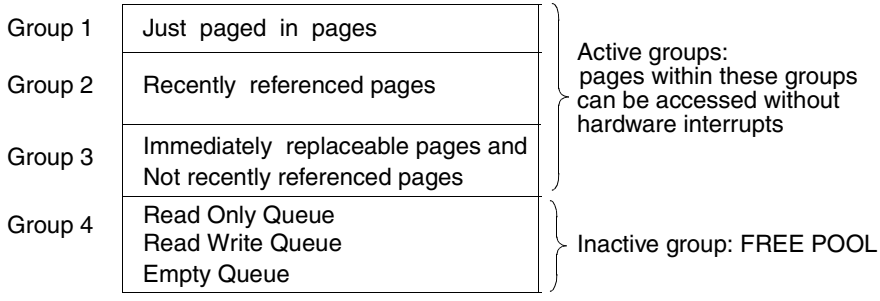
  If the “waiting time limit” is exceeded for one or more tasks, deactivation will take place in the following manner:

  Variable-priority tasks belonging to a category whose MPL value is greater than MIN MPL are deactivated first (provided deactivation is permitted in connection with “waiting time runout”: DWTR parameter in the TINF macro).
  The process is terminated as soon as the amount of pageable main memory bound by the waiting tasks falls beneath the 25% limit.

  If this action is not sufficient to fall beneath the 25% limit, the deactivation process will be extended to include waiting tasks with fixed priority as well as categories whose MPL value is less than MIN MPL (provided deactivation is allowed). Regardless of the 25% limit, all tasks which satisfy the specified conditions and have been waiting for longer than 30 seconds will be deactivated.

Paging management algorithm (System working set method, SYS-WS)

This method manages the main memory pages globally.The pages are split up into four groups according to their “access age” as follows:



The pages in each group are chained with one another.

Principle of execution

When the event “page not in main memory” occurs (page fault), a page frame from group 4 is filled by the requested page by means of a paging I/O operation (Page Read) and entered in group 1 (group transition 4 → 1); reference bit = ON.

As a rule, as many pages as possible are kept active. As a result the “free pool” is relatively small. However, it must have a certain minimum size in order to satisfy paging requests that follow in rapid succession.

Once this minimum size has been reached a routine is initiated to fill the “free pool” with pages from group 3 (group transition 3 → 4).

Pages whose reference bit is ON are added to the end of the chain of group 2 and the reference bit is deleted.

Pages which are found twice in a row with reference bit=OFF are placed in group 4.

Group 3 continues to be searched until the required number of pages to fill the “free pool” has been found.

If there are not enough available pages in group 3, group 3 is filled by transferring all the pages in group 2 to group 3 (group transition 2 → 3).

In the same way group 2 is then filled by rechaining all the pages from group 1 to group 2 (group transition 1 → 2).

Since the paging requests both fill group 1 automatically and initiate the page replacement algorithm, consistency is ensured.

The simple SYS-WS algorithm results in a reduction in paging management outlay, albeit at the expense of reduced accuracy in determining the working set requirements of the individual tasks. The reduction in outlay is especially noticeable in the case of large main memories.

If a task is deactivated, it retains its working set even if it is in the inactive state. If the task is reactivated, no “page reclaims” are necessary to restore the working set.

While the SYS-WS procedure knows the UPG value (used pages) of every task, it has no means of determining which phase the task is in (Even inactive tasks have a UPG value $\neq 0$.) .

Calculation of the PPC value (planned page count) is therefore approximate: the UPG value is compared with an assumed mean working set value, dependent on the category type.

Since as many pages as possible are kept active, the total UPG value is practically always in the magnitude of the available paging main memory and has only limited relevance. The working set requirements are expressed exclusively by the PPC value. This value represents only a rough estimate. With todays large memory configurations advantages result from the fact that no outlay is necessary for estimating exact memory requirements, as they are superfluous.

9.3.2 CPU management

The following concepts are relevant:

- Initiation
  Allocation of the resource CPU to an active, executable task (which is in the “active, ready” state).
- Deinitiation
  Withdrawal of the resource CPU when delays are expected (transition to the “active, not ready” state).
- Preemption
  Deinitiation of a task followed by initiation of another task with a higher priority.
- CPU management control function
  Micro Time Slice (MTS)

Initiation

From the available active ready tasks, the dispatcher routine of the task initiator selects and passes control to the task with the highest initiation priority (→ task in control).

With multiprocessors there are CPU-local queues (see also section “Controlling tasks via queues” on page 261).These queues are populated cyclically when tasks are activated. For each CPU, the task with the highest initiation priority is selected from the queue assigned to that CPU. Higher priorities which may be assigned to tasks in queues which belong to other CPUs are of no significance. The strategy “foreign tasks take precedence over idle states” is also adopted, i.e. a task from a “foreign” queue is initiated before the CPU becomes idle as a result of underloading. In this case, those queues containing the most tasks (“longest Q1”) are accessed.

If the TANGRAM function is used, the local CPU queues are populated according to the TANGRAM assignment algorithms rather than cyclically (see section “Introduction to the TANGRAM concept” on page 233). In these circumstances, the “idle in preference to foreign” strategy is adopted.

Initiation priority

Calculation of the initiation priority is analogous to that of the activation priority. The following additional points are taken into account:

- the point in time t_A of the most recent system service slice runout
  (the difference: current point in time minus t_A is formed)
- the “task in control” is given preference when calculating priorities
  (the formula P = P + 5 is used in order to keep the preemption rate as low as possible)

Deinitiation

In the interest of efficient CPU utilization, the system will withdraw control from a task (even if this task has top priority) whenever there are immediately foreseeable wait periods:

- For short wait periods there is a transition to the “active, not ready” state.
  This applies to wait periods such as: performing an I/O operation (DMS I/O, paging I/O), waiting for bourse or ITC events, VPASS 0 wait call in program.
- For long wait periods deactivation will also occur.
  This applies to wait periods such as: terminal input in interactive mode, PASS wait call, VPASS wait call in program.

Preemption

At the initiation level, preemption is entirely dependent on the initiation priority.

In order to preempt the “task in control”, the priority of the task to be newly initialized must be:

- higher by at least the value 5
- higher by the value 10 if the task in control is in CPU state TPR.

After an SVC call which has not led to deinitiation of the task, no preemption takes place. Only when another task moves to the head of the CPU queue while this SVC is being processed, and when the above-mentioned conditions are fulfilled, does the task calling the SVC lose the CPU.

CPU management control function

- Micro Time Slice (MTS)

  The “micro time slice” is the maximum amount of CPU time which a task can occupy without interruption.

  The CPU time used in the TU and TPR CPU states is added together to determine the CPU intensity of the individual task. In the case of a “micro time slice runout”, the priority is checked or redefined. When multiprocessors are used, following a “micro time slice runout” the task with the highest priority is selected from all the queues (Q1).

  The size of the MTS varies between two outer limits, and is dependent on CPU intensity. After each I/O operation involving a wait period, the MTS is redefined:

  ```
  MTS = Current CPU time - CPU time of the penultimate I/O
  ```

  The limits are dependent on the CPU speed; the faster the CPU, the lower the limit values.

  The more I/O-intensive the characteristics of a task, the smaller its associated MTS (which, however, must not fall beneath a certain minimum value). Since I/O-intensive tasks have low CPU intensity, no “micro time slice runout“ occurs.

  If the task characteristics change in the direction of greater CPU intensity, a “micro time slice runout“ occurs relatively quickly, followed by a calculation of priority. This ensures that the system can respond quickly to changes in workload characteristics from I/O-intensive to CPU-intensive (as it does not have to wait for the periodic priority check). This type of workload change is particularly critical, since the domination of CPU-intensive tasks has a corresponding adverse effect on the degree of simultaneous processing, thereby reducing throughput.

9.4 Controlling tasks via queues

Tasks existing within an operating system can be in any of five different states:

- TASK IN CONTROL
- ACTIVE READY
- ACTIVE NOT READY
- INACTIVE READY
- INACTIVE NOT READY

The number of tasks in the state TASK IN CONTROL can only be as great as the number of CPUs the server has.

The remaining 4 states are wait states. They can be divided into 2 groups, depending on whether they have already been allocated the right to use main memory (ACTIVE) or not (INACTIVE).

Tasks are assigned to the individual wait states with the aid of queues; for reasons of clarity the “NOT READY” states are further subdivided into events to show why the task is not executable.

In order to improve the MP factor for multiprocessors, the queues for the active space of each CPU are provided separately (locally for each CPU) and the remaining queues are common to all CPUs (see figure 15 on page 262).

The PEND/UNPEND routines are used to change the state of a task (characterized by moving from one queue to another):

- UNPEND routines move the task from its present waiting position towards use of the CPU.
- PEND routines put the task in a wait state due to an intervening event.

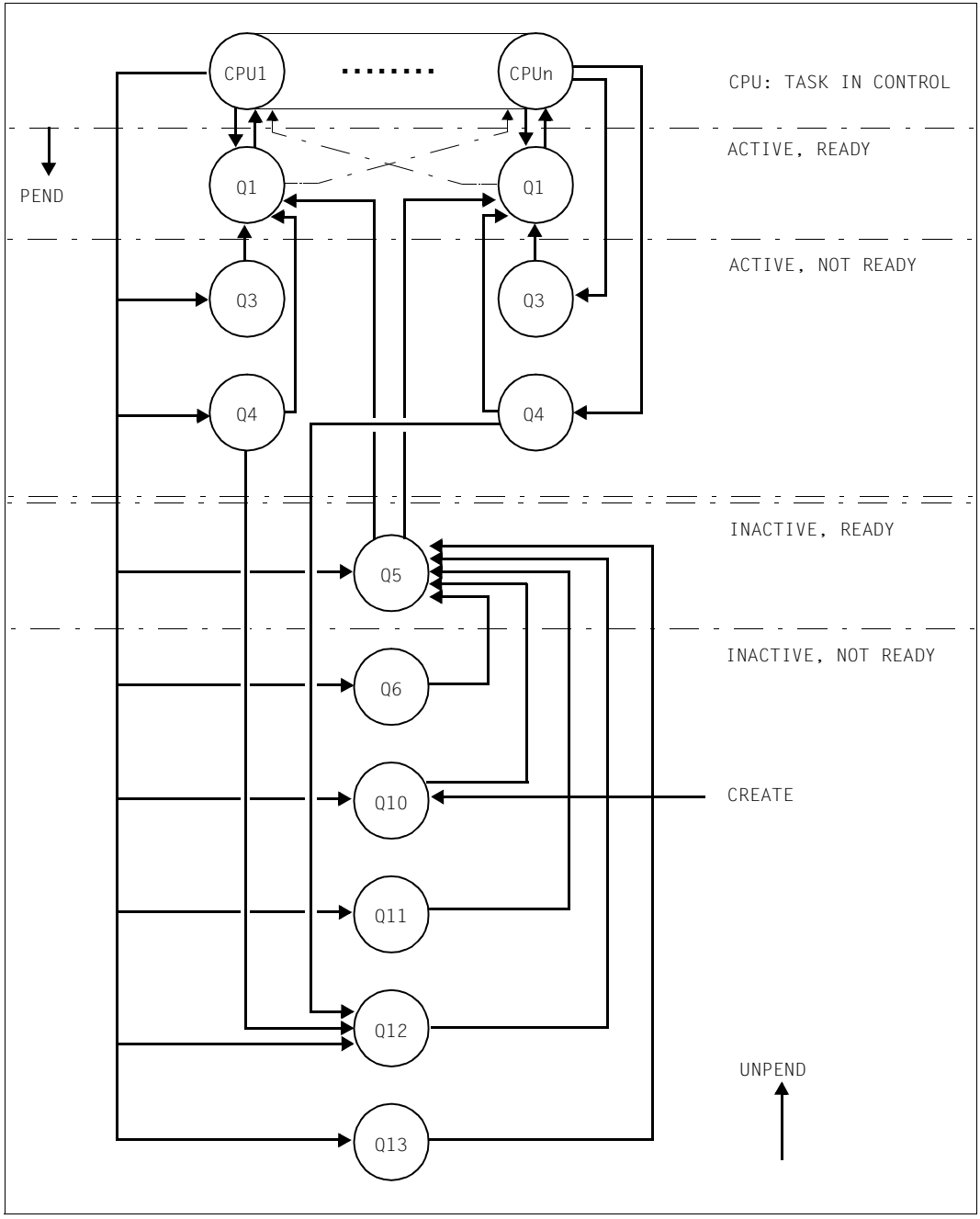


Figure 15: Queue transfers

Overview of the individual queues

Queues Q0, Q1, Q2, Q3 and Q4 are provided once for each CPU.

- TASK IN CONTROL

  Q0 Task occupies CPU

- ACTIVE, READY

  Q1 CPU queue:
  contains active ready tasks waiting to use the CPU.

- ACTIVE, NOT READY

  Q2 Special case (not shown in the diagram): monitoring system tasks

  Q3 Waiting for termination of paging I/O

  Q4 Waiting for termination of DMS I/O (disk, tape)
  Waiting for bourse events (“short wait”)
  Waiting for termination of `VPASS 0`, `VPASS MSEC=Y`

- INACTIVE, READY

  Q5 Memory queue:
  contains inactive ready tasks waiting for authorization to use main memory. This queue consists of 4 to 16 sub-queues corresponding to the following predefined categories:

  ```
  SYS
  TP
  DIALOG
  BATCH
  ```

  and up to 12 further categories which can be defined by systems administration.

  Q6 “PCS not admitted” queue.
  This queue contains inactive ready tasks which are not allowed to be activated by the Performance Control Subsystem.

- INACTIVE, NOT READY

  Q10 HOLD queue:
  Contains newly generated tasks (call: `$CREA`)
  Waiting for peripheral devices (`/SECURE-RESOURCE-ALLOCATION`)
  Stopped by systems support (`/HOLD-TASK`)
  Tasks cannot be terminated (serious errors)

  Q11 Inactive system tasks

  Q12 Waiting for terminal input (`WRTRD`, `RDATA`)
  Waiting for bourse events (“long wait”)
  Waiting for an answer to a console message

  Q13 Waiting for termination of `PASS`, `VPASS` calls
  Waiting for ITC event

9.5 Performing I/O operations

The processing of I/O operations in BS2000 is performed by the Data Management System (DMS). As compared to the execution of instructions in the CPU (microseconds or fractions thereof), the physical processing of I/O operations takes longer by a full order of magnitude (milliseconds) and thus has a decisive influence on time behavior.

In the case of interactive applications, the focus is on I/O operations on disks (more precisely: logical volumes). The section below describes in greater detail how the processing time for an I/O operation is divided up.

9.5.1 I/O operations on disks

At the **logical** level, the user formulates his/her I/O requests in the application program using macros (GET, PUT, GETKY, STORE, PAM, etc.). When higher programming languages are used, the READ or WRITE statements are converted accordingly by the relevant runtime system.

The expansion of the macro contains an equivalent SVC call which, however, is not executed with every macro.

In the case of sequential input operations (GET), the access method (SAM, ISAM) provides the next virtual record from the input buffer into which the data block was physically read. The SVC call is executed only when there are no further records in the input buffer.

In the case of sequential output operations (PUT), the access method (SAM, ISAM) performs the SVC call only when there is no space left in the output buffer.

Before an update operation PUTX (ISAM) can be performed, the corresponding records must have been read (GET, GETKY). Only when the record length increases in the course of the update will the SVC call be executed.

If the called DMS access method discovers that a physical I/O operation is to be performed, the SVC call `$EXCP` (Execute Channel Program) or `$XCPW` (Execute Channel Program with Wait) is used to trigger the I/O operation. These calls are not immediately available to the user.

The time relations on the channel or disk side are identical for both calls; the only difference is the reaction of the calling program routine.

With the $XCPW call, the program run is interrupted (placed in the wait state from Q0 → Q4 by the system-internal PEND routine) and will not resume until the I/O operation has terminated.

The Q4 queue is simply a “collector” for tasks waiting for execution or for termination of I/O operations (and bourse events).
In general, the length of Q4 does not necessarily reflect an I/O bottleneck. The important queues in this respect are those in front of the individual I/O devices (device queues, see also the report group DEVICE of openSM2).

With the $EXCP call, the program continues asynchronously to the processing of the I/O operation (performs other functions unrelated to the I/O operation). At a given time, it issues an explicit wait call. For the sake of simplicity, only the $XCPW call is presented in more detail below.

If the logical volume is occupied at the time of the $XCPW call (busy processing I/O operations for other users), a wait time is incurred in front of the volume. A higher volume workload causes a longer wait time (see section “Software service time” on page 268).

When the PAV function is used, there are several device addresses (alias devices) available for each logical volume (base device). Concurrent I/O jobs can be processed via these, which prevents excessive delays.

As soon as the volume is free (or at least a free alias device is available), the operating system sends the instruction START SUBCHANNEL (SSCH) together with the “Operating Request Block” (ORB), which contains a description of the possible access paths in the “Logical Path Mask”, to DCS.

If all access paths are occupied (which is very rare with multiple access paths), a corresponding wait time is incurred (function pending time: cf. the report group SERVICETIME of openSM2) before the I/O operation can be initiated by means of the instruction START IO (SIO). As a rule, the wait time for a free access path is so small as to be negligible.

On SQ servers, actual handling of the input/output (triggered by the command that is equivalent to SSCH) is performed by the I/O drivers of X2000 on a separate CPU.

