



An Oracle White Paper  
June 2013

# Oracle Real Application Clusters (RAC)

Executive Overview .....	1
Oracle Real Application Clusters – Overview.....	2
Business Continuity and High Availability .....	5
Scalability and Agility.....	6
Cost-effective Workload Management .....	7
Standardized Deployment and System Management .....	11
Oracle Real Application Clusters New Features .....	12
Oracle RAC Support for Oracle Multitenant .....	12
Application Continuity and Transaction Guard .....	13
Oracle Flex ASM .....	15
What-If Command Evaluation .....	17
Policy-managed Database and Cluster Management .....	18
Summary and Conclusion.....	20

## Executive Overview

Business continuity, high availability, scalability, flexibility and agility combined with easy management are the pillars of successful IT infrastructure and cloud deployments. The Oracle Database with Real Application Clusters (RAC) has been the solution of choice for thousands of Oracle customers for their database systems for more than a decade. Oracle Real Application Cluster 12c continues this success, providing significant enhancements in all of the areas that matter to your business success.

Originally focused on providing improved database services, Oracle RAC has evolved over the years and is now based on a comprehensive high availability (HA) stack that can be used as the foundation of a database cloud system as well as a shared infrastructure that can ensure high availability, scalability, flexibility and agility for any application in your data center.

This paper describes the benefits of Oracle RAC and discusses key enhancements in each layer of the latest version of the Oracle RAC stack contributing to an even higher level of application availability using an Oracle RAC database. Some of the key benefits of Oracle RAC being discussed in this paper include, but are not limited to:

- Business Continuity and High Availability
- Scalability and Agility
- Cost-effective Workload Management
- Standardized Deployment and System Management

## Oracle Real Application Clusters – Overview

The Oracle Database with the Oracle Real Application Clusters (RAC) option allows running multiple database instances on different servers in the cluster against a shared set of data files, also known as the database. The database spans multiple hardware systems and yet appears as a single unified database to the application. This enables the utilization of commodity hardware to reduce total cost of ownership and to provide a scalable computing environment that supports various application workloads. Oracle RAC is Oracle's premier shared disk database clustering technology.

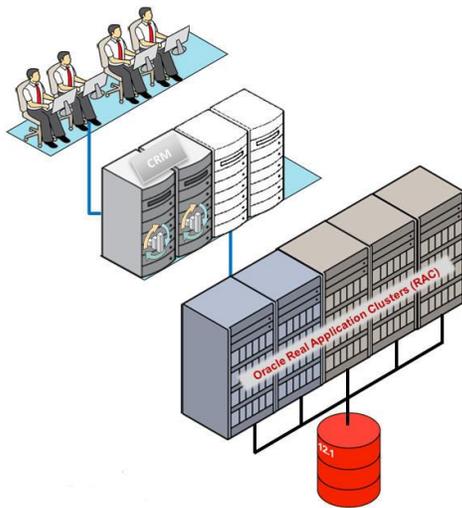


Figure 1: Oracle RAC 12c and Oracle RAC Stack Overview

### The Oracle RAC Stack

An Oracle RAC database requires Oracle Clusterware to be installed on the system prior to installing an Oracle RAC enabled database home. Oracle Clusterware (OCW) is an integral component of the Oracle Grid Infrastructure (GI) suite of products, which includes Automatic Storage Management (ASM) and the Oracle Cloud File System (CloudFS).

Oracle Grid Infrastructure, including Oracle ASM / CloudFS and Oracle Clusterware, and the Oracle Database with the Oracle Real Application Clusters (RAC) option constitute the Oracle RAC Stack.

Oracle Grid Infrastructure is the foundation for an Oracle RAC database system using Oracle Clusterware for automatic resource placement, adjustment and management and Oracle Automatic Storage Management for efficient and reliable storage management.

## Oracle Clusterware

Oracle Clusterware is a technology transforming a server farm into a cluster. A cluster is defined as a group of independent, but connected servers, cooperating as a single system. Oracle Clusterware is the intelligence in this system that provides the cooperation.

Oracle Clusterware was introduced with Oracle Database 10g Release 1 as the underlying clustering software required for running Oracle Real Application Clusters (RAC). As part of the Oracle RAC Stack, Oracle Clusterware is also used by the clustered version of Oracle ASM and is tightly integrated into the Oracle RAC Stack.

Oracle Clusterware is a complete, free of charge clustering solution that can be used outside of Oracle RAC. In such environments Oracle Clusterware is commonly used for automatic resource placement, adjustment and management of any kind of application. In both environments, Oracle Clusterware is responsible for maintaining node membership and ensuring fencing.

### New In Oracle Clusterware 12c

Depending on cluster deployment and management requirements, **Oracle Clusterware 12c can be installed in one of three ways:** 1) Standard Cluster (typical Oracle RAC cluster, configured like pre-Oracle Clusterware 12c versions) 2) Oracle Flex Cluster (a new cluster topology available with Oracle Clusterware 12c that combines traditional, tightly coupled and new loosely coupled servers in a single cluster) 3) Application Cluster (an optimized deployment customized for non-database applications)<sup>1</sup>.

Oracle Flex Cluster is an innovative multi-tier HA and global resource management solution for business critical end-to-end cloud computing.

## Oracle Automatic Storage Management (ASM)

Since Oracle RAC utilizes a shared disk architecture, the volume management and file system used for storing database data must be cluster-aware. Oracle Automatic Storage Management (ASM) is the recommended (clustered) volume manager for the Oracle Database.

Oracle ASM manages all data: Oracle database files, Oracle Clusterware files, and non-structured data such as binaries, external files and text files (via Oracle CloudFS). Through its low cost, easy administration and high performance characteristics, Oracle ASM is the storage technology of choice for managing storage for Oracle databases.

