Universidade do Minho
Escola de Engenharia

Ricardo Miguel da Silva Neto Godinho

# Availability, Reliability and Scalability in Database Architecture

Junho de 2010

**Universidade do Minho**
Escola de Engenharia

Ricardo Miguel da Silva Neto Godinho

# Availability, Reliability and Scalability in Database Architecture

Thesis
Mestrado em Engenharia Informática

Under supervision of
**Professor Doutor José Machado**

Junho de 2010

É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA TESE APENAS PARA EFEITOS
DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE
COMPROMETE;

Universidade do Minho, ___/___/______

Assinatura: ________________________________________________

## Acknowledgements

Endereço os meus sinceros agradecimentos ao Professor José Machado, meu orientador na universidade, pela sua disponibilidade para prestar auxílio nos momentos mais difíceis, em que as dúvidas e as incertezas pairavam, durante a realização do mestrado, bem como na elaboração da dissertação.

Agradeço ainda aos meus Pais por me terem ensinado a pensar e a enfrentar as dificuldades ao longo de toda a minha vida e serem uma presença constante e estimuladora durante todo o mestrado.

Gostava também de deixar os meus agradecimentos aos meus colegas da Oramix, que, pela sua experiência na área da administração de base de dados, sempre se disposeram a prestar auxilio nas minhas dúvidas e nas tarefas mais complexas.

Finalmente, queria deixar uma palavra de agradecimento a todos os meus amigos que, de uma forma ou de outra, nunca deixaram de me apoiar e me deram força para chegar aqui.

**Obrigado a todos...**

## Resumo

Disponibilidade, Fiabilidade e Escalabilidade são propriedades essenciais num sistema de base de dados. Estes sistemas têm necessidades vitais em matéria de administração, devem desempenhar a melhor performance e estão preparados para evitar ou recuperar situações de desastre. A solução para que estes objectivos sejam atingidos passa pela escolha de uma arquitectura robusta e pela correcta configuração e integração das diversas ferramentas tecnológicas. Neste contexto, mostra-se que o Oracle 10gR2, sistema gestor de base de dados, oferece um conjunto de tecnologias para a resolução de problemas de operação, gestão e administração de dados em ambientes críticos e é um excelente candidato para o arquivo e processamento de informação em situações onde se exige disponibilidade e fiabilidade permanente da informação, assim como se pretende dotar os sistemas com funcionalidades cada vez mais potentes e actuais.

## Abstract

Availability, Reliability and Scalability are essential properties in a database management system. These systems have critical needs in terms of administration and must demonstrate better performance and be prepared to prevent or self-recover situation of disaster. The solution to the fulfillment of these requirements is to select a robust architecture, a correct configuration and the integration of several technological tools. In this context an architecture based on the Oracle 10gR2 Database engine will be proposed. This oracle suite offers a set of technologies to operational problem solving, as well as management and storage in critical environments. It is shown that it is a good candidate for archiving and data processing in situation where availability and reliability are mandatory and when it is essential to have more powerful and updated functionalities for users and administrators.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Context

Databases and the internet have enabled worldwide collaboration and information sharing by extending the reach of database applications throughout organizations and communities. Their significance and impact on information systems emphasize the importance of high availability in data management solutions. Small businesses and global enterprises have users all over the world who require access to data 24 hours a day. Without this data access, operations can stop and revenue is lost. Users, who have become more dependent upon most solutions, now demand service-level agreements from their Information Technology (IT) departments and solution providers. Increasingly, availability is measured in dollars or euros and not just in time and convenience.

Enterprises have been using their IT infrastructure to provide a competitive advantage. It increases productivity and empowers users to make faster and increasingly informed decisions. Although these systems present great advantages and an increasing dependence on these infrastructures constitute a risk that has to be dealt in order not to undermine the overall working process of a company. If a critical application becomes unavailable, then the entire business can be in danger. Revenue and customers can be lost, penalties can be owed, and bad publicity can have a lasting effect on customers and on company's stock price. Thereby, it is critical to examine the factors that determine how to protect your data and to maximize the availability to users.

## 1.2 Motivation and Objectives

Examining and addressing all the possible causes of downtime is one of the true challenges in designing a high availability solution. It is important to consider the main causes of planned and unplanned downtime.

Planned downtime can be just as disruptive to operations as unplanned downtime, especially in global enterprises that support users in multiple time zones, up to 24 hours per day. In this context, it is important to design a system to minimize planned interruptions. As shown by the schema in Figure 1.1, causes of planned downtime include routine operations, periodic maintenance and new deployments.