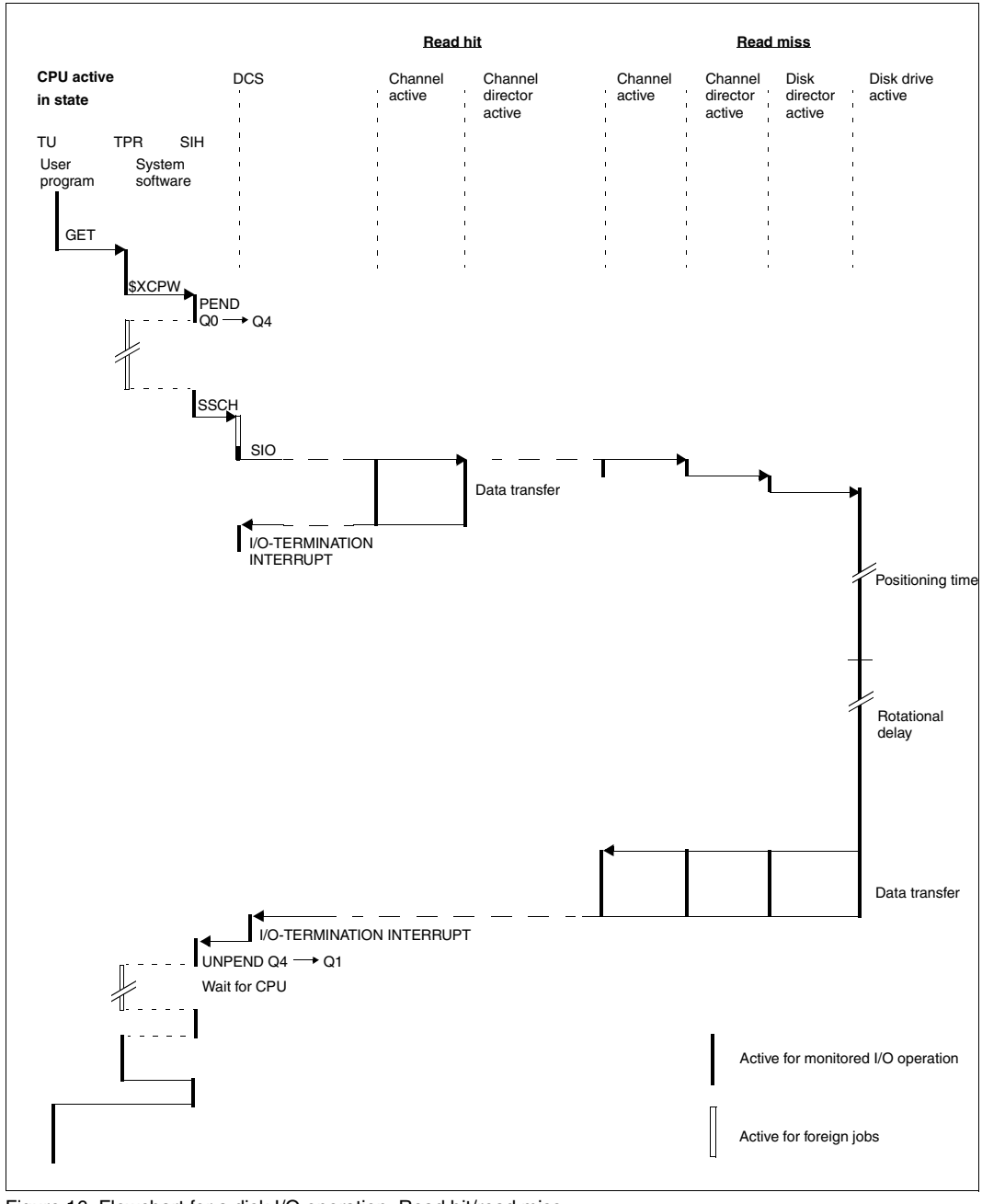


Figure 16: Flowchart for a disk I/O operation: Read hit/read miss

9.5.2 Hardware service time

The hardware service time refers to the time from the START SUBCHANNEL (SSCH) instruction through to the I/O termination interrupt.

In the case of disk storage systems, the hardware service time varies greatly depending on whether a cache hit or a cache miss is involved.

In the case of inputs/outputs with a block size of 4 KB, a typical value for a cache hit is approx. 1 ms. In this case the value for a cache miss is approx. 10 ms (see the section “Properties of external disk storage systems” on page 45).

9.5.3 Software service time

The software service time per I/O operation which the user sees mainly comprises:

- the wait time ahead of the volume concerned and
- the hardware service time for performing the physical I/O operation (see DISK report group of openSM2: SOFTWARE-DURATION and HARDWARE-DURATION).

The wait time W ahead of the volume is derived according to the rules for queue formation depending on:

- the distribution of the times between the I/O requests (interim arrival times),
- the distribution of hardware service times and
- the volume utilization.

Without delving further into queue theory, the following two formulas can be used to estimate the average wait time W:

1. When assuming the prerequisite “M/M/1” (worst-case scenario):

M	Exponential distribution of interim arrival times
M	Exponential distribution of hardware service times
1	one console

   ```
   W = S * U / (1-U)
   ```

S	Average hardware service time
U	Volume utilization

Example

```
S = 6 ms; U = 30%

W = 6 ms * 0,3 / (1 - 0,3) = 2,6 ms
```

When the volume utilization is 30%, the I/O requests must wait an average of 2.6 ms before the volume is released.

2. When assuming the prerequisite “M/D/1” (best-case scenario):

   M Exponential distribution of interim arrival times

   D... Constant hardware service time

   1 1 console

```
W = S * U / (2 * (1-U))
```

A comparison of the two formulas shows that, given a constant hardware service time (M/D/1), the average wait time W is only half as long as with exponential distribution of the hardware service time (M/M/1).

In practice neither a strictly exponential distribution of hardware service times nor a constant service time occurs. The wait times which occur as a result of the volume utilization are around the average of the results of the two formulas.

The table below illustrates the factor by which the hardware service time is dilated irrespective of the volume utilization.

Volume workload (%)	Dilation factor	
	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

The wait time ahead of the volume constitutes a major part of the I/O time. The utilization of volumes which are shared by multiple users should not exceed 30%.

When the PAV function is used, several device addresses (alias devices) are available for each logical volume (base device). This significantly reduces the volume utilization visible to the outside. (SM2 forms the mean value using the base device and the associated alias devices.)

10 Use of the openSM2 monitoring system

openSM2 (BS2000/OSD) is a software product for monitoring and measuring the performance of BS2000/OSD servers. It is ideal for observing load behavior and resource utilization, evaluating trends and analyzing problems. Introductory information on openSM2 is provided in the “openSM2” manual [18].

openSM2 (BS2000/OSD) and openSM2 (Open Systems) form the openSM2 product line, a powerful solution for monitoring heterogeneous IT environments.

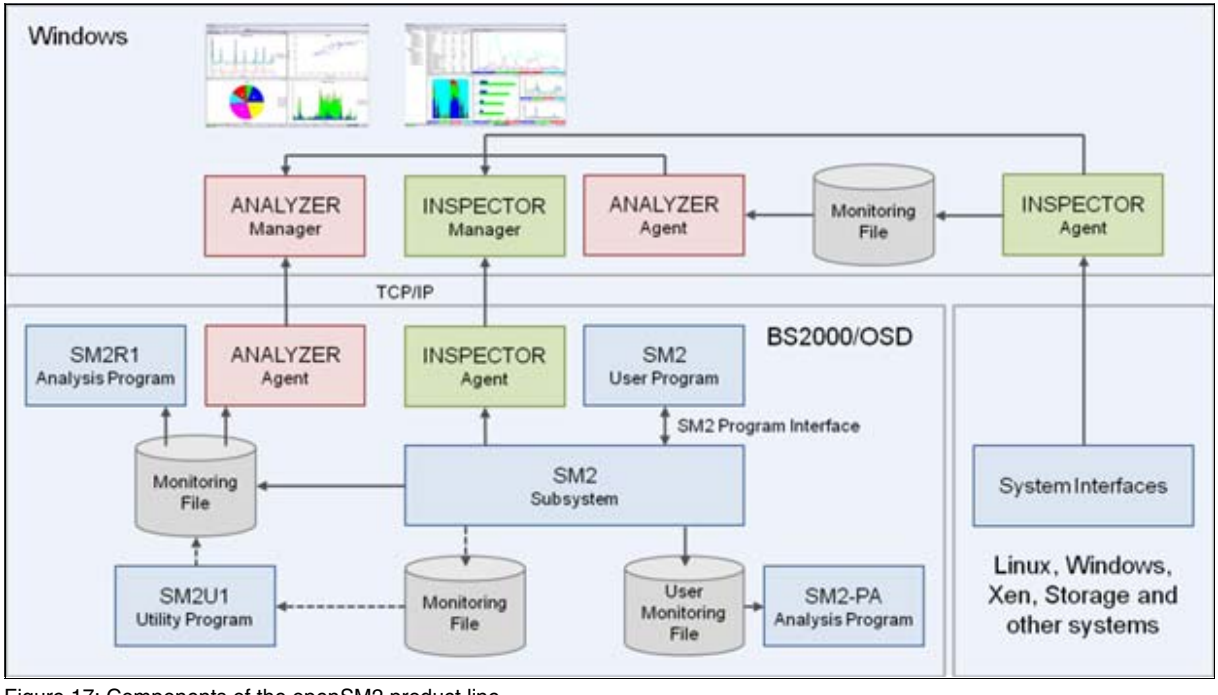


Figure 17: Components of the openSM2 product line

10.1 SM2 monitoring program

The SM2 monitoring program calculates the monitored data at regular intervals, the monitoring cycles (10 seconds to 1 hour). At the end of each monitoring cycle the monitored data is written to a buffer and to the output file.

The monitored data is recorded

- once at the end of the monitoring cycle (for ascending counters through calculating the differences to the data of the preceding monitoring cycle)
- during the entire monitoring cycle (for varying monitored variables)
  - Sampling
    by recording a sample several times in the monitoring cycle (sampling cycle) and calculating the mean value of all sampled values at the end of the monitoring cycle
  - Eventing
    by activating a monitoring routine when particular events occur, e.g. start of an input/output

Type	Accuracy	Overhead	Monitoring period	Data volume
Sampling	Dependent on the frequency of the samples	Dependent on the frequency of the samples	Any length	Manageable
Eventing	High	Dependent on the number of events	Short (15-30 min.)	Can be considerable

Use of the SM2 monitoring program is recommended as follows:

- Constantly (ongoing on the system)
- When load problems occur
- Before a new application is to be integrated into the server
- Before and after conversion to new versions (e.g. operating system, application software, databases)
- When new functions are used (e.g. NK-ISAM, PCS, DAB)
- When the hardware is changed

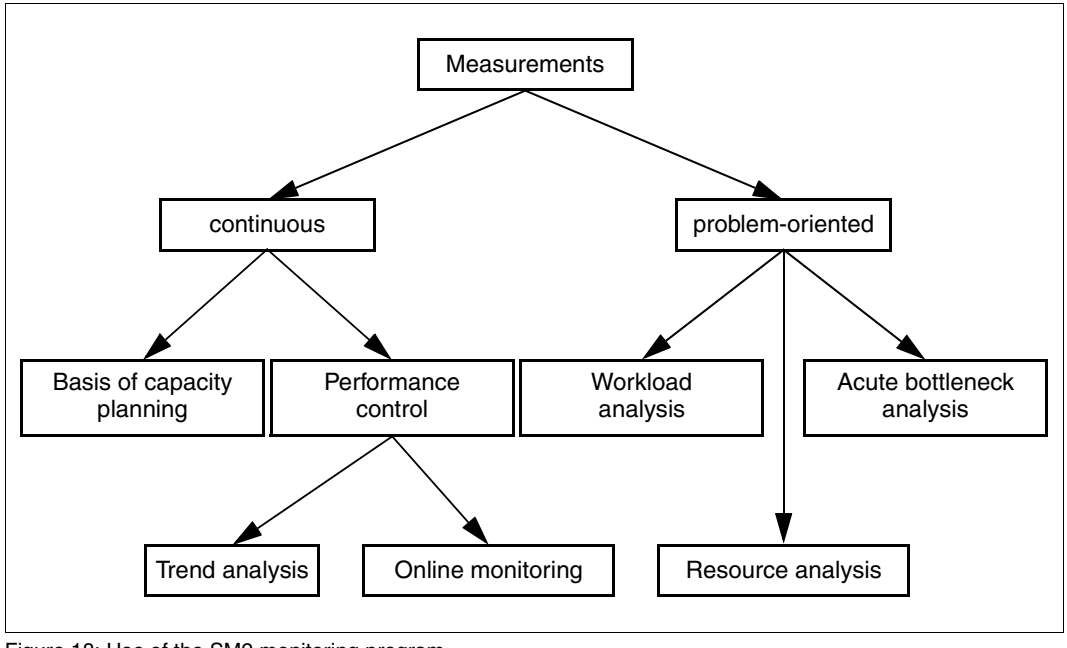


Figure 18: Use of the SM2 monitoring program

10.2 Routine system monitoring

10.2.1 Parameters for measurement and analysis

Each session should include an SM2 run, the results of which are entered in a file. With long-term measurements, the reliability of the measured data (which is dependent on the number of samples taken) can be ensured if one sample is taken per second during an analysis subinterval of one hour:

Sampling period (SAMPLING-PERIOD)	1000 milliseconds
Monitoring cycle (OFFLINE-PERIOD)	5 minutes
Analysis subinterval	1 hour
Monitoring period	from System Ready to Shutdown

It is not necessary to evaluate each and every session. However, at least random samples in the form of daily evaluations must be made quite frequently.
Analyses over longer periods of time are the best means of evaluating trends and providing a sound basis for backing up investment decisions. The number of annual reports should be kept to a minimum.
When unusual circumstances arise, e.g. in the event of complaints, queries or changes in the configuration, it is not sufficient to analyze solely the one day in question; analysis of a second day is additionally required for purposes of comparison. If the graphs for the two analyses are placed next to one another and the figures therein are compared, relevant statements can be made much faster by noting the differences. It should be borne in mind that corresponding graphs in the two analyses might be printed using different scales, which could make the before-and-after comparison somewhat more complicated.

SM2 is also used in the support of system measurements. To this end it supplies several reports which go beyond the interests of routine system monitoring based on standard measurement criteria.

10.2.2 Evaluating characteristic reports

For routine monitoring purposes, the following reports of the ANALYZER analysis routine have proved to be especially helpful:

Report group	Report	Meaning
CATEGORY-CPU	CPU utilization (TU + TPR) for category	CPU utilization for category [1]
CATEGORY-IO	IOs for category	DMS I/O rates for category [1]
	Duration of non paging IOs for category	Hardware/software service time for category [1]
CATEGORY-QUEUE	Active and inactive task for category	Number of tasks for category
CHANNEL	Utilization	Channel utilization [2]
CPU	Utilization real	CPU utilization (real value; also for VM2000)
	Sum SVC calls	SVC calls
DISK	Utilization	DIsk utilization (logical volumes)
IO	IOs for device classes	DMS I/O rates
	Paging IOs	Paging I/O rates
MEMORY	Paging area utilization	Paging area utilization
	Page frames	Size of the System Working Set
PERIODIC-TASK-JOBNAME		Utilization data for job name:
	CPU time	CPU share
	IO	I/O rates
PERIODIC-TASK-TSN		Utilization data for TSN:
	CPU time	CPU share
	IO	I/O rates
PERIODIC-TASK-USERID		Utilization data for user ID:
	CPU time	CPU share
	IO	I/O rates
VM2000	VM2000 utilization	Load shares of VM2000 guest systems [3]
	VM2000 Hypervisor	VM2000 hypervisor share [4]
WORKING-SET	Main memory utilization	Available main memory pages

[1] with monitoring program SYSTEM

[2] with type FC channel, only with monitoring program CHANNEL-IO

[3] with monitoring program VM (the values for all guest systems are supplied only on the monitor system)

[4] with monitoring program VM (only on S servers)

10.2.3 Comments on various reports

Reports of the report group BCAM-CONNECTION

These reports provide useful results on the share of response times caused in the server. The transmission times of the messages and, if applicable, the status in the network are not recorded. The reports are therefore only suitable for monitoring the load situation in the host system. Information on the response times experienced by the user (see chapter “Performance expectations from the user viewpoint” on page 15) cannot be provided.

Report “CPU utilization (TU+TPR) for category” of the report group CATEGORY-CPU

This report and the report of the report group CATEGORY-IO described below are only obtained when the SM2 monitoring program SYSTEM is activated. The report is used to find which category is the source of an excessively high CPU workload or to monitor the distribution of CPU capacity over the different categories. The latter is also of assistance in setting PRIOR or PCS.

Report “IOs for category” of the report group CATEGORY-IO

This report is used to monitor the number of I/O operations for the categories. It is of little relevance for system control, since the number of I/O operations of the category is not taken into account as part of category control. This report is more of a starting point for solving performance problems. For a more in-depth analysis of bottlenecks, the resource requirements (CPU time, number of I/O operations) can be recorded up to task level by using the TASK monitoring program.

Report “Utilization” of the report group CHANNEL

In the case of channels with disk peripherals connected, a utilization value of 60% should not be exceeded.

The CHANNEL-IO monitoring program must be activated to measure the utilization with type FC channel. The utilization cannot be measured directly with type FC channel, it is determined indirectly by comparing the actual number of I/O operations, under consideration of the block size, with a throughput value that could be achieved in laboratory measurements.