---

<sup>1</sup> For more information about Oracle Clusterware, see [www.oracle.com/goto/clusterware](http://www.oracle.com/goto/clusterware)

For performance and high availability, Oracle ASM follows the principle of “stripe and mirror everything”. Intelligent mirroring capabilities allow administrators to define 2- or 3-way mirrors to protect vital data. When a read operation identifies a corrupt block on a disk, Oracle ASM automatically relocates the valid block from the mirrored copy to an uncorrupted portion of the disk.<sup>2</sup>

#### **New In Oracle ASM 12c**

Oracle ASM 12c introduces Oracle Flex ASM, a new Oracle ASM deployment model that increases database instance availability and reduces the Oracle ASM related resource consumption. Oracle Flex ASM facilitates cluster based database consolidation, as it ensures that Oracle Database 12c instances running on a particular server will continue to operate, should the Oracle Flex ASM instance on that server fail.

#### **Oracle Cloud File System (CloudFS)**

The Oracle Cloud File System simplifies and automates storage management functions and increases storage utilization, uptime and agility to deliver predictable performance and availability for all general purpose and database files. CloudFS includes a cluster volume manager and a cluster file system:

- Oracle ASM Dynamic Volume Manager (ADVM) is the general purpose cluster volume manager for Oracle ACFS as well as 3rd party file systems.
- Oracle ASM Cluster File System (ACFS) is a general purpose POSIX and Windows compliant cluster file system with advanced data services such as snapshots, replication, encryption, security, etc.

#### **New In Oracle Cloud File System 12c**

With Oracle Grid Infrastructure 12c, the Oracle Cluster File System, also known as Oracle ACFS, is even more flexible. In addition to using Oracle ACFS to store unstructured data (configuration file, application specific data files), it can now be used for Oracle (RAC) Database files as well as shared database homes and backup sets.

---

<sup>2</sup> For information about Oracle ASM and the Oracle Cloud File System, see [www.oracle.com/goto/asm](http://www.oracle.com/goto/asm)

## Business Continuity and High Availability

Oracle Real Application Clusters (RAC) is the foundation for data center high availability (HA). Based on Oracle Grid Infrastructure as its management framework that provides the means for planned and unplanned as well as runtime and capacity-on-demand management, the Oracle RAC Stack ensures uninterrupted data center operations for applications of any kind.

Oracle RAC is an integral component of Oracle's Maximum Availability Architecture, which provides best practices to provide the highest availability for your data center. In this context, one of the key advantages of Oracle RAC is the inherent fault tolerance provided by multiple physical servers used by the database. Since the servers in the cluster run independently, the failure of one or more servers does not affect others. This architecture also allows a group of servers to be transparently put online or offline, while the rest of the system continues to provide uninterrupted database services.

Oracle RAC provides the following key characteristics, essential for HA data management:

- *Reliability* – The Oracle Database is known for its reliability. Oracle RAC takes this a step further by removing the database server as a single point of failure. If an instance fails, the remaining instances in the cluster remain open and active. Oracle Clusterware monitors all Oracle processes and immediately restarts any failed component.
- *Error Detection* – Oracle Clusterware automatically monitors Oracle RAC databases as well as other Oracle processes (Oracle ASM, instances, Listeners, etc.) and provides fast detection of problems in the environment. It also automatically recovers from failures often before users notice that a failure has occurred.
- *Recoverability* – The Oracle Database includes many features that make it easy to recover from various types of failures. If an instance fails in an Oracle RAC database, it is recognized by another instance in the cluster and recovery will start automatically. Fast Application Notification (FAN) and Fast Connection Failover (FCF) and especially the Oracle RAC 12c Application Continuity feature make it easy to mask any component failure from the user.
- *Continuous Operations* – Oracle RAC provides continuous service for both planned and unplanned outages. If a server (or an instance) fails, the database remains open and applications continue to be able to access data, allowing for business critical workloads to finish, mostly without a delay in service delivery.
  - Most database maintenance operations can be completed without downtime and are transparent to the user. Many maintenance tasks can be performed in a rolling fashion so application downtime is minimized or eliminated.

### New In Oracle Database 12c

New with Oracle Database 12c is the ability to contribute to the success of your business by ensuring that business critical applications continue to operate despite failures on system level.

**Application Continuity (AC)** is a new technology that protects applications from instance and session failures by re-playing affected “in-flight” transactions on another database instance.

Application Continuity uses the new **Oracle Database 12c Transaction Guard** feature that ensures that applications always get a “known result” (expected result) back from the database despite a failure on the database level or user induced “misoperation” such as using a double-click on purchase or the browser’s “back button” while checking out on a web-based order application. Such operations could otherwise result in unexpected outcome. Application Continuity requires certain Oracle Database 12c client access driver, while the Transaction Guard API can be used natively by various applications.

## Scalability and Agility

Oracle Real Application Clusters provides a unique technology for scaling applications. Traditionally, when database servers ran out of capacity, they were replaced with new and larger servers. As servers grow in capacity, they are more expensive.

For databases using Oracle RAC, there are alternatives for increasing the capacity. Applications that have traditionally run on large SMP servers can be migrated to run on pools of small servers in a cluster. Alternatively, the investment in the current hardware can be maintained and new servers can be added to the pool (or can constitute a new pool) without downtime.

While all servers in the cluster must run the same operating system and the same version of Oracle, they do not have to be of the same capacity. Oracle RAC users today often run server pools that fit their needs, using servers of (slightly) different characteristics.