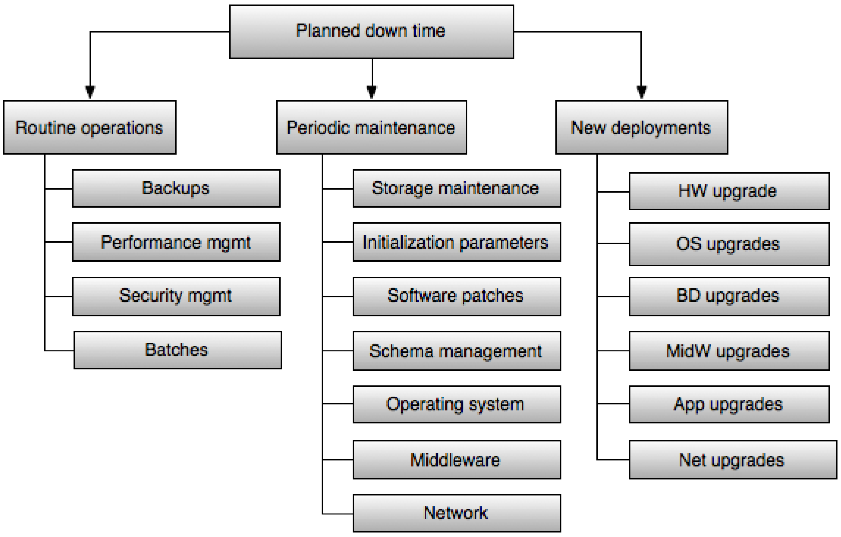

Figure 1.1: Planned Downtime

Moreover there are other causes that can interrupt systems operations. These causes are under the unplanned downtime and they include software failures, hardware failures, human error and disasters, as shown by the schema in Figure 1.2.

The purpose of this dissertation is to study the Oracle Database Suite, install each of these architectures as technical best practices and test each of them to be able to select the most suitable architecture for an organization, taking into consideration the underlying architecture as means towards improving the availability of the overall system. It intends to analyze the existing best practices and

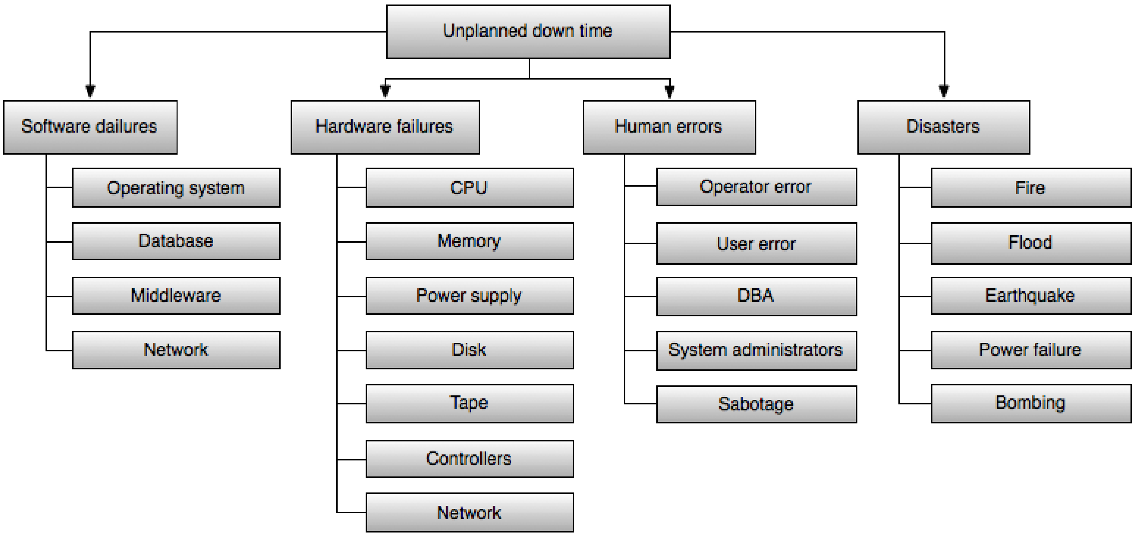

Figure 1.2: Unplanned Downtime

possible architecture taking in consideration Availability, Reliability and Scalability.

## 1.3 Structure of the document

After describing the context, the motivation and objectives of this work, it will be exposed a study of several features and architectures offered by Oracle 10gR2 that allows for the archive of a high availability solution.