Channel utilization with VM2000 operation:

- Type S channel: since only the total utilization of a channel can be measured, it is not possible to determine the utilization parts of the separate guest systems. The total utilization of a channel by all guest systems is therefore always output in each guest system.

- Type FC channel: the utilization is only determined with activated CHANNEL-IO monitoring program.It is determined via the guest-specific I/O rate. The utilization is therefore always output related to guest system. The sum value of all guest systems should not exceed 60%.

Channel loads for magnetic tape cartridges may be up to 100%.

Excessively high channel loads may indicate that too many devices are attached to the same channel. However, the use of page chaining also makes the channel load too high. For this reason this function should only be used if it brings about distinct advantages. This is always the case if all the blocks transferred in one I/O operation are required and there is no need for further I/O operations.

Reports of the report group CPU

The “Utilization real” report shows the CPU utilization monitored (also in VM2000 operation). The “Utilization normed” report should not be used, in particular not in VM2000 operation!

Assessment of the CPU workload varies according to the type of application and type of load. A workload of 100% is feasible and desirable for pure batch applications. Balanced utilization of the remaining resources is to be strived for.

In the case of mixed loading with TP, interactive and batch applications, the main application should comprise no more than 70% of the total CPU workload for the system (in the case of multiprocessor systems, depending on the number of CPUs, up to 90% is acceptable for the main application). An ideal background load triggers neither I/O operations nor paging; being extremely compute-intensive it is well-suited to making use of surplus capacity. This combination of features makes it possible to take full advantage of CPU capacity; in practice, however, it seldom occurs. Any one I/O operation or paging transfer can interfere with the TP and interactive applications. I/O-intensive loads and programs with extensive memory requirements should not be loaded in the background. The background load must be made clearly subordinate to the main application by assigning appropriate priorities or PCS settings. Ideally, the files of the main application should reside on other disk drives (and at least on other volumes) and be accessible via other paths.

The SIH share depends, among other things, on the number of I/O operations of all kinds and the intensity of task communication. The SIH portion should not exceed 20% (S servers) or 10% (SQ servers). A high percentage of TPR operations is caused by SVC calls, which perform resource-intensive system services. The workload can be regarded as normal if the following formula applies:

`TU + TPR > 3 * SIH` (S server) or `TU + TPR > 7 * SIH` (SQ server)

Now and again there are instances where the load violates the above rules without actually impairing performance behavior. As long as the response times are good there is no cause for intervention in these cases.

Generally the applications run on an SQ server in /390 mode. As programs which run in this mode have a higher CPU requirement than on S servers, a high share in the processor state TU is normal (some 60 %, standardized as 100 % utilization). If a higher standardized TU share is shown, this means the performance of the SQ server is below the nominal value (the official RPF value). However, of the (standardized) TU share is lower, this means the performance is above the nominal value.

“Time” report of the DISK report group

The “Time” report is available only if the SAMPLING-DEVICE monitoring program with recording of hardware and software service times is activated. This report, in conjunction with the “Utilization” report, can be used to determine whether there is a heavy load on the disk peripherals. If software service times occur which are more than 10% over the hardware service times, I/O operations must sometimes wait in front of an occupied volume.
The SM2 monitoring program DISK-FILE and the SM2 online report DISK-FILE can then be used to ascertain the files affected.

If the application works with asynchronous inputs/outputs, an increased number of long software wait times can occur without a tuning measure being required or possible.

Report “IOs for device classes” of the report group IO

This report should be assessed in conjunction with the report “Utilization real” of the report group CPU. Certain loads are both compute-intensive and have a high number of I/O operations. Consequently, the most important criterion is that the channels and volumes are not overloaded. The standard values for the maximum number of DMS I/O operations per second are determined in accordance with the rule that (depending on the speed of the system) only 15 to 25% (S servers) or 8 to 12% (SQ servers) of the CPU capacity in processor states SIH+TPR should be used for processing the I/O operations. In certain situations it is feasible to exceed this limit.

See also section “Standard values for BS2000/OSD servers” on page 331.

Report “Paging IOs” of the report group IO

Paging enables the efficient utilization of a relatively small main memory as compared to the number of virtual address space requests.

Thanks to the availability of main memory at a reasonable cost, there is generally sufficient main memory to ensure that dynamic paging does not represent a problem. What follows below therefore only needs to be taken into account in special cases.

Paging is the result of a lack of memory. It burdens both the peripherals and processor. For this reason the paging rates (page reads per second) shown in section “Standard values for BS2000/OSD servers” on page 331 should not be exceeded. Violating the maximum recommended values is only permitted if the response times are acceptable **and** if all other workloads (CPU, channels, volumes) are low.

In the event of bottlenecks on the channel or paging volumes, it may be the case that the specified paging rates have not been reached, yet the response times are still unsatisfactory. This is caused by excessively long waiting times for PAGE READ. Poor configuration of the paging peripherals has serious detrimental consequences and must be avoided at all costs. Poor configuration is readily indicated by report „Utilization“ of the report group CHANNEL and DISK.

TP mode is very sensitive as far as paging is concerned - and even more so, the higher the number of important operations performed by relatively few central tasks. Such central tasks must not be burdened with paging I/O operations, since the resulting delays would be passed on to many other tasks. Paging tends to increase at breakneck speed. For this reason it is not advisable to exploit the main memory of a system to its full extent (see also the notes on the “Main memory utilization” report of the report group WORKING-SET as well as the sections “Monitoring main memory utilization” on page 31 and “Paging activities” on page 299).

Report “Paging area utilization” of the report group MEMORY

The maximum occupancy level of the paging area should be 50%.

If the average occupancy level of the paging area is not greater than 50%, peak paging loads with acceptable paging I/O times can be intercepted. Sufficiently large contiguous areas are then available for high-performance transfer of modified memory pages.

A considerable increase in response times occurs at the latest when peak paging loads occur when the average occupancy level is 70%.

Report “Page frames” of the report group MEMORY and “Main memory utilization” of the report group WORKING-SET

The value “Available pages (NPP)” specifies how much real main memory is still available for paging.

Paging operations start to occur if the value “Number of page frames for system global set” (SWS) reaches approx. 80 - 90% of NPP.

If the SWS value increases to above 90% of NPP, an expansion of the main memory should be taken into account. In VM2000 operation it is often (at least as a transitional measure) sufficient to distribute main memory more efficiently between the guest systems. With these high SWS values you are warned against adding new applications or changing the version of frequently used software. This could result in a significant rise in the paging rate.

Reports of the PERIODIC-TASK-... report groups

These reports are very useful for analyzing possible reasons for resource bottlenecks (in particular CPU and IO). Depending on the application it can make more sense to perform the analysis on the basis of the user ID, the job name or the TSN.

Reports of the VM2000 report group

S server

Measurements on the monitoring system can be used together with the “VM2000 utilization” report to analyze the CPU utilization of all the guest systems. Adding the value from the “VM2000 hypervisor” report enables the total utilization of the server to be obtained.

SQ servers

The total utilization of the server can be determined via the utilization of the various CPU pools. This information is provided in the “CPU pool utilization” report. The values for the CPU pool utilization supplied there relate to the CPU utilization of all CPUs in the pool concerned. Consequently they do not show the CPU utilization with respect to all available CPUs of the server. They can therefore not simply be added together, but must be “standardized” using the formula below:

```
Utilization [%] for CPU pool * Number of attached real CPUs for CPU pool
/ Total number active CPUs
```

The total utilization of the server is the sum of all standardized values for the available CPU pools.

When not only BS2000/OSD guest systems are available but also Linux and Windows guest systems, the utilization created by these guest systems must also be taken into account. The reports of the “Pool” and “Xen” report groups are required for this purpose. You can obtain these reports using the software product openSM2 (Open Systems).

Under VM2000 (depending on the number and load of the guest systems), the utilization of the BS2000 CPU can be 5-15% higher than in native operation.

10.3 Performing a bottleneck analysis

10.3.1 Basic procedure

Before discussing bottleneck analysis in detail we should recall the basic problems involved in meeting performance expectations:

Performance expectations of the individual user

The performance expectations of the individual user are temporal in nature (response time, elapsed or dwell time). The time taken to process the user's requests to the IT system has a direct effect on his/her work pattern and hence on his/her “productivity”.

Processing time comprises service time and the time spent waiting for the relevant resources. While service time reflects the immediate resource requirements, wait time depends on the workload placed on the resources. Therefore, the lighter the resource workload, the easier it is to satisfy the performance expectations of the user.

Performance expectations of the organization running the server

For economic reasons, the performance expectations of the organization running the server are directed towards attaining maximum “system throughput” (high transaction or job throughput rate).

Therefore, to meet the performance expectations of both the individual user and the systems support, it is always necessary to reach a compromise.

Understandably, investment planners also demand the fullest possible utilization of resources. However, in actual practice, various standard values have emerged for using resource capacity. These values should never be exceeded unless absolutely necessary. In most cases they do not represent technical limits. If exceeded, they lead to inefficient utilization of the IT system and make it more difficult to meet the performance expectations of the individual users.

Therefore, before carrying out measurements, it is essential to clarify which performance expectations are not being met:

- System-oriented performance expectations (the “system throughput” leaves something to be desired → the performance of the **entire IT system** is unsatisfactory).

  Indications include:

  – excessively low transaction rate (as an immediate offshoot of generally poor response time behavior)
  – excessively low throughput rate (dwell time for jobs generally too long).

- User-oriented performance expectations (the performance expectations of **individual users** are not being met).

  Indications include:

  – excessively long response time for certain types of transaction
  – excessive dwell time for particular jobs.

If the performance problem is **system-oriented**, the causes most likely lie in an overload on one or more resources.

If **user-oriented**, it is essential to find out the reasons why delays arise when handling special load requests.

As the use of additional monitoring programs leads to an increase in resource requirements (mainly CPU time), it should be ensured that the monitoring programs are disabled again when monitoring has been completed or removed from the start procedures.

10.3.1.1 Parameters for measurement and analysis

Normally, measurements for bottleneck analysis are made at the moment of peak workload. To facilitate analysis, as much measurement data as possible should be collected. The measurement period should be from 30 minutes up to a maximum of one hour.

An analysis interval of 15 minutes should be used as the basic reference value for judging the measurement values (average value for 3 measurement value subintervals of 5 minutes each).

The collection of measurement data by SM2 is partly event-related and partly done by sampling. To ensure the reliability of the samples, it is necessary to collect a certain number of measurement values per analysis subinterval (minimum 1500). A sampling period of 400 milliseconds is recommended.

For each monitoring cycle, SM2 forms the mean value of the collected measurement values. In order to illuminate the range of deviation per analysis subinterval, it is helpful if each analysis subinterval is composed of two or more monitoring cycles.

Recommended measurement parameters

Sampling period (SAMPLING-PERIOD)	400 milliseconds
Monitoring cycle (OFFLINE-PERIOD)	60 seconds
Analysis subinterval	5 minutes
Monitoring period	0.5 - 1 hour

With bottleneck analysis measurements, it is preferable to **output the observed values to the measurement file** rather than display them on the screen. Consequently file analysis with ANALYZER is preferred to the online analysis with INSPECTOR.

10.3.1.2 Measurement values for investigating system-oriented performance problems

For detecting resource overloads the following data is required:

- workload values: CPU, channels, devices (disks), main memory
  With disks, SM2 sees exclusively logical volumes. If the SHC-OSD software product is in use, the STORAGE-SYSTEM monitoring program supplies additional data about the physical drives of Symmetrix systems.
- number of I/O operations per device
- number of tasks per device queue
- working set requirements per task category.

To make clear the dependence of response time behavior on resource workload, it is helpful to maintain the following statistics in parallel:

- connection-specific response-time statistics (monitoring program BCAM-CONNECTION)

The use of the following monitoring programs is also recommended:

- Extended system statistics (monitoring program SYSTEM)
  Among other things, this monitoring program provides information on dilations and dwell times in the system queues for each category, as well as information on the hardware service time (hardware duration) and software service time (software duration) for DMS and paging I/O operations.
- Extended device statistics (monitoring program SAMPLING-DEVICE)
  The monitoring program is started automatically when SM2 is called, but the hardware and software service times for each volume are not recorded. If monitoring is also to include these values, the monitoring program must be restarted with the statements `//SET-SAMPLING-DEVICE-PARAMETER DISK-SERVICETIME=*ON` and `//CHANGE-MEASUREMENT-PROGRAM TYPE=*SAMPLING-DEVICE`.
- Channel statistics (monitoring program CHANNEL-IO)
  This monitoring program is used to ascertain the number of I/O operations and the number of PAM blocks for each channel.
- Catalog statistics (monitoring program CMS)
  This monitoring program provides information on the number, type and duration of the accesses to catalog entries on each pubset.
- VM2000 statistics (monitoring program VM)
  In VM2000 operation this monitoring program should only be started in the monitoring system (but there by default). In the monitoring system it supplies the CPU share of all guest systems and on S servers the share of hypervisor active and idle.

10.3.1.3 Measurement values for investigating user-oriented performance problems

Besides the above-mentioned global system workload values, additional data is required for detecting delays. To begin with, the corresponding task selection can be carried out using the periodic task statistics (monitoring program PERIODIC-TASK).

The following monitored variables are necessary for further analysis:

- resource depletion per task (TASK monitoring program)
  The monitoring program provides:
  – the CPU time requirements
  – the number of SVC calls
  – the number of I/O operations and I/O times, broken down according to the volumes defined in the monitoring program
  – the duration of wait states
  – Paging activities
  – the message lengths per terminal I/O.

  An additional user task measurement with subsequent analysis by SM2-PA (program analyzer) is recommended for analyzing the CPU requirements of individual tasks in more detail using program counter statistics and SVC statistics.
- SVC statistics (SVC monitoring program)
  The monitoring program provides the frequency of each SVC call classified by processor states TU and TPR.
- DAB hit rate (DAB monitoring program)
  The monitoring program provides the number of accesses per specified DAB subarea.
- wait times in front of individual devices (SERVICETIME monitoring program)
- ISAM pool hit rate (when NK-ISAM is used; ISAM monitoring program)
  The monitoring program provides the number of accesses per specified SAM pool.
- number of access operations to particular files (FILE monitoring program)
- UTM statistics (UTM monitoring program)
  The monitoring outputs data for each application, including the dwell time in UTM, the dwell time broken down according to database share, TAC-class wait time and time for distributed processing. In addition, the program provides average consumption values for CPU and I/O operations in UTM and in the database system (only SESAM/SQL and UDS), as well as the number of database calls for each dialog step or asynchronous process.

UTM values are output under the following conditions:

- the UTM-SM2 subsystem must have been started
- UTM-SM2 is started automatically when the operand `MAX SM2=ON` is specified for the KDCDEF run.
  UTM-SM2 is started subsequently via the UTM administration interface with `KDCAPPL` and `SM2=ON`, together with the readiness to pass values.
- the SM2 monitoring program UTM must have been activated
- the UTM application must have been set to supply data to SM2.
  The passing of values is controlled by means of the `MAX SM2=ON/OFF/NO` operand in the KDCDEF run as follows:
  - if `SM2=OFF` (default value) is set, the passing of values for each application can be activated at a later point in time via the UTM administration interface with `KDCAPPL` and `SM2=ON`;
  - if `SM2=ON` is set, no further administration is required;
  - if `SM2=NO` is set, the passing of values is generally prevented and cannot be activated at a later point in time.
- database-specific consumption values (I/O operations, CPU time) are output only under the following conditions:
  - BS2000 accounting is activated;
  - the UTM accounting record UTMA is activated (with `/MODIFY-ACCOUNTING-PARAMETERS, ADD-RECORD-TYPE=UTMA`);
  - UTM accounting is activated (`KDCAPPL`, parameter `ACC=ON`);
  - in SESAM/SQL, the recording of statistics is activated with the statement `ACC,TP=ON,CPU`.

Further information is provided in the manual “openUTM - Concepts and Functions” manual [19] and in the manual “Using openUTM Applications under BS2000” [22].

- Database statistics
  The SESAM-SQL and UDS-SQL monitoring programs supply monitored data about the SESAM/SQL and UDS/SQL database systems.

  Prerequisite in SESAM/SQL:

  To transfer statistical data from SESAM/SQL to openSM2, start SESMON in batch mode:

```
/START-SESAM-PERF-MONITOR
//SET-MONITOR-OPTIONS ...,OUTPUT=*SM2
```

  When `OUTPUT=*SM2`, only one DBH can be specified for each SESMON instance. A new SESMON instance must be started for each further DBH for which data is to be output.

The interval at which SESMON transfers the data to openSM2 is automatically set to approx. 30% of the SM2 monitoring cycle. It cannot be set manually.

Prerequisite in UDS/SQL:

Transfer of the statistical data from UDS/SQL toopenSM2 is initiated either when the UDS monitor is started using the `MEDIUM=S,n` statement or, during ongoing operation, with the `/INFORM-PROGRAM MSG='ADD MEDIUM=S'n'` command. It can be terminated again using the `/INFORM-PROGRAM MSG='FINISH MEDIUM=S'` command.

`n` is used to define the cycle in seconds ($5 \leq n \leq 999$) in which the UDS monitor transfers the data to SM2. It should be set to considerably less than the monitoring cycle set in SM2 so that data can be transferred several times in an SM2 monitoring cycle.

The monitored values are supplied asynchronously to openSM2 from the database systems and apply for cycles defined for one or more database systems which do not need to match the SM2 cycle exactly. Here there can be differences in both the length of the cycles and in the temporal displacements between the database system and the SM2 cycles.