Using the enhanced server pool usage and definition features with Oracle Database 12c, servers of different capacity can actively host workloads of different importance, ensuring that critical Service Level Agreements (SLA) can be met even during times of demand fluctuation. This way, the Oracle RAC architecture automatically accommodates rapidly changing business requirements and the resulting workload changes without the need for user intervention.

In order to dynamically allocate application workload, application users or mid-tier application server clients should connect to an Oracle Database using Dynamic Database Services. Using Dynamic Database Services enables Oracle to automatically balance user load among the multiple servers in a server pool by allocating services that represent user workload to one or more database instances.

This provides the Database Administrator (DBA) with the flexibility of choosing whether specific application workload is allocated on all or a subnet of the databases instances running in the cluster. It also enables administrators to painlessly add processing capacity as application requirements grow.

The Oracle Cache Fusion technology integrated into the Oracle RAC database immediately utilizes the CPU and memory resources of the new node(s) automatically without any need to manually re-partition the data ever.

## Cost-effective Workload Management

Oracle RAC includes innovative technologies to manage workloads in the cluster while providing the best application throughput given the configuration and high availability for the application. Oracle RAC is therefore the only database solution providing performance availability on the market today.

Performance availability, in addition to high availability and performance management, includes the ability to ensure that databases services are available to the application without a noticeable change in response time or overall quality.

Various components integrated in the Oracle RAC Stack contribute to its ability to provide performance availability for the Oracle Database. Unlike for other solutions on the market, those components and functionality are part of the stack and can be used at no extra costs.

### Introduction to Server Pools

Oracle Grid Infrastructure 11g Release 2 introduced server pools as a method to logically apportion a cluster into groups of servers offering database or application services. The scalability and availability of those databases and applications are controlled by server pool properties. Each server pool can be configured with a minimum and maximum size to govern scalability. The allocation and availability of resources hosted in server pools can be managed between pools by configuring the importance value.

Oracle Clusterware will allocate servers to user defined pools when a cluster reconfiguration takes place. Oracle Clusterware will allocate servers in order of importance. The number of instances maintained for the database is defined by the cardinality of the server pool.

### Introduction to (Dynamic Database) Services

Workload Management relies on the use of Services, a feature of the Oracle Database. Services hide the complexity of an Oracle RAC database by providing a single system image to manage workload.

In order to manage workloads or a group of applications, Dynamic Database Services can be defined and assigned to a particular application or to a subset of an application's operations. This allows the breakup of workloads from applications into manageable components based on business requirements such as service levels and priorities. For example, online users can use one service, while batch processing can use another and reporting can use yet another service to connect to the same database.

A service can span one or more instances of an Oracle database and an instance can support multiple services. The number of instances offering a service can either be managed dynamically by the DBA or automatically using policy-managed databases, independently of the application. When outages occur, services are automatically restored to surviving instances.

Services are integrated with many features of the Oracle Database. Application users for example can automatically be assigned to a Resource Manager consumer group to limit their consumed resources such as CPU, depending on a service. Batch jobs can be assigned to a specific job class based on their

service. The use of services also achieves location transparency for queues when using Oracle Streams Advanced Queuing, while inter-node parallel query can be restricted to instances hosting the service.

Oracle Database Quality of Service Management can be used to recommend/implement promotion of a service to a different consumer group in the Oracle Database Resource Manager plan, for example, to give the workload using that service more frequent access to the CPU resources.

### **Introduction to SCAN**

Single Client Access Name (SCAN) acts as a cluster alias for databases in the cluster and was introduced with Oracle Grid Infrastructure 11g Release 2. The benefit is that the client's connect information does not need to change if you add or remove nodes in the cluster. Having a single name to access the cluster allows applications to use EZConnect compatible clients and a simple JDBC thin URL to access any database running in the cluster, regardless on which server(s) in the cluster the database is concurrently running.

### **Connection Load Balancing**

Oracle Services provide connection load balancing for database connections on Oracle Net level. Client-side load balancing, which balances connection requests across all SCAN listeners in the cluster, is achieved by using the SCAN on the address list of the client connect string. SQL\*NET will randomly select one of the SCAN IP addresses. If the server chosen is not available, the next server in the list is tried.

Server-side load balancing is achieved using the SCAN listener. Each SCAN listener is aware of all instances in the cluster providing each service. Based on the goal defined for the service, the listener chooses the instance that will best meet the goal and the connection is routed to that instance through the local listener.

### **Fast Application Notification**

Fast Application Notification (FAN) provides integration between an Oracle RAC database and the application. It allows the application to be aware of the current configuration of the server pools at any given time so that application connections are made only to those instances that are currently able to respond to the application requests.

The Oracle RAC Stack HA framework immediately posts a FAN event when a state change occurs within the cluster. Integrated clients receive these events and can immediately react. For DOWN events, application interruption is minimized by cleaning up connections to the failed instance, in-flight transactions are interrupted with an error and returned to the application.

The new Application Continuity feature uses FAN to enable the replay, in a non-disruptive and rapid manner, of a request against the database after a recoverable error that makes the database session

unavailable. The request can contain transactional and non-transactional calls to the database. After a successful replay, the application can continue where the database session left off.

For UP events, new connections are created to allow the application to immediately take advantage of the extra resources available. Oracle JDBC, the Oracle Universal Connection Pool, ODP.NET and OCI clients are integrated with FAN. Applications with their own connection pools can take advantage of FAN by using either the Oracle RAC FAN API available with the Oracle Database 11g Release 2 JDBC driver or the Oracle Call Interface callback functionality. Generally, Oracle recommends using the Oracle Universal Connection Pool as the standard pool for connecting to an Oracle Database.