Then we will find the evaluation where we can see a comparative study between the architectures and finally, I will finish with the conclusions and suggestions for future work.

# Chapter 2

# Technology for High Availability

Oracle Database 10gR2 offers an integrated suite of high availability solutions that aim to eliminate or minimize both planned and unplanned downtime.

In this suit it is included some RDBMS core features, such as Automatic Storage Management (ASM), Flash Recovery Area, Online Reorganisation and Redefinition, Recovery Manager (RMAN), Real Application Cluster (RAC) and Data Guard.

The following sections are dedicated to explore each of these features and products that are part of the Oracle Database 10gR2 Suite.

## 2.1 RDBMS features

A High availability environment starts at the bottom of the data architecture, inside the tables, indexes, constrains and packages. Oracle database 10gR2 offers several features to increase the availability, reliability and scalability. The follow sections are dedicated to those features inside the Oracle 10gR2 RDBMS.

### 2.1.1 Automatic Storage Management

In previous versions of the Oracle Database suite and in almost other competing databases, management of data files for large databases has been consuming a good portion of a Database Administrator's time. The number of data files in large databases could be of hundreds or even thousands. The Database Administrator must coordinated and provided names for these files and then optimize their storage.

Since Oracle Database 10g, storage management for the database has been very simplified with the usage of Automatic Storage Management (ASM). ASM provides a vertical integration of the file system and the volume manager that is specifically built for Oracle Database files. With this capability, ASM reduces the cost and complexity of storage management tasks, such as creating or laying out databases and disk space management.

With ASM the disk management can be done using familiar create/alter/drop SQL statements avoiding Database Administrators to learn new skills associated to a specific operative systems implementation or make crucial decisions on provisioning. They can also manage a dynamic database environment as it is possible to increase the database size without compromising availability to adjust storage allocation as ASM performs automatic online redistribution after the incremental addition or removal of storage capacity [1].

The ASM distributes input/output (I/O) load across all available resources to optimize performance and will remove the need for manual I/O tuning. It is able to provide management for single symmetric multiprocessing (SMP) machines or across multiple nodes for cluster for Oracle Real Application Clusters (RAC) support.

ASM can maintain redundant copies of data to provide fault tolerance, or it can be built on top of vendor-supplied storage mechanisms. Data management is done by selecting the desired reliability and performance characteristics for classes of data rather than with human interaction on a file oriented paradigm.

### ASM Instances

In Oracle Database 10g there are two types of instances: database and ASM instances. The ASM instance, which is normally named +ASM, is started with the INSTANCE_TYPE=ASM init.ora parameter. This parameter indicates the Oracle initialization routine to start as an ASM instance. Unlike the standard database instance, the ASM instance does not contain physical files, such as logfiles, controlfiles or datafiles, and only requires a few init.ora parameters for startup.

On startup, an ASM instance will start all the basic background processes, plus some new ones that are specific to the operation of ASM. The STARTUP clauses for ASM instances are similar to those for database instances. For example, RESTRICT prevents database in-

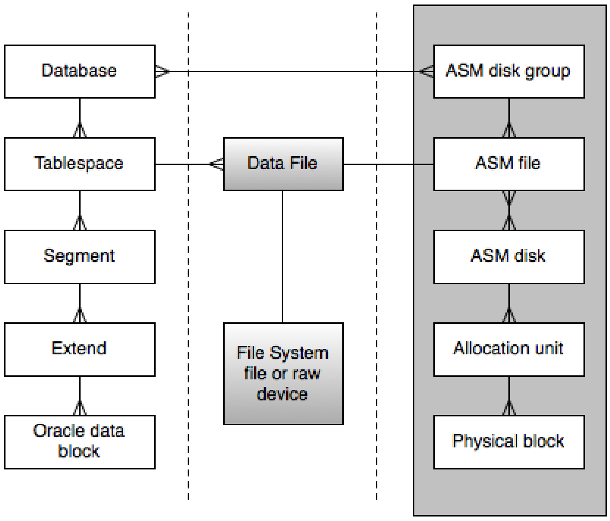

Figure 2.1: ASM Concepts (based on [6])

stances from connecting to this ASM instance, NOMOUNT starts up an ASM instance without mounting any disk group and the MOUNT will mount all defined diskgroups.

ASM is the volume manager for all databases that employ ASM on a given node. Therefore, only one ASM instance is required per node regardless the number of database instances on the node. Additionally, ASM works seamlessly with the Real Application Cluster (RAC) architecture to support clustered storage environments. In RAC environments, there will be one ASM instance per clustered node, and the ASM instances communicate with each other on a peer-to-peer basis using the interconnect [1].