The length of the database system cycle or cycles is used to standardize the monitored values to one second. The values are therefore exact, but they only match the SM2 cycle to a certain degree.

- SYMMETRIX statistics (STORAGE-SYSTEM monitoring program)
  When the SHC-OSD software product is in use, the hit rates of Symmetrix systems are recorded. Data is supplied about logical drives as well as information about the physical drives.

  You require the software product openSM2 (Open Systems) to monitor ETERNUS DX disk storage systems.

- Information on TCP/IP connections (TCP-IP monitoring program)
  The IP and PORT number is supplied for each connection together with the number of transport service data units transferred.
- Number of accesses and data transferred of network devices
  (SAMPLING-DEVICE monitoring program)
- Communication in the computer network (MSCF monitoring program)
  Data is supplied on the communication of the local computer with other computers.
- Data on basic functions in the HIPLEX network (NSM monitoring program)
- Statistics on lock requests from TU, TPR and NSM
  (DLM monitoring program)

- POSIX statistics (POSIX monitoring program)
  Supplies measurements on the use of various POSIX functions.
- Number of accesses of global storage (GS monitoring program)
  Read and write accesses are displayed for each GS partition.
- Detailed disk statistics (DISK monitoring program)
  In the event of the almost exclusive use of disk storage systems with cache, this monitoring program loses its original significance. The reason is the decoupling of the logical volume from the physical device (see the section “Configuration in BS2000/OSD” on page 52).

10.3.1.4 Evaluation of an SM2 measurement

When using ANALYZER (or SM2R1) for analysis purposes, it is best to proceed step by step:

1. Rough analysis

   Bottleneck search by means of pinpointing (analysis subinterval = 15 minutes)

   This locates those analysis subintervals in which the recommended standard values are exceeded.
   The **automatic performance analysis** available in ANALYZER (or SM2R1) provides valuable help in tracking down bottlenecks. This function compares the measurement values collected in the SM2 file with the standard values recommended in the following section, and issues a corresponding message to the user. It is important that only critical periods of productive operation are used for automatic performance analysis.

2. Detailed analysis

   Investigate the critical analysis subintervals (15 minutes) by reducing the analysis subinterval (TIME-STEPS) to the size of the monitoring cycle (OFFLINE-PERIOD) in order to highlight the range of deviation in the measurement values.

To facilitate the assessment of the measurement findings, it should be ensured that the measurement values come from periods in which performance expectations were satisfied as well as from periods in which they were not satisfied. Finding the cause of the problem is much easier when both cases are compared.

The following section attempts to simplify the interpretation of measurement values by suggesting standard values for resource workloads.

10.3.2 Investigating system-oriented performance problems

First, for ease in understanding the suggested standard values for resource workloads, it is helpful to quantify the work of BS2000.

System time (SIH time) is required primarily for handling the following events:

- controlling multiprogramming operation
- carrying out DMS I/O operations
- carrying out paging.

The following standard values apply to a CPU working at full capacity:

The SIH portion should not be more than 20% (S servers) or 10% (SQ servers), even in the case of heavy DMS I/O activity.

With normal loads, the time taken up for controlling multiprogramming operation amounts to 5 - 10% (S servers) or 3 - 5% (SQ servers) of the full capacity of the CPU involved.

With the usual workload and a balanced configuration, the following SIH portions apply for

- executing DMS I/O operations: 3 - 6% (S servers) or 2 - 4% (SQ servers)
- executing paging: max. 2% (S servers) or max. 1% (SQ servers)

When interpreting the values measured with SM2, the user is advised to proceed in the order of the sections which follow.

10.3.2.1 I/O peripherals workload

Standard values for the DMS I/O rate

In section “Standard values for BS2000/OSD servers” on page 331 you will find guidelines for the recommended maximum I/O rate as a function of the CPU type. These values are governed by the rule that no more than a certain share of CPU capacity (for S servers 15 - 25% or for SQ servers 12 - 16% in processor state SIH+TPR) should be used for initiating and handling I/O operations. In individual cases, the recommended values may be considerably exceeded. This can by all means be a meaningful application. In such cases an individual check should be made to determine whether the violation of the recommended values causes performance problems.

ANALYZER reports for assessing the DMS I/O rate

Report group	Report	Meaning
IO	IOs for device classes	Number of DMS I/O operations/s
CATEGORY-IO	IOs for category	Number of DMS I/O operations/s per task category
STORAGE-SYSTEM-SYMMETRIX	Reads	Number of read accesses/s per Symmetrix system
	Writes	Number of write accesses/s per Symmetrix system
	Reads + writes	Total number of accesses/s per Symmetrix system
	Sequential reads	Number of sequential read accesses/s per Symmetrix system

With SHC-OSD the data for Symmetrix systems is recorded with the STORAGE-SYSTEM monitoring program (reports of the STORAGE-SYSTEM-SYMMETRIX report group).

You require the software product openSM2 (Open Systems) to monitor ETERNUS DX disk storage systems.

Tuning measures for reducing the DMS I/O rate
(not sorted according to general importance!)

- Reorganize the ISAM files
  Lower the number of index levels by means of organizational measures (dividing the collected data, checking for data obsolescence).

- Switch to NK-ISAM files, with the resultant savings on I/O operations through the use of self-configuring ISAM pools or ISAM pools which can be selected on a user-specific basis.
  Use of the ISAM pool can be monitored with the aid of reports of the report group ISAM-POOL or ISAM-FILE:

Report	**Monitoring variable**	**Meaning**
Fix operations	Number of fix-hit operations	number of ISAM accesses that have been resolved from the ISAM pool
	Number of fix-IO operations	number of ISAM accesses per second that resulted in an I/O operation to disk (read access)
	Number of fix-wait operations	number of ISAM accesses per second during which the system had to wait for pool pages to become free
Slot operations	Number of reserve slot wait operations	number of wait states per second resulting from bottlenecks in the internal pool management which were caused by the pool being too small
Used pages	Number of total pages	number of PAM blocks in the ISAM pool which (after deducting the space required for management data) are available for buffering data
Index operations	Number index operations	total number of index accesses
	Number index hit operations	share of index accesses dealt with from the ISAM pool

The ISAM pool is the correct size (for details of calculating the size, see the “DMS Macros” manual [8]) when the following applies:

$$\frac{\#\text{FIX-HIT OPERATIONS}}{\#\text{FIX-HIT OPERATIONS} + \#\text{FIX-IO OPERATIONS}} \geq 0{,}85$$

The following must also be true:

```
Number of fix-wait operations = 0
Number of reserve slot wait operations = 0
```

In order to generate an ISAM pool that is of sufficient size, enough main memory capacity must be available to prevent the effect of reducing the number of DMS I/O operations being compensated for by an increase in the number of paging I/Os.

- Raise the blocking factor (for both disk and tape processing)
  Increasing the block size is always advisable in the event of sequential access. In the case of random-access processing (PAM, ISAM), this condition must be checked on an individual basis. Increasing the block size reduces the number of I/O operations and thus offloads the device and the CPU, but not the channel.
- Increase the size of the internal data buffer for database applications (please refer to the relevant database manuals)
  Whenever an increase in the size of the internal buffers leads to a corresponding increase in the size of the user address space, adequate main memory capacity must also be provided.
- Reduce the I/O rate to KDCFILE in UTM applications by increasing the size of the UTM cache area (see the “openUTM Administering Applications” manual [20]).
- Use the software product DAB (ADM-PFA)
  By reading in up to 16 PAM blocks at a time (32 PAM blocks with AutoDAB), the number of I/O operations required is significantly reduced when processing is mainly sequential. Prerequisites are sufficient main memory and a corresponding CPU reserve capacity (CPU intensity increases when the physical I/O delay is reduced). See the “DAB ” manual [5] for information on using this function.
  The hit rate on the specified DAB subareas can be checked with the help of the reports “Reads for internal area” und “Writes for internal area” of the report group DAB. If AutoDAB is used, `/SHOW-DAB-CACHING` provides an overview of the files which are currently subject to caching. See also the section “ADM PFA Caching” on page 180.
- Use HIPERFILEs utilizing the cache media MM and GS (see section “Working with HIPERFILEs” on page 176).

If the common value for the DMS I/O rate is less than the suggested standard value (see the section “Standard values for BS2000/OSD servers” on page 331), the next step is to check the workload on the channels and the attached peripheral devices.

Standard values for the channel workload

In the case of channels with ETERNUS DX or Symmetrix disk storage systems connected, the workload should not exceed 60%. This applies for both native operation and operation under VM2000 (see also information on the “Report “Utilization” of the report group CHANNEL” on page 276).

Reports of the report group CHANNEL for assessing the channel workload

Report	Meaning
Utilization	Channel workload (channel busy state)
Data transfer	Number of PAM blocks transferred
IOs	Number of I/O operations per channel

Tuning measure for channel overload: reconfiguration

With Symmetrix systems, all channels that belong to a **single** Symmetrix box are subjected to a relatively evenly distributed load. Temporary overloads (15 minutes) for application-related reasons can be tolerated.
More channels have to be made available for overloads that last any longer. If type S channels are in use, it is recommended to exchange them with type FC channels, see page 35.

Standard values for the device workload

Disks (more precisely: logical volumes)

If volumes are shared by two or more users, the size of the workload is a key criterion for the wait time in front of each volume. With online applications, the wait time should not be more than one third of the hardware service time.

Therefore it is essential to determine whether the volume concerned is being shared by two or more users (or tasks) or whether it is assigned permanently to one user.

For volumes shared by two or more users (or tasks) (public volumes, private volumes with e.g. one UTM application encompassing several tasks, shared private volumes, shared public volumes), the workload should not exceed 30% (based on 15 minutes).

When the PAV function is used, several device addresses (alias devices) are available for each volume (base device) (see also page 55). I/O jobs can be processed via simultaneously via these device addresses. A higher workload for the base device (not visible because SM2 obtains the average value for the base device and alias devices) therefore does not lead to a longer waiting time in front of the volume. The volume workload visible from the outside (≙ the average value for the base device and alias devices) should not exceed 30%.

For volumes assigned to a single user (or task) a workload of 100% is feasible.

Magnetic tape cartridge units

Usually magnetic tape cartridge units are permanently assigned to a single task. Therefore, a workload of 100% is feasible.

The user should make sure here that the workload is not made artificially heavy by the “processing” of an excessive number of inter-block gaps.

ANALYZER reports for assessing device workload (volumes)

Report group	Report	Meaning
CATEGORY-IO	Duration of non paging IOs for category	Hardware and software service time for DMS I/O operations per task category
	Duration of paging IOs for category	Hardware and software service time for paging I/O operations per task category
DISK	Utilization	Load per volume; In each case, the load is broken down into the amounts for DMS and paging I/O operations
	IOs	Number of I/O operations per second and volume
	Queue length	Average length of queues for the individual volumes
	Data per IO	Number of PAM blocks per I/O operation and volume
	Time	Hardware and software service time for DMS I/O operations per volume
	Number alias devices	Number of alias devices per base device
SERVICETIME	Duration of IOs for device	Hardware service time for I/O operations per volume with a detailed breakdown and the waiting time in front of the volume

Report group	Report	Meaning
STORAGE-SYSTEM-SYMMETRIX	Pubset reads	Number of read accesses/s and hit rate per Symmetrix pubset
	Pubset writes	Number of write accesses/s and hit rate per Symmetrix pubset
	Pubset reads + writes	Total number of accesses/s and hit rate per Symmetrix pubset
	Pubset sequential reads	Number of sequential read accesses/s per Symmetrix pubset
	Device reads	Number of read accesses/s and hit rate per Symmetrix device
	Device writes	Number of write accesses/s and hit rate per Symmetrix device
	Device reads + writes	Total number of accesses/s and hit rate per Symmetrix device
	Device sequential reads	Number of sequential read accesses/s and hit rate per Symmetrix device
	Device sequential writes	Number of sequential write accesses/s and hit rate per Symmetrix device
	Physical disk IOs	Total number of accesses/s and write accesses/s per physical disk device
	Director IOs	Total number of accesses/s and hit rate per channel director

The software product SHC-OSD is required to record data from Symmetrix systems with the STORAGE-SYSTEM monitoring program.
The software product openSM2 (Open Systems) is required to record data from ETERNUS DX systems.

Tuning measures for device overloads (volumes)

The first thing to do is always to check the hardware service time (report “Time” of the report group DISK with the SAMPLING-DEVICE monitoring program switched on), taking into account the number of transferred PAM blocks per I/O operation.

In very rare cases the analysis must be continued as follows:

a) If the hardware service time is only negligibly longer than the DEVICE CONNECT TIME (report “Duration of IOs for device” of the SERVICETIME monitoring program or report group), you are dealing with read hits or fast write hits.

   The overloading of the logical volume is caused by the frequency of the accesses and leads to wait times for the tasks in front of the volume. In this case, it makes sense to use the PAV function. If you do not, it is necessary to relocate particularly frequently used files to other, less busy volumes (if possible also on a different physical disk drive).

b) If the hardware service time contains a significant share of DEVICE DISCONNECT TIME (report “Duration of IOs for device” of the report group SERVICETIME), the cause may be one of the following:

   – Insufficient support for the read caching due to too small a cache (read-hit rate too low, report “Reads for device” of the report group SYMMETRIX-DEVICE or report “Device reads” of the report group STORAGE-SYSTEM-SYMMETRIX).

   – If there is a lack of “breathers” during continuous write loads, cache contents cannot be saved to disk in the interim (large proportion of delayed fast writes, recognizable by a write hit rate < 100%, report “Writes for device” of the report group SYMMETRIX-DEVICE or Report “Device writes” of the report group STORAGE-SYSTEM-SYMMETRIX).

   – A competition situation between logical volumes on the same disk drive (check the volumes per disk drive with the command `/SHOW-SYMMETRIX-DEVICE-CONFIG UNIT=*BY-VOLUME (...), INF=*PHYSICAL` of the SHC-OSD software product; can be controlled with report “Physical Disk IOs” of the report group STORAGE-SYSTEM-SYMMETRIX).

c) A share of REMAINING SERVICE TIME (report “Duration of IOs for device” of the report group SERVICETIME) in the hardware service time indicates the following:

- Dilation of the hardware service time by other guest systems under VM2000.
  In this case, you should check the setting for the CPU quotas.
- Use of REC / SRDF (see page 50).
  In case of FBA format the share of REMAINING SERVICE TIME is similar in size to the DEVICE CONNECT TIME. If it is considerably higher, the remote data links are not powerful enough, i.e. they must be increased in number.
  With FBA formatting, the DEVICE CONNECT TIME is higher than the time required to transfer the data (this is a property of the FBA protocol: the disconnect comes later than with the CKD format). A REMAINING SERVICE TIME part, that is the same size as the DEVICE CONNECT TIME, indicates a number of “Remote Data Links” that is too low.

10.3.2.2 Paging activities

Main memory management is characterized by the paging mechanism, which allows the efficient use of a relatively small main memory in relation to the virtual address space requirements.

Because paging both causes time delays and adds to the workload on the processor and the disk peripherals, the paging rates specified in section “Standard values for BS2000/OSD servers” on page 331 should not be exceeded.

You should also make sure that a sufficient number of volumes is available for performing the paging I/O operations. Around 30 paging I/O operations per second can be implemented for each volume (given a workload of 10% and a read hit rate of 70%).

Clues as to the cause of the paging rate are provided by the ratio of the memory used by the system (SWS) to the number of main memory pages available for paging (NPP).

See also the notes on “Report “Page frames” of the report group MEMORY and “Main memory utilization” of the report group WORKING-SET” on page 280.

The tables in the section “Standard values for BS2000/OSD servers” on page 331 contain representative main memory configurations for various servers. The actual main memory required depends on the workload. TP and interactive loads usually have a considerably higher main memory requirement than batch loads.

10.3.2.3 CPU workload

As a rule, the resource CPU is used simultaneously by many tasks. Here, too, the size of the workload is a key criterion for the amount of time a task has to wait for the CPU.

The average hardware service time of the resource CPU per user request (e.g. between two DMS I/O operations) is relatively small as compared to the execution of I/O operations.

Consequently, the wait time in front of the resource CPU has considerably less effect on response or dwell time than the wait time in front of disk drives (volumes). Therefore, a heavier workload is feasible as compared to shareable volumes. However, for the **main application** a workload of **70%** should not be exceeded (with multiprocessor systems - depending on the number of CPUs - a maximum workload of 90% can be tolerated). When multiple guest systems are used, generally a 5 - 15% higher overall workload than in native operation is possible.

In addition, the task scheduler PRIOR allows tasks to be given precedence by means of priority assignment. This makes it possible to use part of the remaining **30%** (or 10%) by running **low-priority tasks** (with correspondingly long wait periods in front of the resource CPU). Utilization of 100% of the CPU without adversely affecting the main application will only be possible in rare instances.

ANALYZER reports for assessing the CPU workload

Report group	Report	Meaning
CATEGORY-CPU	CPU utilization (TU+TPR) for category	CPU workload in the TU and TPR processor states for each task category
CPU	Utilization real	CPU workload in the TU, TPR and SHI processor states (actual values). When VM2000 is being used, this report is important for determining the CPU share of the guest system vis-à-vis the overall system
	Active logical machines	Number of active logical machines (number of CPUs used by BS2000/OSD).
	Sum SVC calls	Sum of all SVC calls in the TU and TPR processor states

Heavy CPU workload

CPU workloads are considered heavy when they exceed 90%.