### **Load Balancing Advisory**

Load balancing distributes work across all of the available Oracle RAC database instances. Oracle recommends that applications use connection pools with persistent connections that span the instances that offer a particular service. When using persistent connections, connections are created infrequently and exist for a long duration. Work comes into the system with high frequency, borrows these connections, and exists for a relatively short duration.

The load balancing advisory provides advice about how to direct incoming work to the instances that will provide the optimal quality of service for that work. This minimizes the need to relocate the work later.

By using the Load Balancing Advisory and runtime connection load balancing goals, feedback is built into the system. Work is routed to provide the best service times globally, and routing responds gracefully to changing system conditions. In a steady state, the system approaches equilibrium with improved throughput across all of the Oracle RAC instances.

The load balancing advisory is deployed with key Oracle clients, such as the listener, the Universal Connection Pool, Oracle WebLogic Server Active GridLink for Oracle RAC, and the ODP.NET Connection Pools. Third-party applications can also subscribe to load balancing advisory events by using JDBC and Oracle RAC FAN API or by using callbacks with the Oracle Call Interface (OCI).

### **New In Oracle Database 12c**

With Oracle Grid Infrastructure 12c SCAN has been enhanced in order to 1) support IPv6 based IP addresses, 2) support multiple subnets in the cluster and 3) to restrict Service Registration.<sup>3</sup>

Starting with Oracle Grid Infrastructure 12c, Cluster nodes can be configured to use either IPv4 or IPv6 based IP addresses for the Virtual IPs (VIP) on the public network, while more than one public

---

<sup>3</sup> For information about SCAN, see [www.oracle.com/technetwork/products/clustering/overview/scan-129069.pdf](http://www.oracle.com/technetwork/products/clustering/overview/scan-129069.pdf)

network can be defined for the cluster. Database clients and applications can connect to either IPv4 or IPv6 VIP addresses. The SCAN listener automatically redirects client connects to the appropriate database listener within a given subnet considering the IP protocol requested by the client, while SCAN listeners can be defined for each subnet in the cluster.

Listeners managed by Oracle Grid Infrastructure can be configured to restrict clients from accessing a database registered with a listener using various conditions; for example, the subnet from which these clients are connecting. The same feature can be used to restrict database instance registration with a particular (SCAN) listener as part of the dynamic instance registration to prevent an accidental registration of a database instance with listeners in a shared environment.

**Effective with Oracle Database 12c, Oracle Database Quality of Service (QoS) Management is part of the standard Oracle RAC license** and can be used for the policy-based management of all Oracle RAC databases that support server pools (Oracle Grid Infrastructure and Oracle Database 11g Release 2 and higher required).

Oracle Database Quality of Service Management is an automated, policy-based feature that monitors the workload requests for an entire system, manages the resources that are shared across applications and adjusts the system configuration to keep the applications running at the required performance levels. It responds gracefully to changes in system configuration and demand, thus avoiding additional oscillations in application performance levels. <sup>4</sup>

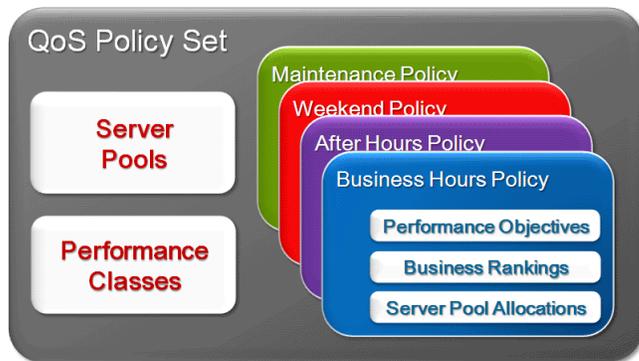


Figure 2: Oracle Quality of Service (QoS) Management Overview

<sup>4</sup> For more information about Quality of Service (QoS) Management, see: <http://www.oracle.com/technetwork/products/clustering/overview/qosmanagement-508184.html>

## Standardized Deployment and System Management

The key management strength of Oracle RAC is the efficient management of workload in the cluster, for which it utilizes Oracle Grid Infrastructure as its management framework that provides the means for planned and unplanned as well as runtime and capacity-on-demand management.

Oracle RAC uses the Oracle Grid Infrastructure to **support two types of management:**

1. Administrator-managed Oracle RAC databases are the traditionally supported way of managing Oracle RAC databases, in which instances are assigned to servers in the cluster, mostly used in dedicated database systems.
2. Policy-managed Oracle RAC databases allow for more management flexibility by abstracting fixed resource allocation (e.g. assigning instances to run only on specific servers in the cluster).

Policy-managed databases use server pools, a logical partitioning of clusters, and Dynamic Database Services to flexibly allocate and automatically manage workloads across one or many databases in the cluster. Policy management automatically adjusts resource allocation for critical workloads based on pre-defined user input. These policies drive resource allocation and capacity adjustment in the event of planned or unplanned outages, changing demand or for programmed cluster re-configurations.

Oracle Database Quality of Service (QoS) Management uses server pools to create logical pools of servers within a cluster to provide workload isolation. Oracle Database QoS Management can recommend/implement moving a server from one server pool to another based on measured and projected demand, to restore or to maintain the Performance Objectives currently in effect.

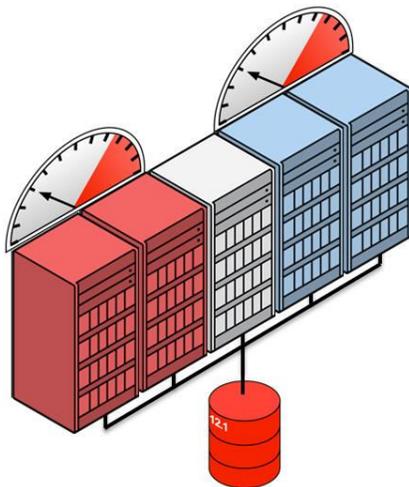


Figure 3: Policy-managed Oracle RAC databases for more flexibility and better HA

## Oracle Real Application Clusters New Features

### Oracle RAC Support for Oracle Multitenant

Oracle Multitenant is a new option to the Oracle Database 12c Enterprise Edition that helps customers reduce IT costs by simplifying consolidation, provisioning, upgrades, and more. It is based on a new architecture that allows a multitenant container database to hold many pluggable databases (PDBs). The idea is that an existing database can be simply adopted, with no changes in the application tier, as a pluggable database. In this architecture, Oracle RAC provides the local high availability that is required when consolidating various business critical applications on one system.

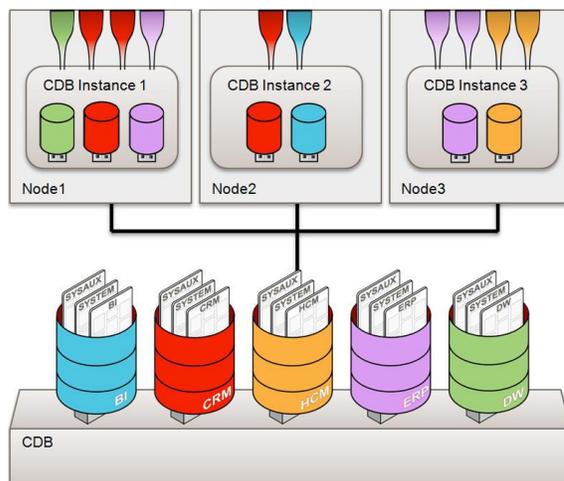


Figure 4: Oracle RAC for PDB Support - Overview

When using PDBs with Oracle RAC, the multitenant container database (CDB) is based on Oracle RAC. Each pluggable database can be made available on either every instance of the RAC CDB or a subset of instances. In any case, access to and management of the PDBs are regulated using Dynamic Database Services, which will also be used by applications to connect to the respective PDB – as they would in a Single Instance Oracle Database using Oracle Net Services for connectivity.

## Application Continuity and Transaction Guard

Application Continuity (AC) is a new technology that protects applications from instance and session failures by re-playing affected “in-flight” transactions on another database instance in the cluster.

It enables the replay, in a non-disruptive and rapid manner, of a request against the database after a recoverable error that makes the database session unavailable. The request can contain transactional and non-transactional calls to the database. After a successful replay, the application can continue where the database session left off.

This benefits users, developers and administrators at the same time. Users are not left in doubt whether their funds transfers, flight bookings or other transactions have gone through successfully, while administrators do not need to reboot mid-tier machines to recover from logon storms.

Transaction Guard is a reliable protocol and tool that returns the outcome of the last in-flight transaction after an outage that makes the database session unavailable. Without Transaction Guard, applications and users who attempt to retry operations following an outage can cause logical corruption by committing duplicate transactions or committing transactions out of order.

Transaction Guard prevents the costs of ambiguous errors that lead to user frustration, customer support calls, and lost opportunities. Transaction Guard is safer and performs better, with lower overhead, than homegrown solutions striving for a known outcome and at-most-once execution.

The loss of a database session can affect the user in many ways. For example:

- *Confusion* – Users do not know what happened to their application’s requests like funds transfers, orders, payments, bookings and so forth.
- *Usability* – User may lose uncommitted data, needing to log in again and re-enter or resubmit their data.
- *Disruption* – As a consequence of a failure, administrator may have to reboot mid-tier machines to handle the load balance and incoming logon storm, leading to yet another outage experience for the user.

### How it works – Overview

The fundamental assumption for Application Continuity to work and make it unnecessary for the application to maintain individual error handling is that the failure encountered represents a “recoverable error”. A recoverable error is defined as follows:

- *A recoverable error* is an error that arises due to an external system failure, independent of the application session logic that is executed. Recoverable errors can occur following planned or unplanned outages of foreground processes, network components, servers, storage components or database instances. Subsequent to a recoverable error, the application typically receives an error code that can leave the application not knowing the status of the last operation submitted. In such cases, AC is invoked making it unnecessary for an application to maintain error numbers in their code in order to react to such respective failures individually.

Given the basic assumption, applications must continue to include error handling for these cases:

1. Non-recoverable errors, such as invalid input data.
2. Recoverable errors when replay has encountered one of the documented restrictions.

### Summary – Application Continuity and Transaction Guard

If the database session becomes unavailable due to a recoverable error, Application Continuity attempts to rebuild the session and any open transactions to the correct states. If the transaction is successful and does not need to be re-executed, the successful status is returned to the application.

If the replay is successful, the request continues safely without duplication.

If the replay is not successful, the database rejects the replay and the application receives the original error. To be successful, the replay must return to the client the exact same data that the client would have received previously in the request, and that the application potentially made a decision on.<sup>5</sup>

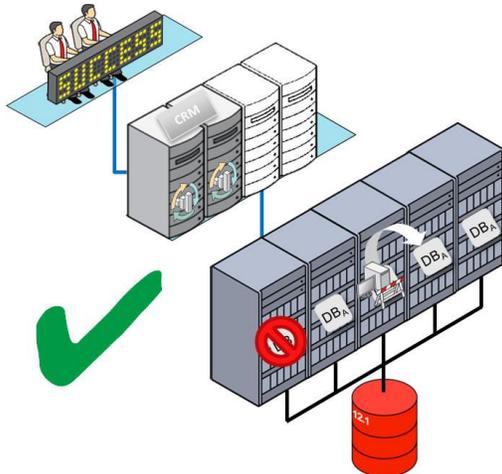


Figure 5: User Experience *with* Application Continuity

<sup>5</sup> For more information about Application Continuity, see:  
<http://www.oracle.com/technetwork/products/clustering/ac-overview-1967264.html>

## Oracle Flex ASM

Oracle Flex ASM is a new Oracle Automatic Storage Management (ASM) deployment model that increases database instance availability and reduces the Oracle ASM related resource consumption. Oracle Flex ASM facilitates cluster based database consolidation, as it ensures that Oracle Database 12c instances running on a particular server will continue to operate, should the Oracle Flex ASM instance on that server fail.