A heavy CPU workload does not necessarily mean that the resource CPU is the source of the bottleneck.

If you have determined that the rate of CPU utilization is high (the number of CPUs used by BS2000 must match the number of actually available CPUs), you must first check whether the CPU resources are being used efficiently. You can do this by studying the relationship between the workload in processor states TU and TPR and the workload in processor state SIH (see the notes on report “Utilization normed” on page 277).

Try to achieve the following ratio:
`TU + TPR > 3 * SIH` (S server) or `TU + TPR > 7 * SIH` (SQ server)

If the ratio is poor, the investigation should be conducted with the objective of reducing the SIH time, which is essentially nonproductive for users:

- Checking the level of the DMS I/O rate (see section “I/O peripherals workload” on page 292).
- Checking the level of the paging rate (see section “Paging activities” on page 299).
- All calls to BS2000 are handled via SVCs. The processing of SVC interrupts involves corresponding overhead in the system, which is charged to the account of the calling task.

If the number of SVC calls per second shown in the tables in the section “Standard values for BS2000/OSD servers” on page 331 is reached, the overhead required for the interrupt analysis and termination handling of the system routine (i.e. not the resources used for the system routine itself) amount to approx. 5% of CPU time.

The total number of SVCs can be determined via report “Sum SVC calls”. The TASK monitoring program of SM2 can be used to count the number of SVCs for each task. Further classification in the form of SVC statistics is possible by means of user task measurement for selected tasks followed by SM2-PA analysis.

The following relationship helps to assess whether the resource CPU is forming a bottleneck in conjunction with heavy CPU workload **and** efficient utilization:
`Wait time for using the CPU / hardware service time of the CPU`

Standard value

The resource CPU can be regarded as overloaded when the main application yields the following ratio or the background application yields a correspondingly higher ratio.

```
Wait time for using the CPU / hardware service time of the CPU > 3
```

If long wait times for use of the CPU occur for the main application, there could be one of two reasons:

- the system control parameters (in particular the priority assignment) were not set in accordance with the actual requirements;
- the total CPU time requirement of all tasks is too high (critical case).

Further information is provided by the dwell time percentages in the system queues of the SM2 monitoring program SYSTEM (evaluation with the SM2R1 statement `//PRINT-QUEUE-TRANSITION`).

Example 1

CPU workload:	96 %
TU portion:	36 %
TPR portion:	45 %
SIH portion:	15 %

Main application: TP task category

```
Dwell time in 01 (%) / Dwell time CPU (%) = 5.4% / 1.2% = 4.5
```

background application: Batch task category

```
Dwell time in 01 (%) / Dwell time CPU (%) = 2.1% / 1.4% = 1.5
```

By choosing a poor setting for the system control parameters, the background application BATCH does not run in the background as intended. This forces the tasks in the main application TP to wait too long for use of the CPU.

Remedy: Assign a higher priority to main application TP or use PCS (see section “Introduction to PCS concept” on page 222).

The CPU workload **cannot** be deduced from the dwell time percentages in the system queues. (The sum of the dwell times in all system queues is 100%.)

Example 2

CPU workload: 96 %
TU portion: 36 %
TPR portion: 45 %
SIH portion: 15 %

Main application: TP task category

```
Dwell time in O1 (%) / Dwell time CPU (%) = 5.4% / 1.2% = 4.5
```

background application: Batch task category

```
Dwell time in O1 (%) / Dwell time CPU (%) = 16.8% / 1.4% = 12
```

The total CPU time required by all tasks is too high. The CPU resource is overloaded.

If the CPU workload is heavy and the performance demands of the individual users are not being met despite efficient utilization and short wait times, the resource requirements per user request (e.g. CPU time required per transaction) are too high (see section “Investigating user-oriented performance problems” on page 306).

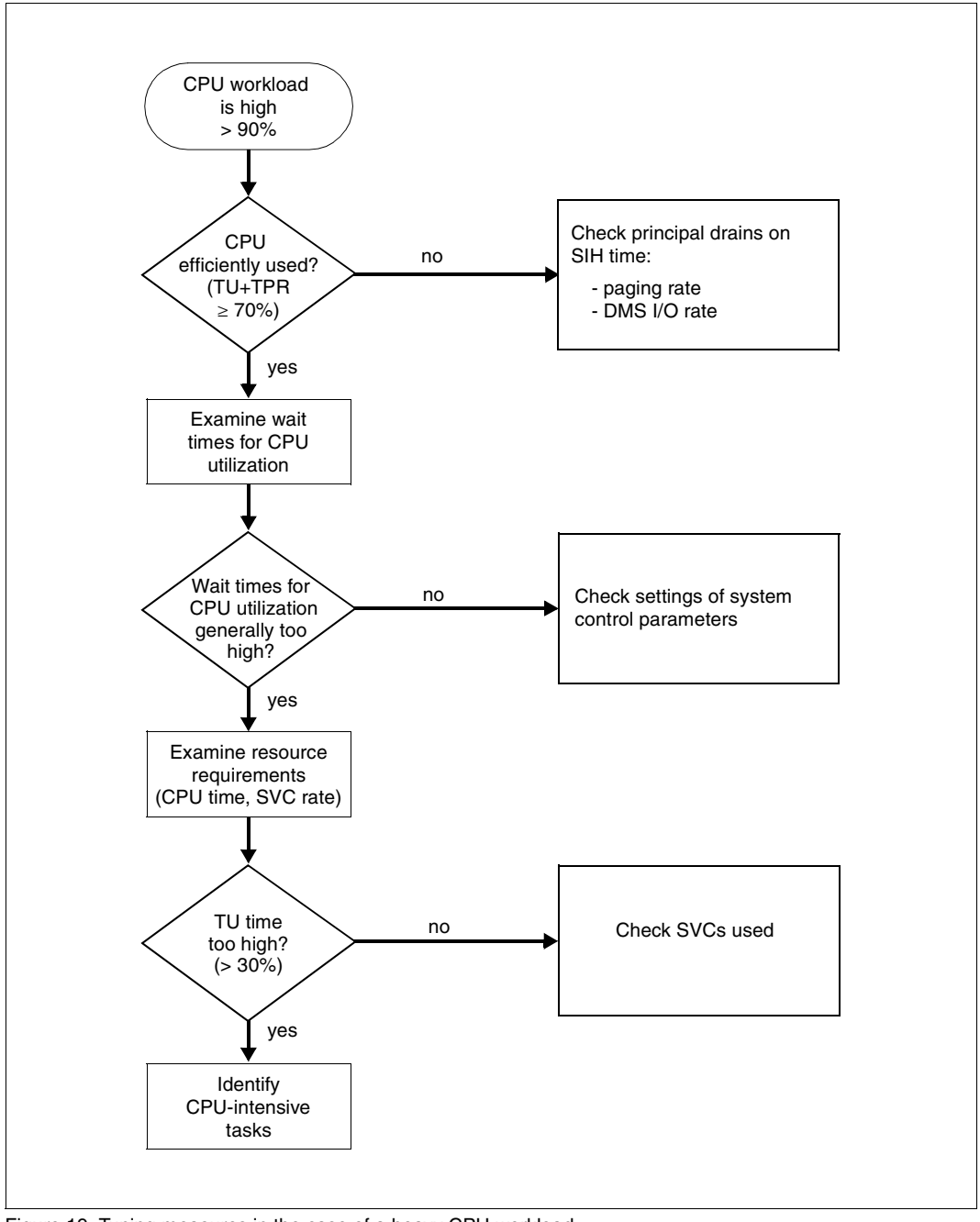


Figure 19: Tuning measures in the case of a heavy CPU workload

Low CPU workload

If performance problems arise without a correspondingly heavy CPU workload (<90%), one of the following causes may be involved:

- Conflict situation on the DMS I/O side
  (see section “I/O peripherals workload” on page 292)

  Overloading of volumes containing files which are required equally by a large number of users (e.g. particular user control files or the system file TSOSCAT or SYSEAM) has a particularly unpleasant effect.

  The SM2 monitoring program FILE can be used to calculate the number of I/O operations on these files.

- Paging rate too high (see section “Paging activities” on page 299)

  In the case of severe main memory shortage (i.e. when the paging rate far exceeds the standard values given), bottlenecks result on the public volumes with regard to the number of paging I/O operations to be executed per second. This leads to a forced CPU wait state (IDLE).

- Delays caused by PASS/VPASS calls

  If the SVCs PASS/VPASS are called too often, tasks are placed in a wait state too frequently and cannot utilize the CPU.

- Bottlenecks caused by server tasks

  Some applications have large portions of their workload handled by specialized server tasks. If the progress of these tasks is impeded by paging, deadlocks, resource overloads, unfavorable priorities, etc., this can detrimentally affect the entire application. The server task itself acts as a resource.

- Insufficient parallel processing because number of tasks too small

  Not only do the service times of the relevant hardware resources affect performance, but also the amount of time during which a task is occupied with CPU and I/O activity in order to perform a desired user function.

Example

If the average processing time per transaction is 0.5 seconds (0.2 seconds CPU time + 0.3 seconds I/O time), it is possible to obtain the following maximum transaction rate (transactions/s) for a task (not including the wait time for the relevant hardware resources):
`1 / 0.5 sec. per transaction = 2 transactions per second`

This yields a CPU workload of `0.2 * 2 = 0.4` or 40%.
A higher transaction rate or more favorable utilization of CPU capacity can only be obtained by raising the number of tasks (for details, see section “Task occupancy time” on page 307).

10.3.3 Investigating user-oriented performance problems

It is considerably more difficult to locate the causes of user-oriented performance problems than to investigate system-oriented performance problems (using SM2).

If response or dwell times are unsatisfactory for certain users, it is necessary to interpret the task-specific measurement values as well as to assess the global workload values.

In addition, SM2 user task measurement should be performed:

`/START-TASK-MEASUREMENT` allows users or systems support to specify tasks which they want to have monitored.
Besides task-specific measured values, program counter statistics and SVC statistics can also be requested. The measured values are written to a user-specific file and analyzed by the software product SM2-PA (see the “SM2-PA” manual [29]).

10.3.3.1 Task occupancy time

The task occupancy time consists of the CPU time and the I/O time of a desired user function plus the wait time for the corresponding hardware resources due to their workload.

The longer the task occupancy time, the smaller the number of user requests which this task can execute per unit of time.

Example

Task occupancy time for an average transaction in TP mode:

Hardware service time of CPU	200 ms
Wait time for CPU (with 60% CPU workload, assuming “M/M/1”)	300 ms
Hardware service time of all I/O operations	400 ms
Waiting for I/O (with 30% workload, assuming “M/M/1”)	170 ms
	1070 ms

If this application requires a higher transaction rate than `1 / 1.07 sec. per transaction = 0.93 transactions per sec.` at peak load time and if only **one** task is available, that task will create a bottleneck.

The resultant wait periods in front of the task (incoming messages accumulate in the BCAM buffer) can be many times longer than the task occupancy time (see also report “Inwait time distribution” of the report group BCAM-CONNECTION on page 276).

The SM2 monitoring program TASK provides information on the task occupancy time.

This task occupancy time can be estimated from the duration entries for the various wait states and the CPU time used. Difficulties arise if it is unsure whether certain wait states are voluntary or inadvertent, i.e. whether or not a task requested the service which is provided for it itself.

The following ratio indicates that the task occupancy time is too long (D = Duration):

```
(CPU time + CPU-WAIT(D) + DISK-IO-WAITS(D) + NON-DISK-IO-WAITS(D)) /
/ dwell time > 0,7
```

The `dwell time` **does not** include the wait time for other tasks (it may consequently be necessary to add this).
The computed CPU time refers to the TU and TPR processor states; the times for I/O activity refer solely to software service time.

10.3.3.2 Blockages caused by other tasks

Conflict situations in connection with access to shared hardware resources (essentially public/private volumes)

A SOFTWARE DURATION value (volume occupied) more than 10% higher than the corresponding value for HARDWARE DURATION is a clear indication that other tasks are impeding processing (exception: asynchronous input/output, see the explanation of the “”Time” report of the DISK report group” on page 278).

In this case, a task must wait for the volume before initiating the I/O operation, because the it is busy. The SM2 monitoring program TASK (with `INFO=HIGH`) indicates which other tasks are likewise occupying this volume with their I/O operations.

The problem can be solved by using the PAV function or file relocation.

Conflict situations in connection with updating shared data

Conflict situations can arise during the updating of shared data (e.g. tables) whose contents must be kept consistent for purposes of job processing

Conflicts of this sort can be regarded as forced serialization. They can be identified in the task-specific characteristic data by a relatively high proportion of

- “active waits” when serializing via the bourse mechanism or issuing `VPASS 0` wait calls
- “inactive waits” when the `PASS` wait call is used.

There are, however, many instances in which SM2 can make no distinction whatsoever between such a state and a normal condition. User systems (e.g. UTM, UDS, SESAM/SQL, ORACLE) now offer their own monitors for use in pinpointing such conflicts.

Problems in connection with multi-level task concepts (server tasks or handler tasks)

Functions shared by all tasks are very frequently isolated and concentrated into a **single** task (e.g. I/O operations for special data sets, database calls, etc.).

If the task occupancy time for this central task is too long, wait states will result for the tasks which draw upon it.

A bottleneck ahead of a UTM application as a result of too small a number of (UTM) server tasks can be measured directly (see the explanations in section “Transaction mode and BCAM” on page 188).

Often wait states can be detected by the increasing number or duration of “active waits” computed by the SM2 monitoring program TASK for the calling tasks. In the case of TP tasks, the number of “active waits” also contains not only those “waiting for input/output” but also those “waiting for terminal input”. This portion is negligible when the central task has a long task occupancy time.

The server task must be programmed so as not to require any locks, which it might then have to wait out. If the server task is not given priority over other tasks, it likewise runs the risk of encountering sudden losses in throughput. Due to the way jobs are handled between the job source task and the server task, congestion may occur in the event of programming which is not error-free, causing the server task to come to a halt. All of these problems are indications that the especially high quality of the coding required for server tasks is not being attained.

Example (simplified)

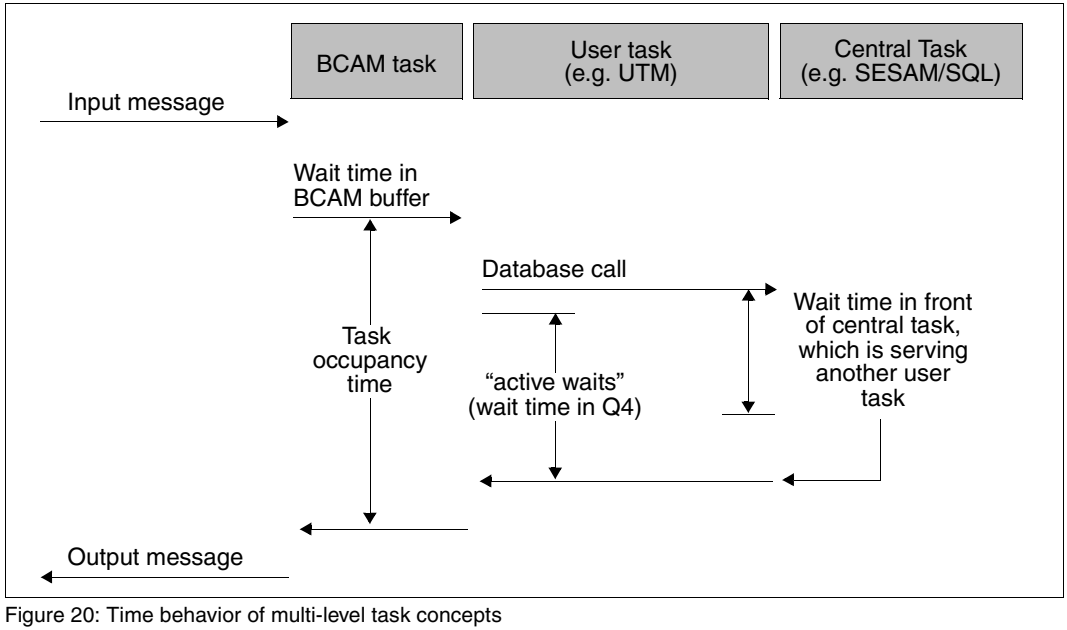


Figure 20: Time behavior of multi-level task concepts

10.3.3.3 High resource requirements

If the performance expectations of the user are not being satisfied despite the fact that resources are being utilized efficiently and have short wait times, this indicates that the resource requirements per user request are too high (e.g. CPU time requirement and/or number of I/O operations per transaction).

Indications regarding resource requirements can be found in the figures supplied by the SM2 routine TASK on

- CPU time consumption
- number of SVC calls in TU
- number of DMS I/O operations

For a more detailed investigation of application program behavior, SM2 user task measurement with an SM2-PA analysis is required:

- Program counter statistics enable those program areas to be pinpointed which are frequently executed.
- The SVC statistics contain all the SVC numbers called and their call addresses.

10.3.4 Influence of the network

Previous recommendations have all concentrated on improving the efficiency of the CPU. As described in chapter “Performance expectations from the user viewpoint” on page 15, good transaction times are essential to productivity. Measures for optimizing response times in transaction mode and with BCAM are described in section “Transaction mode and BCAM” on page 188.
As already shown in figure 1 on page 17, network runtimes are included as part of the transaction times. These runtimes may be greater than the actual response time in the host. For this reason it must be ascertained whether or not unsatisfactory transaction times are attributable to the network.
If the reports of the report group BCAM-CONNECTION indicate that response times are acceptable yet predominantly long transaction times are observed at the terminals, this means there is a bottleneck in the network.