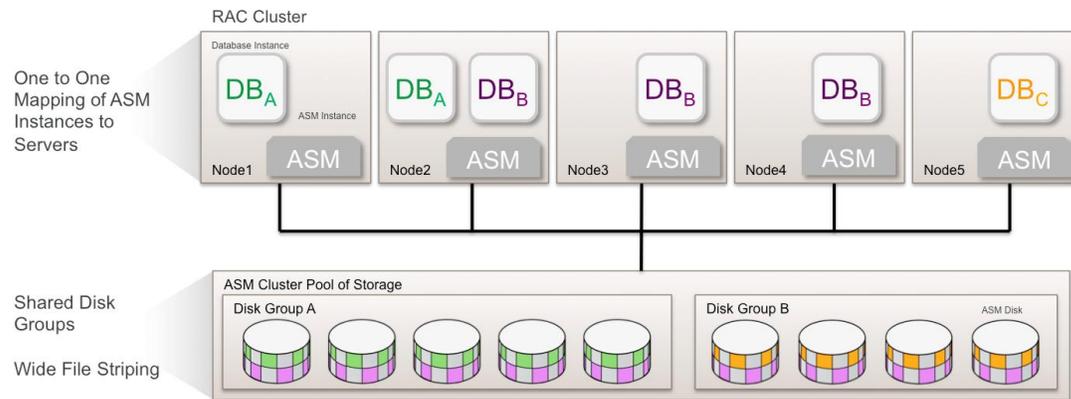


Figure 6: Oracle ASM before Oracle Database 12c

Before Oracle Database 12c, an Oracle ASM instance was required to run on every node in the cluster. Database instances would connect locally, on the same server, to the Oracle ASM instance using an OS based authentication. A failure of the local Oracle ASM instance would lead to a failure of all of the database instances running on this server.

As a result of the new Oracle RAC 12c strategy to allow enablement of new features as demand requires it, no change is enforced when upgrading to Oracle ASM 12c and Oracle ASM can be used as with previous versions with a configuration as shown in figure 6.

In order to fully benefit from the new Oracle ASM capabilities, Oracle Flex ASM needs to be enabled. Oracle Flex ASM can be enabled either during the initial installation of Oracle Grid Infrastructure 12c or as a post installation step. Oracle Flex ASM needs to be enabled for the whole cluster.

Once Oracle Flex ASM is enabled the most noticeable effect is that the one-to-one mapping of Oracle ASM instances to servers in the cluster is removed and clusters with three or more servers will host a set of only three Oracle Flex ASM instances floating in the cluster (as cluster resources). Two server clusters will maintain a one-to-one mapping.

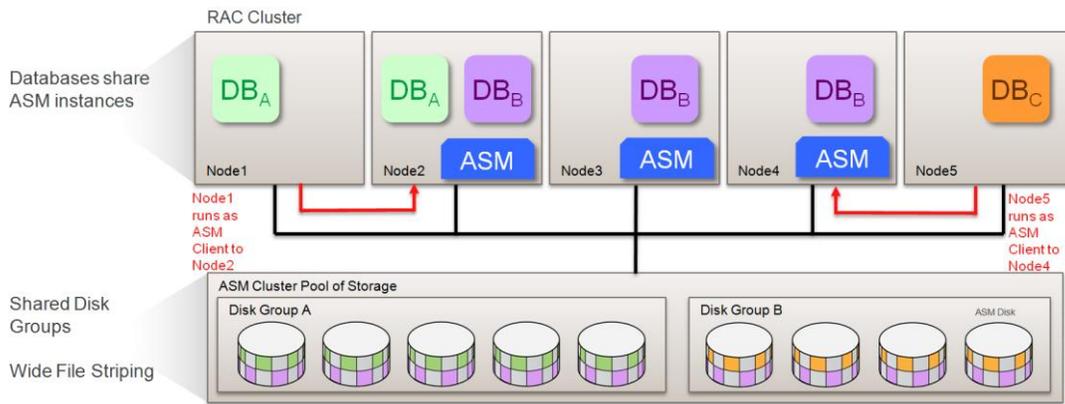


Figure 7: Oracle Flex ASM is enabled - one-to-one-mapping of instances to nodes in the cluster removed

Enabling Oracle Flex ASM also enables an inter-node Oracle ASM instance failover, which together with Oracle Database 12c and later databases ensures continuous database operation despite the absence of an Oracle ASM instance on the local node.

Oracle Flex ASM can protect database instances on a particular server from failures (unplanned) as well as planned downtime due to scheduled maintenance such as applying OS or rolling database patches. Especially in consolidated environments with multiple database instances running on a server, the independence of a database instance from a local Oracle ASM instance significantly improves database availability.

**How it works – Overview**

If an Oracle Flex ASM instance fails on a particular node, the Oracle Flex ASM instance is failed over to another node in the cluster, while the Oracle Database 12c or later instances will continue to operate, using a non-local Flex ASM instance in the cluster.