A network is a complex entity with many interdependencies. In view of the great variety of possible network structures, only the most important characteristics and general conditions are indicated here. Describing how to deal with such problems comprehensively is beyond the scope of this manual.

Load requirements

Load type	Requirements
TP or interactive mode	Short response time
Data backup, file transfer	Throughput
Internet, e-mail	Throughput, response time

Possible bottlenecks

- Performance of network connection (e.g. older-type HNC)
- Client system (e.g. hardware performance, structure of the application, networking parameters)
- Network configuration with conflicting goals with regard to response time / throughput (e.g. parallel operation of transaction-oriented and throughput-oriented applications)
- Network structure (e.g. configuration of network nodes, routers, switches, hubs)
- Network overload by third party (e.g. the Internet)

10.4 Capacity requirement of SM2

To record the monitored data SM2 requires the resources CPU and disk storage. The requirement depends on:

- the type of monitoring programs and their parameters
- the hardware configuration
- the load profile (principally the number of tasks and the I/O intensity)

In all the resource utilization by SM2 is relatively low. Occasionally a check should be made to see whether a restriction of the monitoring programs is necessary or possible. However, a restriction of the monitoring programs is detrimental to the quality of performance monitoring.

Monitored data

The resource requirement of SM2 was measured in a laboratory test under a typical OLTP load which utilizes approx. 70% of the server capacity.

The following CPU requirement for SM2 in absolute percentage values was measured:

- 0.5 - 1% for basic monitoring (standard and SYSTEM monitoring programs)
- Additional 1% for recording response times and the UTM monitoring program
- Additional 1% for utilization of the PERIODIC-TASK and TASK monitoring programs
- Additional 1% for utilization of the STORAGE-SYSTEM monitoring program (with `//SET-STORAGE-SYSTEM-PARAMETERS ADDITIONAL-DATA=*SYMMETRIX(TYPE=(*PHYSICAL-DISK,*DIRECTOR)`), and activation of I/O time monitoring of the disk devices (with `//SET-SAMPLING-DEVICE-PARAMETERS DISK-SERVICETIME=*ON`).

These values were monitored using the standard sampling period of 800 ms. When the sampling period is changed, the CPU requirement of SM2 changes linearly with the share of periodically recorded data. For example, in basic monitoring with recording of response times and the UTM monitoring program, the CPU requirement of SM2 drops from 1.5% to just under 1% when the sampling period is changed to 2000 ms.

Disk storage space is required to store the monitored data. It is not possible to specify standard values in absolute measures as the requirement is too heavily dependent on the relevant hardware and software load and the configuration of the monitoring programs. However, it must be borne in mind that in particular the PERIODIC-TASK monitoring program (and still more the DISK monitoring program, which is no longer recommended) makes the output file increase by 50% or more compared to basic monitoring.

10.5 Performance analysis with COSMOS

COSMOS is an event-driven monitor for recording detailed monitored data for the specific performance diagnosis of BS2000/OSD systems. This is an SM2 monitoring program and as such can be controlled using SM2 commands.

COSMOS records the entire course of the system events over time. Approx. 90 different types of event can be registered, e.g. start and end of an I/O operation, system calls (SVCs), and generation and termination of tasks. Each event is identified by a 4-character name. To record the events, COSMOS interfaces, so-called “hooks”, are implemented at various places in the system. Depending on the parameters selected the hook is “opened”, i.e. the data capture code is activated each time the relevant event occurs and a record is written. When COSMOS is terminated, all hooks open at this point are closed.

The data can be collected for all tasks or for tasks selected according to specific criteria (user ID, category, job name or TSN).

The monitored data recorded is written to output files (to tape or disk). Special analysis programs are available for analyzing the COSMOS output files. Generally specialist knowledge is required to evaluate and analyze COMOS measurements.

11 Optimizing performance when developing application software

This chapter provides recommendations for the economical use of resources in the processing of files and notes on the high-performance loading of programs and products.

11.1 Selecting the access method

UPAM and BTAM

UPAM and BTAM are the basic access methods for disks and tapes respectively. Because they are of minor importance for small applications, they have been unjustly neglected. They should always be taken into account when a processing operation is being planned. They include certain functions not available with other access methods. It is often possible to improve performance considerably by using them in preference to other access methods at no detriment to user-friendliness. UPAM, for example, permits random access using the half-page number.

The following advantages apply to both UPAM and BTAM:

- a simple interface for the application program
  There is only one action macro for DMS for each method. The action required is formulated using current parameters. Every read/write request from the program causes a physical I/O operation.
  UPAM and BTAM use shorter path lengths compared with SAM and ISAM, as the blocking and unblocking operations on the records are not necessary.
- a choice between synchronous and asynchronous I/O operations
  asynchronous I/O operations can reduce the runtime of a program considerably
- unrestricted access to the characteristics of the storage medium; optimum utilization of these characteristics is possible
- chaining of jobs to save SVC calls and processing of files created using other access methods
- chaining of I/O operations to reduce system overhead (chained I/O).

Additional advantages with UPAM:

- random access (similar to keyed access) using the half-page number
- possibility of processing files whose contents are damaged
- shareability even when updating
- possibility of combining asynchronous I/O calls with eventing

Additional advantages with BTAM:

- the block length is freely selectable within the range allowed by the hardware
- data carriers from foreign operating systems can be processed.

FASTPAM

Essentially everything that was said about UPAM also applies to the access method FASTPAM.

However you should bear in mind the following restrictions:

- The file format must be NK4
  (`BLKCTRL=NO`, block size=multiple of 4 KB).
- All accesses are made via absolute block numbers.
- There are no synchronization mechanisms when two or more users access the same file.
- FASTPAM supports multisystem networks by means of shared pubsets, but not shared private disks or remote file access.

FASTPAM provides better performance than UPAM because FASTPAM does not run the initialization and validation routines required for I/O operations prior to each I/O, but only once, before the file is opened.

Before the OPEN, the user defines a system environment consisting of ENVIRONMENT and IO-AREA-POOL. These are system and user storage areas used again and again in file accesses.

The maximum improvement in performance is achieved by making the system areas resident. In order to do this, the user must have the appropriate authorization in the user catalog (`DMS-TUNING-RESOURCES=*EXCLUSIVE-USE`) and make a suitable entry in the `RESIDENT-PAGES` operand in the user catalog and when starting the task.

SAM

The SAM access method is designed for the sequential processing of records. It should always be selected if files are not to be shareable and random access is not necessary. This record structure is simple and yet flexible and allows both fixed- and variable-length records. Depending on the application, both types of record length can help to save memory capacity.
The fact that keyed access is not used means that path lengths are considerably shorter than is the case with ISAM. If required, it is a simple operation to convert the file into a file with a key (e.g. using an editor).

The advantages of SAM are:

- reading and writing are performed record by record
- limited updating facilities and high performance (PUTX)
- short program runtimes achieved by reading the next block in advance
- files can also be processed using PAM
- high performance achieved by eliminating shareability during updating

ISAM

ISAM is the most highly developed access method in BS2000. It offers a large number of convenient features which are achieved at the cost of increases in the memory overhead and path lengths. For reasons of compatibility and protection against obsolescence, it often happens that files are kept as ISAM files even though their type and application make them typical SAM files (e.g. source programs). Since it is very easy to convert SAM to ISAM, this is not necessary.

However, ISAM is not a pure keyed access method. ISAM functions best with a set of accesses where a keyword is followed by a series of sequential accesses (indexed-sequential). The distribution and frequency of the keys and in particular whether they are changed during the life of a file, all influence the number of physical I/O operations required to read a particular record. ISAM should only be used if one or more of the following characteristics are required:

The advantages of ISAM are:

- keyed accesses which are largely independent of the distribution of the keys and of any changes which may have been made
- efficient sequential reading following keyed access
- shareability even when updating
- an acceptable deterioration in access performance even after a very high number of single-page extensions

NK-ISAM (Non-Key ISAM) supports both disks with the CKD format (Count Key Data) data format and with the FBA format (Fixed Block Architecture) disks.

Due to the introduction of ISAM pools, the index areas (partly also data areas) are held in the virtual memory. This makes possible a sharp reduction in the physical I/O operations with indexed-sequential accesses, and leads to marked runtime reductions.

BS2000/OSD automates and optimizes the creation of buffer areas (NK-ISAM pools) for NK-ISAM files which are opened with `SHARUPD=YES`. NK-ISAM pools then no longer need to be configured.

NK-ISAM permits the use of secondary keys. If, besides the primary key, additional selection criteria per logical record are maintained in the form of secondary keys, a given set of data can be searched and processed in a number of different ways.

Compared to an application without secondary keys, the resource requirements are slightly higher as long as the “index tree” is not changed; if it is changed (e.g. GET-ELIM, GETKY-ELIM, STORE), resource requirements increase sharply.

PAM

In addition to ISAM, it is possible to imagine further keyed access methods which can better fulfill more specialized requirements. With the aid of PAM, BS2000 offers users the facility of implementing their own access methods. The following example of a hash access method shows that this is possible with a minimum of overhead.

The following characteristics are either specified or required:

- 90% of accesses should be possible with a single I/O operation
- certain frequently used blocks should be able to be read without an I/O operation even if DAB is not used
- the frequency with which changes are performed is high, but it is only very rare that records are deleted, created or significantly extended
- the distribution of the keys is known and follows simply formulated rules
- the file is **not** read sequentially
- sufficient memory is available.

Given these conditions, the number of the block containing the record can be determined from the key with the aid of a simple algorithm. The extent to which the blocks are filled is so limited that the vast majority of updates do not cause it to overflow. In the unlikely event that a block should overflow, the block contains a pointer to the next block, which in turn contains a pointer to the next block should it also overflow. Buffering frequently used blocks does not cause any problems because the user implements the algorithm himself.

Although a special conversion algorithm for converting the key to the block number must be implemented every time that a key is allocated, this process can, compared with the global ISAM procedure, produce considerable increases in performance with large files.

Databases

Databases can be seen as an important extension of the range of access methods. Both the overhead involved in installation and maintenance of the data resources as well as the access path lengths are greater with databases than with other access methods.

They do, however, offer the following additional benefits.

- the way in which data and access paths are organized can be defined by the user and adapted to suit his/her data resources
- unfinished transactions can be reset
- with the aid of logging, the state of the data resources can always be reconstructed to the state applicable shortly before a crash.
- access rights can be defined and monitored
- the amount of redundant data can be reduced as a result of the possibility of creating complex relationships between records
- Queries for multiple keys with the option of logical links permit savings on application program parts.

The software product LEASY is an enhancement of the BS2000 access methods for TP mode.
Based on ISAM, LEASY permits transactions to be reset and all the collected data to be fully restored. In addition, access according to secondary keys and a CALL interface are offered. The use of LEASY must always be considered when no database features are required outside of TP mode.

DIV

DIV (Data in Virtual) is an access method which differs from the traditional access methods such as ISAM, SAM or UPAM in that it functions without structuring the file in records or blocks, without I/O buffer and without special I/O macros (e.g. GET, PUT).

DIV processes only PAM files of the type NK4 (BLKCTRL=NO, the block size being a multiple of 4 KB).

DIV regards the file as a linear sequence of bytes. The DIV function MAP enables a file or a file area to be assigned to an area in the virtual address space. The virtual address space assigned to a file area then forms a “window” in which the pages of the file appear automatically when the customary CPU commands are used to access the corresponding pages in the virtual address space. Data modified in one window can be written to the file using the DIV function SAVE.

In the steady state the file or file area resides in the user's address space and is thus subject to paging. When the file areas are large and insufficient main memory is available, the paging rate increases correspondingly.

Performance is increased when data in the window which was read into the window by preceding accesses (via paging) is accessed repeatedly.

DIV read

The maximum performance enhancement occurs when data in the window is accessed; no disk access is required.
If data access needs to be handled using page transfer, the path length per access is approx. 20% shorter than with read access using UPAM.

DIV write

Secure writing is implemented by writing the modified page back to the file after every write access (DIV-SAVE). In this case the path length per access is approx. 35% longer than with a UPAM write access. The extra effort can only be reduced by decreasing the number of SAVE-IOs (e.g. saving the file once only, when CLOSE is issued). If page transfer is required for write access, the path length is approx. 10% longer than with UPAM write access.

The access method has a great influence on the performance of the entire system and should consequently be selected with care.

11.2 Selecting the processing mode

The processing modes for a file are very closely related to the access methods. Although they can be newly defined each time a processing operation is carried out, they are almost invariably embedded in the processing program and hence fixed. Their influence on system performance is similar to that of the access methods and they should therefore be chosen carefully. In all three phases of file processing (OPEN, access, CLOSE), decisions are made either explicitly or implicitly concerning the choice of mode.

11.2.1 OPEN

Which access options are available for other file users is defined by the type of OPEN call used when the program is designed. Consequently it should be obvious that processing modes exceeding the requirements of the intended application should not be chosen. A person who only wishes to read and does not open the file with INPUT prevents other users from gaining read access. In the case of files which are of general interest, the effort of making a decision concerning reading/updating at the time of the OPEN command is worthwhile.

Opening a file is a complicated process. Because of the overhead involved, it should be done as rarely as possible. Even the requirement that a file should be locked for as short a time as possible is subordinate to this requirement. If necessary, one must resort to the access method PAM or ISAM using SHARUPD mode.

If it can be accepted that a file is not 100% up-to-date, then it is worth installing a copy for read accesses. This should be write-protected in order to prevent users from opening the copy file for writing purposes.
This solution is more efficient because using SHARUPD mode causes an increase in the access overhead. If the probability of a collision is small, then the additional effort involved here is not worthwhile.

11.2.2 Accesses

In this case also, the rule applies that files should only be locked if it is vital for correct operation. Performance is reduced if too much emphasis is placed on safety considerations. It is possible that a lock or even SHARUPD mode can be avoided by allowing overbooking and by separating the component parts of records.

The use of the RELSE call to force a state of consistency for an open file should be used with caution because of the physical I/O operations involved.

Keyed accesses in ISAM are more complex than sequential accesses. If necessary they can be replaced in part by sequential accesses after the program or the file has been reorganized.

If files have to be accessed in a multiprocessor system, then the pros and cons of transferring a file or parts of it in advance should be considered. This depends on the structure of the file and of the accesses. It is also of great importance whether a multiprocessor system access can be executed via the network or via disks operated in shared pubset or shared private disk mode and which can be addressed by several processors.

If files are being created or extended, then an improvement in performance can be achieved by avoiding the secondary space allocation.

The more compact file structure which results from this gives rise to a variety of long-term improvements in performance.

It should also be mentioned at this point that parts of a file can be supported and monitored selectively by DAB (report “Reads resp.Writes for internal area” in the report group DAB of openSM2).

11.2.3 CLOSE

Care should be taken that open files are always locked for certain groups of users. In addition to this, they occupy memory space and the copy on the disk is not consistent. Therefore, it is advisable to close the file as soon as it is no longer needed. This should not, however, result in otherwise superfluous OPEN calls. In the case of tape files, CLOSE involves rewinding the tape, which blocks further access to the drive concerned for some time. Consequently it is advisable to issue the call in time to prevent subsequent operations from being obstructed.

If the contents of the file have been reduced considerably and if they are likely to remain small for a long time, then its size should be reduced after the execution of CLOSE by means of `/MODIFY-FILE-ATTRIBUTES` (operand `SPACE=*RELEASE(...)` given) or `FILE` macro with a negative value for the `SPACE` parameter.

11.3 High-performance loading of programs and products

This section will first provide a general description of the processes involved in loading a program/product. This forms the basis for a description of the measures which should be considered by users in order to optimize the load time for a program/product.

It is generally assumed that the programs or modules are stored in PLAM libraries. This means that the programs/modules are linked with BINDER and loaded with `/START-EXECUTABLE-PROGRAM`.
The information provided here relates only to working with LLMs. No explicit reference is made to the linking and loading of phases, although a number of the measures described for speeding up the loading process can certainly also be used in such cases.

11.3.1 Basic stages involved in linking/loading a program/product

Diagram of the entities involved (the LLMs to be loaded are located in the library specified in the load call):

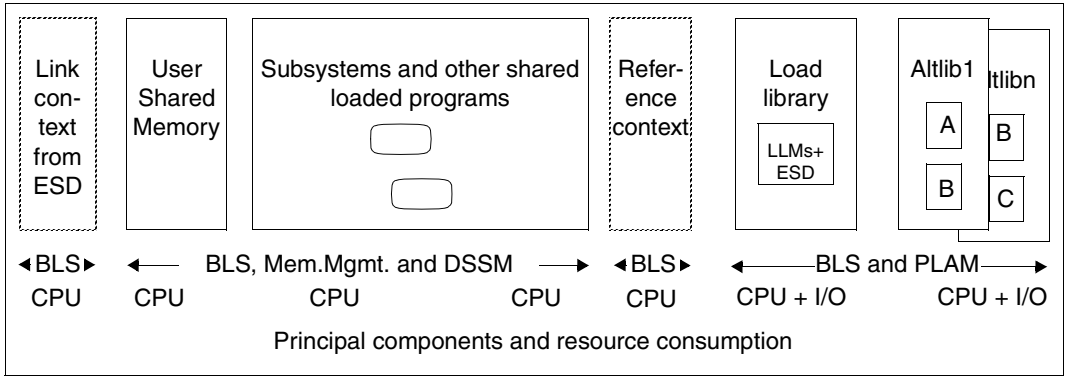


Figure 21: Stages involved in linking/loading a program/product

External references are resolved from left to right, i.e. the system searches the entities in the following sequence:

a) Link context (according to the ESD information in LLM); this is empty if the program is a standalone program (with `/START-EXECUTABLE-PROGRAM`)

b) User Shared Memory (= memory pools); a number of memory pools can be specified when `/START-EXECUTABLE-PROGRAM` is issued (default: deactivated)

c) (Pre-loaded) DSSM subsystems and other shared loaded programs (default: activated when `/START-EXECUTABLE-PROGRAM` is issued)

d) Reference context in memory: this is created dynamically during loading and is used to satisfy external references which could not be resolved in the link context

e) Library specified in the load call

f) alternative libraries (Altlibs) in the specified sequence.