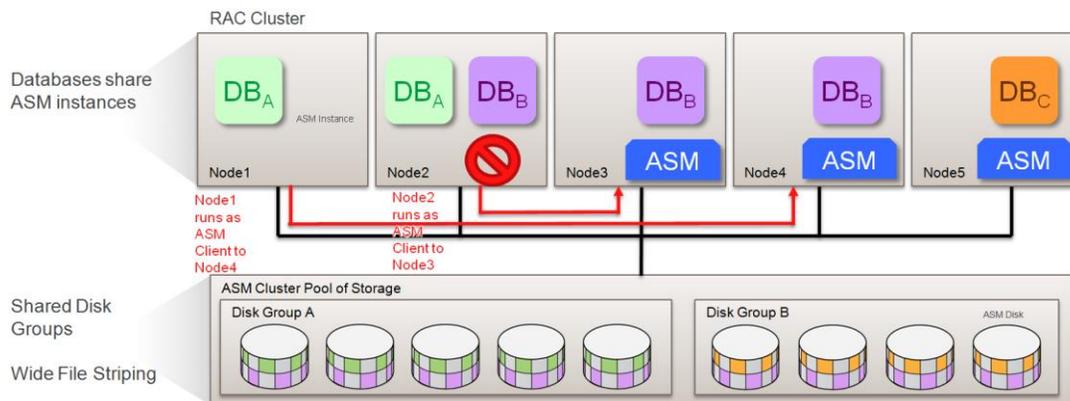


Figure 8: Oracle Database instances continue to operate after a local ASM instance failure

## What-If Command Evaluation

Understanding resource placement prior to an event such as a server outage or an online relocation of resources is critical when managing complex and dynamic environments. The What-If Command Evaluation feature facilitates the management of today's dynamic infrastructure environments.

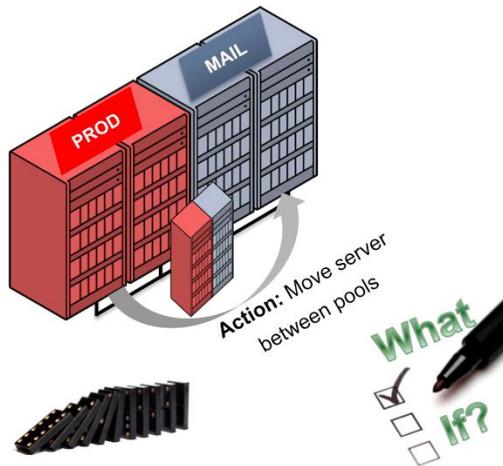


Figure 9: What-If Command Evaluation

As the name indicates, the What-If Command Evaluation feature is designed to evaluate the impact of a command executed against the cluster prior to its execution. There are two levels of the What-If Command Evaluation: 1) a view for cluster administrators 2) a view for database administrators. The difference between the two – other than the fact that they are provided by either CRSCCTL (for the cluster administrator) or SRVCTL (for the database administrator) – is their information detail.

Cluster administrators are expected to manage the cluster as a whole, overseeing all applications being hosted in the cluster, while database administrators will most likely want to know the impact of a command on the database(s) that they are responsible for. In addition, the commands required to perform daily or occasional management tasks differ for each group of administrators, requiring the same functionality to be available for each command. The CRSCCTL command tool also provides a setting to influence the level of details provided as part of the result output.

The What-If Command Evaluation can be used with different command options for the SRVCTL and CRSCCTL tools, allowing for nearly all management scenarios to be evaluated prior to actually executing them. Special commands (e.g. SRVCTL predict) or command options (e.g. the CRSCCTL command option `-fail`) can be even used to predict the impact of a component failure.

The What-If Command Evaluation does not provide transactional security. This means that the evaluation is performed on a given state of the cluster at the moment the evaluation is started. The result will consider this state. Assuming that the cluster state remains unchanged between the start of the evaluation and the actual execution of the command, the result will be as projected by What-If. If the cluster state changes, the result of the actual execution may be different than its projection.

## Policy-managed Database and Cluster Management

Policy-managed database and cluster management is instrumental when it comes to managing highly available (database) systems in dynamic environments. Policy-managed environments allow for an automatic and dynamic adjustment of a system due to changes in demand. They also ensure service availability on systems that have been subject to multiple failures in a row. As most applications cannot distinguish whether a change in the cluster is of accidental nature or a result of a planned maintenance operation, policy-based management can be used to improve planned and unplanned operations.

### Enhanced Server Pool Usage and Definition with Oracle Grid Infrastructure 12c

Oracle Grid Infrastructure 12c enhances the use of server pools by introducing server attributes stored on a per-server basis and server category-based server pool definitions. Using server attributes, Oracle Grid Infrastructure 12c understands and actively uses the concept of heterogeneous servers in a cluster by storing pre-defined as well as user-defined server attributes on a per server basis.

Server attributes are automatically determined internally on startup of the Oracle Clusterware stack on a particular server and stored persistently until the stack on the server is restarted. The default set of server attributes (user-defined attributes can be added) includes, but is not limited to “SERVER\_NAME”, “MEMORY\_SIZE” and “CPU\_COUNT”.

Server attributes can then be used by server categories to define server pools. Server categories define a server pool based on attributes rather than the sole number or name of a server that shall be part of the pool. Server categories use regular expressions to define what type of servers can be used in a particular server pool. This allows for the definition of entry level requirements for a server to join a server pool and thereby can be used to ensure that all servers in a server pool provide a minimum capacity to run the workload assigned to this server pool via a Dynamic Database Service.

It needs to be noted that the Oracle RAC Stack has traditionally allowed for servers of different sizes (i.e. capabilities) to be included in the same cluster. The basic cluster management assumption, however, was that servers in the cluster were similar in capability. It was left to the user to map workload to servers of differing compute power or other specific attributes accordingly. This type of manual resource management is not only a burden to the administrator; it is inherently error prone when considering capacity requirements in the event of outages. Server pool definitions using server categories based on physical or user defined server attributes provide an automated and efficient way to manage environments with varying workload requirements and with servers of varying capacities.

### Using Policy-Managed Environments for Scalability and Agility

One of the main use cases for policy-managed database is the automatic adjustment of database environments to changes in demand. Another use case, especially with Oracle Grid Infrastructure 12c, includes non-database applications (policy-managed cluster management) using policy sets.

A change in demand can be triggered by an anticipated or an unanticipated increase in demand for one or more applications hosted in the cluster<sup>6</sup>. It can, however, also be a result of a planned maintenance operation (a server needs to be taken down for maintenance) as well as a re-occurring adjustment of the cluster to accommodate for 24x7 operations across different geographic regions and time zones or to meet end-of-month or end-of-year changes in demand. Policy-managed environments provide the infrastructure to automatically handle all of those scenarios.

### **Using Policy-Managed Environments for Better High Availability**

Policy-managed environments are the easiest way to adequately and automatically accommodate for multiple failure scenarios in the cluster. Administrator-managed systems, in which resources such as database instances and Dynamic Database Services are statically assigned to servers in the cluster, handle outages by manually defining static “preferred” and “available” placement preferences.

In administrator-managed environments, resources are defined to preferably run on a certain instance (Dynamic Database Services are assigned to database instances) or servers (in case of the database instances) and only if this specific instance or server is not available, will the resource, e.g. database service, be hosted on an alternative instance or server in the cluster. There are no policy decisions, all servers are considered equal and the workload to server assignment is static and manually defined.

Policy-managed environments remove the burden and need for such static definitions. Using policies and policy sets, administrators can easily define server pools of different importance and map critical workloads to specific server pools of pre-established importance. Policies guarantee, based on available compute resources in the cluster, that the server pool of greater importance, and by extension the business critical workload assigned to this server pool, will get the compute resources necessary to satisfy performance or availability service level agreements.

Server categories ensure that server assignments to server pools will meet the required minimums. If a server outage arises, the importance of the workload will be considered using the server pool definition of importance and predictable policies will be applied to re-allocate the resources in the cluster accordingly. If the remaining cluster size is insufficient to host the required workload, the same policies will be used to potentially move compute resources of a server pool of lesser importance to the server pool of higher importance in order to satisfy the compute resource requirements of business critical workloads.

---

<sup>6</sup> Application monitoring using either Oracle Enterprise Manager 12c or Oracle Quality of Service (QoS) Management is required to automatically react to demand changes triggered by application demand.

## Summary and Conclusion

Business continuity, high availability, scalability, flexibility and agility combined with automatic management are the pillars of successful IT infrastructure and cloud deployments. The Oracle Database with the Real Application Clusters (RAC) option has been the solution of choice for thousands of Oracle customers for their business critical database systems for more than a decade.

Oracle Real Application Cluster 12c continues this success, providing significant enhancements in all of the areas that matter to your business success. New features such as Application Continuity and Oracle Flex ASM as well as noteworthy enhancements in the area of policy-managed database and cluster management as well as a comprehensive database consolidation and Oracle Multitenant support make Oracle RAC the database virtualization solution of choice for your IT infrastructure.

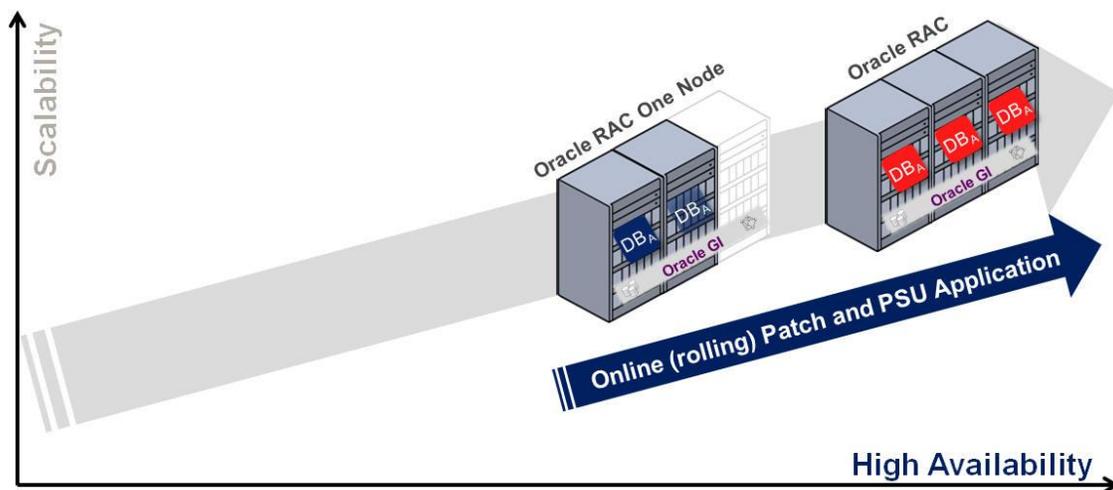


Figure 10: Standardize on Oracle RAC for better Scalability and HA



Oracle Real Application Clusters (RAC)  
June 2013

Author: Markus Michalewicz  
Contributing Authors:  
Burt Clouse, John McHugh

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200

oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0113

**Hardware and Software, Engineered to Work Together**