11.3.2 General notes on improving loading speed

- All the PLAM libraries involved should be organized as efficiently as possible, i.e.
  - they should contain no redundant LLMs
  - they should be completely reorganized, so that the LLMs they contain are not split into an unnecessary number of extents
  - the library itself should comprise as few extents as possible
  - the library should possibly be created using (STD,2).

  All these measures reduce the number of I/Os and improve runtime behavior. The effectiveness of these measures is, of course, dependent on the original structure. A library created with (STD,2) will be larger than the original library by about 1 KB per member. The reduction in I/O operations will be greater, the more members the library has, provided that these members are not too small. Runtime improvements can then amount to over 10%.
- The `/START-EXECUTABLE-PROGRAM` comand should always be used when loading an LLM object. This initiates more effective BLS search procedures, allowing a reduction in CPU time of about 10%.
- If modules have the READ-ONLY attribute, one slice should be generated for read-only modules and one slice for read/write modules. This speeds up the loading process by reducing both CPU requirements and I/O operations.
- The size of the LLMs and thus the PLAM overhead are both reduced if all symbols not required for subsequent BINDER runs are set to “invisible” with the BINDER statement

  ```
  //MODIFY-SYMBOL-VISIBILITY SYMBOL-NAME=...,SYMBOL-TYPE=...,
                             SCOPE=...,VISIBLE=*NO,...
  ```
- If public/private slices are to be used, format 2 should be used in order to reduce the CPU requirements of BLS and DSSM (see the BINDER statement `//SAVE-LLM ...`). `FOR-BS2000-VERSIONS=*FROM-V11`).

  The operand `SUBSYSTEM-ENTRIES` in the BINDER statement `//START-LLM-CREATION` allows you to specify a symbolic name for the links between the private and public slice. This name must also be defined in the DSSM catalog. (DSSM is called for every link with the old format.)
- You should ensure that the LOGGING parameter in the DSSM parameter file is **not** set to ON (default: OFF). ON would cause DSSM data to be logged when subsystems are loaded.

- RESOLVE-BY-AUTOLINK link from **several** libraries:
  The BINDER statement `//RESOLVE-BY-AUTOLINK LIBRARY=A,B,C` can be replaced by the following specification, which improves performance:

  ```
  //RESOLVE-BY-AUTOLINK LIBRARY=A
  //RESOLVE-BY-AUTOLINK LIBRARY=B
  //RESOLVE-BY-AUTOLINK LIBRARY=C
  ```

  During dynamic loading, this means, for instance, that new links which need to be resolved from library B are only searched for in library B (and possibly library C), but not first in library A (and then in B and possibly C).

11.3.3 Structural measures for reducing resource requirements

Unfortunately, it is rarely possible to quantify precisely the effect of each of the measures described below. The effects depend to a large degree on the structure of the load objects, the subsystems and the libraries. One can, however, generally assume that a gain of at least 10% can be achieved with each of the measures.

Reduce the number of libraries or optimize their contents

- Merge alternative libraries

  If alternative libraries are merged to form a single library (this could even be the specified load library), this results in a considerable reduction in search operations in BLS/PLAM, which in turn leads to a saving in I/O operations and CPU time which is sometimes considerable greater than 10%.

  Merging is generally only recommended if no entries or CSECTs exist with the same name. If this is not the case, care should be taken when the libraries are merged, that the “correct” CSECTs or entries are included.

- Link in modules from libraries permanently

  If the contents of the alternative libraries and the specified library never or only rarely change, it may be advisable to link the modules into the load object permanently. This reduces the CPU requirements on loading. On the other hand, static linking means that the load object will increase in size, thus generally causing more I/O operations, which in turn increases the runtime.
  This measure makes sense, therefore, if only small modules are involved and if the alternative libraries contain a particularly large number of entries.

  If you adopt this strategy for improving performance, you must, of course ensure that the link procedure is repeated each time the alternative library is modified.

- Mixed strategy

  If, for instance, the contents of only one of the alternative libraries is modified regularly, it may be advisable to adopt a mixed strategy using both of the methods described above.

Restricting dynamic loading of shared code

- Do not dynamically load/link shared code

  Many programs do not need to dynamically load/link from a subsystem/shared programs or user shared memory. These programs should be started as follows:

  ```
  /START-EXECUTABLE-PROGRAM ...,
     DBL-PARAMETERS=*PARAMETERS(RESOLUTION=*PARAMETERS(SHARE-SCOPE=*NONE)),
     ...
  ```

  This prevents BLS from calling DSSM to search the subsystems for the entry, resulting in an often considerable reduction in CPU time.

- Only dynamically load from system memory

  This is the default. It is not possible to restrict the search for the entry performed by DSSM to specific subsystems or to shared programs only.

- Only dynamically load from user shared memory

  With `/START-EXECUTABLE-PROGRAM` the `DBL-PARAMETERS=*PARAMETERS(RESOLUTION=*PARAMETERS(SHARE-SCOPE=*MEMORY-POOL(...)))` parameter permits limitation to used shared memory. It is also possible to further restrict the operation to specified memory pools.

- Preload user shared memory

  As far as possible, all shareable sections of the applications in use should be preloaded (e.g. FOR1-RTS). This means that they are loaded only once and only need to be linked (with little overhead) for each further load operation. This strategy generally results in a saving of considerably more than 10% in I/O operations and CPU time.

Eliminating ESD information

- Complete elimination

  This measure only applies for standalone programs. These are LLMs which are not called by other programs and which are completely prelinked. For programs of this type, the load operation can be accelerated if the ESD and other BLS information tables are eliminated using the appropriate BINDER call. Call BINDER as follows:

```
/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
```

  This parameter specification means that only those BLS structures are created which are required for loading a simple program, thus reducing the size of the object. This saves CPU time and I/O operations when the program is loaded. All the information required for other processing is then no longer available.

  This means that

  – these parameters cannot be used for load objects which are split into a number of slices
  – no LLM updates can be carried out on an object of this type.
  – relocatable objects (e.g. as part of a runtime system) cannot be created.

  The (supplementary) parameter setting `REQUIRED-COMPRESSION=*YES` should not be selected, since, although up to 10% of I/O operations can be saved, approx 40% more CPU time is required.

- Partial elimination

  When merging LLMs or sub-LLMs to form a prelinked module with a single CSECT, you should specify what ESD information is to remain in the external address book with the BINDER statement `//MERGE-MODULES`. This can result in a considerable reduction in BLS overhead during loading.

11.3.4 Improving the load speed for C programs

You can improve the load speed of C programs by linking the few modules which cannot be preloaded from the C runtime system to the compiled program. Use the following supplementary BINDER statement:

```
//RESOLVE-BY-AUTOLINK LIB=<Partial-Bind-Lib>
```

The <Partial-Bind-Lib> is installed under the name SYSLNK.CRTE.PARTIAL-BIND. It is also necessary to make external names invisible in order to suppress duplicate symbols:

```
//MODIFY-SYMBOL-VISIBILITY SYMBOL-NAME=*ALL,VISIBLE=*NO
```

These two statements result in a considerable reduction in CPU time and particularly in runtime when loading C programs. The load object is only a few pages (approx. 18) longer than the compiled program.

The CRTEC and CRTECOM subsystems must be loaded to allow CRTE to be loaded in the shared code area.

11.3.5 Use of DAB

If the linked object and/or the libraries used are very large, with the result that a large number of I/O operations are required during loading, you should check whether it is possible or necessary to use DAB (Disk Access Buffer).

DAB permits such a considerable reduction in I/O operations on frequently accessed files that in extreme cases the runtime depends only on the CPU requirements.

A cache area must be selected for the load object which is the same size as the load object itself. This is particularly simple if the object is the only member of the library. You can then issue /START-DAB-CACHING AREA=*FILE(FILE-AREA=<lib>),CACHE-SIZE=*BY-FILE to reserve a main memory area with the same size as the library. This is filled the first time the object is loaded, with the result that no time-intensive I/O operations are required for any subsequent load operations.

The cache areas for the libraries can generally be created smaller than the total size of the libraries, since it is generally the case that not all of the modules contained in the libraries will be dynamically loaded. In this case, the cache size must be specified explicitly (in KB or MB) when /START-DAB-CACHING is issued. It is possible to monitor the cache hit rate for each cache area using openSM2 or /SHOW-DAB-CACHING. The hit rate should be at least 80%. If this is not the case, the size of the cache area must be increased.

12 Appendix

12.1 Standard values for BS2000/OSD servers

The tables below show some performance data and standard values for S and SQ servers. The recommendations apply for typical TP operation with a normal background load.

- Relative Performance Factor RPF and number of CPUs.
- Typical main memory configuration.
  The actual main memory requirement depends on the load with which the server is operated. TP and dialog loads normally have a significantly greater main memory requirement than batch loads.

  On SQ servers, the main memory is distributed between BS2000/OSD and Dom0/X2000 (I/O processor). A recommendation for the configuration and distribution of main memory is shown in the relevant tables.
  Information on when more main memory than recommended there should be used is provided in section “Main memory on SQ servers” on page 32.
- Maximum recommended number of DMS I/O operations.
  An IT system is used economically when it is executing a balanced mixture of programs requiring an intensive use of calculation functions and programs requiring an intensive use of I/O functions. The recommended maximum number of DMS I/O operations per second is not an upper limit determined by technical requirements, but guarantees that only a certain proportion of the CPU capacity is used to perform I/O operations. It can be exceeded without problems, provided bottlenecks in the peripherals are avoided.
- Maximum recommended number of paging I/O operations.
  The maximum recommended paging rates specified for transaction mode or interactive mode ensure that only a certain proportion of the CPU capacity is used for paging. To avoid delays through paging (especially in transaction processing mode), you should try to aim for less than the stated values in view of the main memory available.
- Maximum recommended number of SVCs in TU.
  When the maximum number of SVC calls per second is reached, the effort required for the interrupt analysis and termination handling of the system routine (i.e. not the resources used for the system routine itself) amount to approx. 5% of CPU time.

Standard values for S servers

S server	RPF	Number of CPUs	Representative main memory size in GB	Maximum recommended number (per second)			
				DMS I/O operations	Paging input/outputs		SVC in TU
					TP mode	Interactive mode	
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

Table 2: Standard values for S servers (part 1 of 2)

S server		RPF	Number of CPUs	Representative main memory size in GB	Maximum recommended number (per second)			
					DMS I/O operations	Paging input/outputs		SVC in TU
						TP mode	Interactive mode	
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

Table 2: Standard values for S servers (part 2 of 2)

Standard values for SQ servers

SQ servers		RPF	Number of CPUs	Representative main memory size in GB [1]	Maximum recommended number (per second)			
					DMS I/O operations	Paging input/outputs		SVC in TU
						TP mode	Interactive mode	
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

Table 3: Standard values for SQ servers (part 1 of 2)

SQ servers	RPF	Number of CPUs	Representative main memory size in GB [1]	Maximum recommended number (per second)			
				DMS I/O operations	Paging input/outputs		SVC in TU
					TP mode	Interactive mode	
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

Table 3: Standard values for SQ servers (part 2 of 2)

[1] To determine the memory available for BS2000 guest systems, the share for Dom0/X2000 (default: 30%) and, if necessary, the main memory shares for Linux or Windows guest systems must be deducted from this value. It must also be borne in mind that approx. 40% (depending on the system parameter JTABSMEM, see page 348) of the main memory available for BS2000/OSD is used as a JIT buffer and is therefore not directly available to the operating system and applications, see section “Main memory on SQ servers” on page 32.

Example: On an SQ210-40F (main memory: 32 GB) 8 GB of the memory are used for a Linux guest system. BS2000/OSD and the BS2000 applications can then use around 8.6 GB of main memory in accordance with the following calculation:
32 GB - 8 GB (Linux guest system) - 32 * 0.3 GB (Dom0/X2000) = 14.4 GB * 0.6 (JIT) = 8.64 GB.

12.2 Performance relevant system parameters

With the aid of the BS2000/OSD system parameters (which are defined in the SYSOPT-CLASS2 parameter set of the `SYSPAR.BS2.version` parameter file), the system behavior can be adjusted to suit the relevant requirements.

This section describes the following system parameters which are relevant for performance:

For a complete list of parameters with detailed descriptions of each, see the appendix of the “Introductory Guide to Systems Support” [10].

BMTNUM and CATBUFR

Management of and accesses to the file catalog TSOSCAT are carried out internally by the Catalog Management System (CMS). For a functional description and further suggestions for tuning, see section “Creating system files” on page 156.

The system parameters BMTNUM and CATBUFR should always be considered together. When deciding which values to assign to these parameters, you must take into account the number of pubsets with which the system is to be run (cf. CATBUFR).

Parameter name	Value range	STD value
BMTNUM	0 ... 255	32

This parameter defines the number of I/O buffers and Buffer Management Tables (BMT) for the catalog management system (CMS). This number of buffers is created once for each pubset and once globally for all private volumes.

Each block of the TSOSCAT file can contain max. 13 file entries or 18 JV entries. For each CMS I/O buffer (4096 bytes) a table (BMT) is created (approx. 178 bytes). It is used for managing the current status of the block in the buffer.

The default value for BMTNUM is 32.
Each increase also increases the probability that CMS will find a file entry or JV entry without a physical input operation being carried out. The positive effect of this strategy is particularly noticeable in the case of frequent access to adjacent catalog entries and thus to one and the same catalog block.

Parametername	Werte-Bereich	STD-Wert
CATBUFR	N/Y	N

If this parameter is set to Y, the CMS buffers and the BMTs will be created as resident (in class 3 memory).
If it is set to N (default), they will be created as pageable (in class 4 memory).

The pubset-specific setting of the number of CMS buffers and their memory class is carried out using the commands

```
/ADD-MASTER-CATALOG-ENTRY
/MODIFY-MASTER-CATALOG-ENTRY
/IMPORT-PUBSET
```

and the operands `NUMBER-OF-BUFFERS` and `RESIDENT-BUFFERS`.

Defined values are sought in the following order:

1. Explicit parameter specification in /IMPORT-PUBSET
2. Specification with /ADD-MASTER-CATALOG-ENTRY or /MODIFY-MASTER-CATALOG-ENTRY
3. Specification via the system parameters BMTNUM and CATBUFR

If no private disks are used, buffer allocation for private disks can be prevented with BMTNUM=0. In the case of /MODIFY-MASTER-CATALOG-ENTRY (for home and paging pubsets) or /IMPORT-PUBSET the number of CMS buffers and their storage class must then be specified.

The size of the address space requirement per pubset or of the private volumes for the CMS buffers can be calculated using the following formula:

```
CMS buffer (bytes) = (4096 + x) * Number of buffers + 384
```

x = 178 for pubsets; x = 200 for private volumes

If, for example, with smaller systems a large number of memory-resident buffers are created, the catalog operations will be faster but this advantage may be offset by an increased paging rate. If the catalog is accessed frequently, use of the software product SCA (Speed Catalog Access) is imperative.

CONSDDE7

Parameter name	Value range	STD value
CONSDDE7	N/Y	N

The system parameter CONSDDE7 is used to reduce the message scope on the console.

The system message DMSODE7 SAM FILE CLOSED is output to SYSOUT and the console if Y is set, and only to SYSOUT if N is set.

If a magnetic tape is assigned as the output device for SPOOL, this message will appear at the end of every SPOOL output. The default is N.

DEFLUID

Parameter name	Value range	STD value
DEFLUID	:catid:$userid	$TSOS

In the /LOAD- or /START-(EXECUTABLE-)PROGRAM, /CALL-PROCEDURE, /ENTER-PROCEDURE and /ENTER-JOB commands the specification of the catalog and user IDs can also be omitted in the FROM-FILE operand if the required file is not cataloged under the caller's ID.

In a second search run (secondary read), the ID specified under this system parameter is searched.

The possibility of specifying an ID other than the default value $TSOS here serves to relieve the TSOS ID. It is recommended that you save all programs and procedures provided to users by the computer center under a separate library user ID and inform the system of this by means of the DEFLUID option.

DESTLEV

Parameter name	Value range	STD value
DESTLEV	0/1/4/5/6	0

DESTLEV defines whether file extents are to be overwritten with binary zeros when released from the system (/DELETE-FILE or /MODIFY-FILE-ATTRIBUTES with memory release).

With large files this can be a very lengthy process (lasting minutes).

DMPRALL, DMSCALL, DMMAXSC

These three system parameters must be defined in coordination with each other. They are used as defaults for the space allocation when creating and extending files. The values apply to all private volumes. For pubsets they only apply if nothing else is defined in the pubset's MRSCAT entry. In addition, the allocation unit specified when the pubset was created with SIR is considered depending on the data format (K, NK2, NK4).

Defined values are sought in the following order:

1. Direct specification in the /CREATE-FILE or /MODIFY-FILE-ATTRIBUTES command
2. Entry in MRSCAT
   - If specified with /MODIFY-MASTER-CATALOG-ENTRY (ALLOCATION operand), the modification will not take effect until after the next import of the pubset.
   - If /MODIFY-PUBSET-SPACE-DEFAULTS (operand PRIMARY-ALLOCATION / SECONDARY-ALLOCATION / MAX-ALLOC) is used, the definition takes effect immediately.
3. Specification via system parameters

If the values ascertained in this way are not a multiple of the allocation unit defined when the pubset was configured using SIR, they are rounded up to the next multiple.

Primary allocation

Parameter name	Value range	STD value
DMPRALL	3...65535	3

Primary allocation for a file in PAM blocks if the SPACE operand was not given a value in /CREATE-FILE oder /MODIFY-FILE-ATTRIBUTES (or in the FILE macro).

Secondary allocation

Parameter name	Value range	STD value
DMSCALL	3...65535	3

Secondary allocation for a file in PAM blocks if this was not supplied in the SPACE operand in /CREATE-FILE oder /MODIFY-FILE-ATTRIBUTES (or in the FILE macro).

After each file enlargement, the amount of the increase is doubled as compared to the value at the previous enlargement. Secondary allocation (SA) at the (i+1)th enlargement equals twice the secondary allocation at the i-th enlargement: `SA(i+1) = 2 * SA(i)`

This formula continues to be used until double the value of the current secondary allocation exceeds the value DMMAXSC. Thereafter the formula `SA(i+1) = SA(i)` applies.

Maximum value

Parameter name	Value range	STD value
DMMAXSC	3...65535	48

When double the value of the current secondary allocation exceeds the value of DMMAXSC, the value for the secondary allocation is no longer modified.

Example

File enlargements with the values: DMPRALL 15, DMSCALL 3, DMMAXSC 48. In this example, DMMAXSC could contain any value between 48 and 96 without the behavior changing.

15	+	3	=	18	(1st file enlargement)
18	+	6	=	24	(2nd file enlargement)
24	+	12	=	36	(3rd file enlargement)
36	+	24	=	60	(4th file enlargement)
60	+	48	=	108	(5th file enlargement)
108	+	48	=	156	(6th file enlargement, no change to the secondary allocation because the value would then be greater than DMMAXSC)

When a file is created on disk or its size changes, within DMS the smallest amount that can be changed is an allocation unit.

The smallest allocation unit is for

Private volumes: 3 PAM blocks

pubsets: 3 PAM blocks (K, NK2 format)
4 PAM blocks (NK4 format)

Each dynamic file enlargement leads to a series of management I/O operations in the system (updating the entries in the user ID, the file catalog, the F1 label with private volumes) and thus incurs a considerable system overhead. To minimize this overhead, the DMSCALL parameter should be set to at least 4 allocation units.

To reduce the number of dynamic file enlargements (or avoid them), the operand `SPACE=*RELATIVE(PRIMARY-ALLOCATION=xx,SECONDARY-ALLOCATION=yy)` is available to all users in `/CREATE-FILE` oder `/MODIFY-FILE-ATTRIBUTES`.

Every user should try to give the file its expected final size already when creating it, by means of a corresponding primary allocation.

Example

A program writes 456 blocks sequentially with the PAM macro call. The AINF macro call determines the number of I/O operations.

/CREATE-FILE ...,SPACE=REL		**No. of physical I/Os**	**No. of extents acc. to**
(PRI-ALLOC=...,	SEC-ALLOC=...)	**according to AINF**	/SHOW-FILE-ATTR
3	3	480	20
6	6	476	18
30	30	467	9
192	192	459	3
456	30	456	1

The results can vary depending on the respective disk occupancy levels.

EACTETYP, L4MSG, MSGCENTL, MSGCENTN, MSGDLAM, MSGFILii, MSGNOFL

These parameters belong together:

Parameter name	Value range	STD value	Meaning
EACTETYP	0/1/2/3	0	Defines which of the following messages are to be output via the console: BLS0500, BLS0517, BLS0519, BLS0523, BLS0524, BLS0526, BLS0539, BLS0551, BLS0552. 0 none of the messages 1 only message BLS0519 2 all the above messages 3 all the messages except for BLS0519
L4MSG	0/1	0	Controls the output of message EXC044E "ACCEPT ALLOCATION REQUEST..." 0 Message is not output 1 Message is output
MSGCENTL	36...2500	200	Defines the length of an entry in class 4 memory for the message processing. The value must be a multiple of 4. This area of the class 4 memory is used to buffer the most recently and frequently used messages in order to save on file accesses
MSGCENTN	0...32767	32	Defines the number of entries in class 4 memory (see above) for the message processing. *Note:* The class 4 memory requirement can be calculated by multiplying the values of MSGCENTL and MSGCENTN
MSGDLAM	0...99	0	Number of message files to be processed via DLAM
MSGFIL01 MSGFIL02 MSGFIL03 MSGFIL04 : MSGFIL15 *)	filename(n)		Number and names of the message output files. The following default values apply for the MSGFIL01 and MSGFIL02 parameter: MSGFIL01: SYSMES.BS2CP.version MSGFIL02: SYSMES EKP.01 There are no standard names for the parameters MSGFIL03 through MSGFIL15, i.e. these files are not created. The file name must be fully qualified and can only be cataloged with the user ID $TSOS.
MSGNOFL *)	0...15	2	Number of message files that were specified via the MSGFILxx parameter.

*) for details see the section “Message files” on page 169

EAMMEM, EAMMIN, EAMSEC, EAMSIZ

Parameter name	Value range	STD value
EAMMEM	0...2730	0
EAMMIN	4...64512	3000
EAMSEC	1...64512	200
EAMSIZ	4...64512	64512

These parameters define the size of the SYSEAM file. They are valid for all :catid:SYSEAM files if they are created on several pubsets and if nothing else is defined in the MRSCAT entry for each pubset. In addition, the allocation unit specified when the pubset was created with SIR is considered depending on the data format (K, NK2, NK4).

The SYSEAM file is managed in EAM allocation units. When a DMS allocation unit is 3 PAM pages, the EAM allocation unit is also 3 PAM pages; in the case of all other DMS allocation units the EAM allocation unit is 4 PAM pages. The value of EAMMIN defines the minimum size. If necessary, it is extended by the value of EAMSEC (number in EAM allocation units) until the maximum size of 64512 EAM allocation units is reached. Should the SYSEAM file shrink (e.g. in the case of /DELETE-SYSTEM-FILE SYSTEM-FILE=*OMF or /EXIT-JOB), it is reduced in increments of 8 allocation units, but does not go below the value of EAMMIN.

The value of EAMMIN must be set such that in normal operation 80% of all requests to save blocks in the system file SYSEAM can be satisfied without it being necessary to enlarge the file. Experience shows that a size of 4000 allocation units for the initial setting is advisable.

If the value chosen for EAMMIN is too low, this will lead to a higher load on the system due to many dynamic file enlargements and uncontrolled growth of the SYSEAM file.

By choosing suitable values for EAMMIN and EAMSEC or the EAM operand (MINIMAL-SIZE, SECONDARY-ALLOCATION) in /ADD-MASTER-CATALOG-ENTRY und /MODIFY-MASTER-CATALOG-ENTRY, you can keep the variation of the file size of SYSEAM to a minimum or suppress it completely.

It is important to keep the number of required secondary allocations per pubset as low as possible.

EAMSIZ is the maximum value in allocation units within the $TSOS.SYSEAM file that are available to one user alone.

Standard values for the EAM system parameter are heavily dependent on the expected load. The values for EAMMIN and EAMSEC should both be set to a multiple of 8, as this produces optimum adjustment to the internal table structure of EAM.

The value of EAMMEM defines the size of the class 4 memory (in units) used for EAM. It should be a multiple of 8. The pubset-specific definition is implemented using the EAM=*PARAMETERS(VIRTUAL-MEMORY=...) operand in /ADD-MASTER-CATALOG-ENTRY or /MODIFY-MASTER-CATALOG-ENTRY.

This area is created for the home pubset only and is part of the SYSEAM file. First the class 4 memory area is filled, and only once it is full does writing to disk begin (no cache).

Beneficial use (speeding up compilation runs by saving on I/O operations) is only possible if there is sufficient main memory available. Otherwise the benefit of cutting down on physical I/Os on the SYSEAM file will be offset by increased paging of the class 4 memory pages.

ETMFXLOW

Parameter name	Value range	STD value
ETMFXLOW	127...256	256

Tasks with an external priority greater than or equal to ETMFXLOW are not affected by aging in processor state TU. This means that for these tasks the dynamic setting of the internal priority (see section “Introduction to the PRIOR concept” on page 203) is deactivated. Any increase in priority acquired in processor state TPR becomes invalid on transition to TU.

Tasks with an external priority greater than or equal to ETMFXLOW are suitable for definition of a background load. For the main application(s) a correspondingly large external priority difference must be selected. This function must be used very carefully and its effects must be monitored.

L4SPDEF

Parameter name	Value range	STD value
L4SPDEF	66...n	2500

The L4SPDEF parameter determines the limit value for achieving saturation level 4 in pubsets. The level is regarded as achieved when fewer PAM blocks are free than defined here. The default value is recommended.

The value is valid only as long as no other pubset-specific definition is specified using /ADD-MASTER-CATALOG-ENTRY or /MODIFY-MASTER-CATALOG-ENTRY with the ALLOCATION=*PARAMETERS(SATURATION-LEVEL4=...) operand.

SSMAPRI and SSMASEC

Parameter name	Value range	STD value
SSMAPRI	3...65535	24
SSMASEC	3...65535	24

These parameters define the size of the primary and secondary allocation for system output files. The parameters are valid for files created with the following commands:

```
/ASSIGN-SYSOUT
/ASSIGN-SYSLST
```

Whenever one of these system output files is used for the first time, a cataloged file is created with the size defined by SSMAPRI and SSMASEC. The file names are:

```
S.OUT.tsn.yyyy-mm-dd.hhmmss.cnt
S.LST.tsn.yyyy-mm-dd.hhmmss.cnt
```

SSMAPRI Primary allocation in PAM blocks for the task-specific spoolout files or for reassigned system files which were created with an `ASSIGN ...` command.

SSMASEC Secondary allocation in PAM blocks for the task-specific spoolout files or for reassigned system files which were created with an `ASSIGN ...` command.

The optimum values for primary and secondary allocation of these SPOOL output files are heavily dependent on the type of the load. Unless experience indicates otherwise, the default values of 24 PAM blocks for each should be used.

TEMPFILE

Parameter name	Value range	STD value
TEMPFILE	C'#'/'@'/'NO'	C'NO'

The “temporary files” (TEMPFILE) procedure offers users the option of working with cataloged files and/or job variables which are automatically deleted by the system when `/EXIT-JOB` or `/LOGOFF` is specified.

The procedure is very resource-intensive, i.e. it requires a large amount of system address space and CPU capacity. In return it automatically clears away unnecessary work files. Systems support can use this option to specify whether the procedure may be used and how the file and JV names are denoted.

If the option is not set, the “temporary files” procedure is not incorporated in the system (default).

The catalog names for temporary files or JVs are as follows:

```
S.mmm.nnnn.filename
S.mmm.nnnn.jv-name
```

`mmm` is replaced by the “sysid” of the home pubset,
`nnnn` is replaced by the TSN of the task held by the user of the file.

12.3 Buffer size of the CISC firmware

The following system parameters jointly control the main memory usage on SQ servers as of SQ200 for the Just-In-Time Compilation of /390 code through the CISC firmware.

JTABSMEM

Parameter name	Value range	STD value
JTABSMEM	0...65535	0

This parameter defines the maximum JIT buffer memory size. The memory is requested as resident and must therefore match the memory available on the machine.
The parameter default is 0 and in this case the CISC firmware calculates the maximum memory space as follows: `MEMORY-SIZE * 40%`.
The computed value of `JTABSMEM` is logged with the message `HJT0039` on the console. The memory size can also be set to a fixed value by specifying a value between 1 and 65535. The parameter can be raised during operation.

When the threshold value is approached the message `HJT0035` with a corresponding JIT Memory Saturation Level is issued. When level 1 is reached it is advisable to increase the threshold value. When level 2 is reached you are urgently recommended to increase the threshold value, otherwise considerable losses of performance must be expected.

JTSTDMEM

Parameter name	Value range	STD value
JTSTDMEM	1...65535	16

This parameter, JTSTDMEM, determines how much JIT buffer the system can make available to each task within the global JIT buffer, provided that no other task-specific settings have been made by the `/MODIFY-DBL-DEFAULTS` command.

The default size of the task-specific JIT buffer is 16 MB. As the CISC firmware only requests as much memory as is necessary to run the current /390 programs, we recommend that you do not change the STD value. When the JTSTDMEM value is reached the message `HJT0032` is given, the task-specific JIT buffer is reset and rebuilt. As the associated reduction in performance is normally low, the message is only issued in the CONSLOG file.

Enlarging the JIT buffer is only worthwhile in rare cases (e.g. a task with particularly large working set requirements).

To increase the size of the JIT buffer the new size must be specified with the following command (`nn` = number of MB)

```
/MODIFY-DBL-DEFAULTS SCOPE=*CMD-CALLS(CISC-COMPILATION=*YES(WORKSPACE=nn))
```

The change takes effect for all DBL calls (/START–, /LOAD–(EXECUTABLE–)PROGRAM, BIND). The command must be inserted into all Enter jobs or procedures if programs that need a larger JIT buffer are loaded there.

JTMAXMEM

Parameter name	Value range	STD value
JTMAXMEM	1...65535	128

The parameter JTMAXMEM fixes the maximum value which can be set as task-specific using the command /MODIFY–DBL–DEFAULTS.

JTSHMEM

Parameter name	Value range	STD value
JTSHMEM	0...256	64

Specifies how much storage space JITSYS is to use for storing shared compiled code (in MB).

Shared compiled code is created when emulating subsystems loaded in class 4 memory. If no such subsystems exist or these are not to be compiled as “shared”, the creation of shared combined code can be prevented with JTSHMEM = 0. The JTSHMEM value should be adjusted to the total amount of class 3 memory available.

The storage size specified with JTSHMEM is allocated immediately when JITSYS is initialized. Modification of the value during ongoing operation will, until further notice is given, not be supported. The value can only be specified in steps of 4; other values are rounded up to the next multiple of 4.

The default value for JTSTDMEM should be changed together with the BIG-SHARE-SHRSIZE parameter in the MEMORY parameter set of the startup parameter service

The connection to the SHXSIZE parameter of the setup parameter service is indirect:

When the value selected for SHXSIZE is very high and the corresponding number of programs is actually loaded as “shared”, it also makes sense to increase JTSTDMEM.

Related publications

You will find the manuals on the internet at *http://manuals.ts.fujitsu.com*. You can order manuals which are also available in printed form at *http://manualshop.ts.fujitsu.com*.

[1] **ARCHIVE**
User Guide

[2] **BCAM**
User Guide

[3] **BS2000/OSD-BC**
Files and Volumes Larger than 32 GB
User Guide

[4] **SQ Business Server**
Operation and Administration
User Guide

[5] **DAB** (BS2000/OSD)
Disk Access Buffer
User Guide

[6] **BS2000/OSD-BC**
Utility Routines
User Guide

[7] **DRV** (BS2000/OSD)
Dual Recording by Volume
User Guide

[8] **BS2000/OSD-BC**
DMS Macros
User Guide

[9] **BS2000/OSD-BC**
Introduction to DVS
User Guide

[10] **BS2000/OSD-BC**
Introductory Guide to Systems Support
User Guide

[11] **FDDRL** (BS2000/OSD)
User Guide

[12] **HIPLEX MSCF** (BS2000/OSD)
BS2000-Processor Networks
User Guide

[13] **HNC**
High-Speed Net Connect
User Guide

[14] **HSMS** (BS2000/OSD)
Hierarchical Storage Management System
User Guide

[15] **BS2000/OSD-BC**
Commands
User Guide

[16] **LEASY** (BS2000/OSD)
Program Interface and Strategies
User Guide

[17] **BS2000/OSD-BC**
Migration Guide
User Guide

[18] **openSM2** (BS2000/OSD)
Software Monitor
User Guide

[19] **openUTM**
Concepts and Functions
User Guide

[20] **openUTM**
Administering Applications
User Guide

[21] **openUTM**
Generating Applications
User Guide

[22] **openUTM** (BS2000/OSD)
Using openUTM Applications under BS2000/OSD
User Guide

[23] **PCS** (BS2000/OSD)
Performance Control Subsystem
User Guide

[24] **ROBAR** (BS2000/OSD, Linux)
Controlling MTC Archive Systems
User Guide

[25] **SDF** (BS2000/OSD)
SDF Management
User Guide

[26] **SESAM/SQL-Server** (BS2000/OSD)
Database Operation
User Guide

[27] **SESAM/SQL-Server** (BS2000/OSD)
Performance
User Guide

[28] **SHC-OSD / SCCA-BS2**
Storage Management for BS2000/OSD
User Guide

[29] **SM2-PA** (BS2000/OSD)
SM2 Program Analyzer
User Guide

[30] **SPACEOPT** (BS2000/OSD)
Disk Optimization and Reorganization
User Guide

[31] **BS2000/OSD-BC**
System Installation
User Guide

[32] **BS2000/OSD-BC**
system-managed storage
User Guide

[33] **VM2000** (BS2000/OSD)
Virtual Machine System
User Guide

Index