### Concepts

ASM does not eliminate any pre-existing database functionalities. Existing databases are able to operate as they always have. It is possible to create new files as ASM files and leave existing files to be administered in the old way, or eventually to migrate them to ASM.

The diagram 2.1 depicts the relationships that exist between the various storage components inside an Oracle database that uses ASM. The left and middle parts of the diagram show the relationships that exist in previous releases. On the right there are the new concepts introduced by ASM.

At the top of this new architecture hierarchy are the ASM disk groups. Any single ASM file is contained in only one disk group. However, a disk group may contain files belonging to several databases, and a single database may use storage from multiple disk groups. One disk group is made up of multiple ASM disks, and each ASM disk belongs to only one disk group. ASM files are always spread across all the ASM disks in the disk group. ASM disks are partitioned in allocation units (AU) of one megabyte each. An allocation unit is the smallest contiguous disk space that ASM allocates. ASM does not allow an Oracle block to be split across allocation units [9].

### 2.1.2 Flash Recovery Area

The Flash Recovery Area is a specific location for all recovery-related files and activities in an Oracle Database. It is possible to set up the Flash Recovery Area to a directory on a normal disk volume or it can be an ASM disk group. After this feature is enabled, all RMAN backups, archive logs, control file auto backups, and datafile copies are automatically written to Flash Recovery Area and the management of this disk space is handled by Recovery Manager (RMAN) and the database server.

The Flash Recovery Area also works as a repository for mirrored copies of online redo log files, the block change tracking file and a current controlfile.

Making a backup to disk is faster because using the Flash Recovery Area eliminates the bottleneck of writing to tape. More importantly, if database media recovery is required, then datafile backups are readily available. Restoration and recovery time is reduced because they do not need to find a tape and a free tape device to restore the needed datafiles and archive logs [3].

### 2.1.3 Online Reorganization and Redefinition

Online Reorganization and Redefinition in Oracle Database 10gR2 is a very important feature to enhance availability and manageability. This feature offers to DBA's significant flexibility to modify the physical attributes of a table, index, advanced queues, clustered tables, materialized views while allowing users full access to the database.

When redefining a table, the table is locked in exclusive mode only during a very small window that is independent of the size of the table and of complexity of the redefinition, and that is completely transparent to users. With this feature, any physical attribute of the table can be changed online. The table can be moved to a new location, partitioned and converted from one organization (such as heap-organized) to another (such as index-organized). Many other logical attributes can also be changed. Column names, types and sizes can be changed almost instantly, as well as new columns can be added and existing ones can deleted or merged. One restriction however, is that the primary key of the table cannot be modified.

## 2.2 FlashBack

Flashback technology provides a set of features to view and rewind data back and forth in time. The Flashback features offer the capability to query past versions of schema objects, query historical data, perform update analysis and perform self-service repair to recover from logical corruption while the database is online.

To manage these new advances demonstrated by the Flashback tool, an SQL interface is displayed in order to quickly analyze and repair human errors, provide fine-grained analysis and repair localized damage such as deleting the wrong customer order. It also enables correction of more widespread damage, spending a short amount of time in the process to avoid a long downtime timespan. This technology is unique to the Oracle Database and supports recovery at all levels including row, transaction, table, tablespace, and database [3].

Several features are included on Flashback technology, such as Oracle Flashback Query, Oracle Flashback Versions Query, Oracle Flashback Transaction Query, Oracle Flashback Table, Oracle Flashback Drop, Oracle Flashback Database and Oracle Flashback Restore Points.

The following sub sections are dedicated to explain each of these features.

### 2.2.1 Oracle Flashback Query

Oracle Flashback Query provides the ability to view the data as it existed in the past by using the Automatic Undo Management system to obtain metadata and historical logs for transactions. Undo data is persistent and will survive a database malfunction or shutdown. Not only Flashback Query provides the ability to query previous versions of tables but also also provides a powerful mechanism to recover from erroneous operations.

Uses of Flashback Query include recovering lost data or undoing incorrect committed changes, for example, rows that have been deleted or updated can be immediately repaired even after they have been committed. This tool compares current data with the corresponding data at some time in the past, enabling to perform time oriented analysis, such as a daily report that shows the changes in data from yesterday. To be more precise, using Flashback Query it is possible to compare individual rows of table data, or find intersections or unions of sets of rows, check the state of transactional data at a particular time and simplify application design by removing the need to store certain types of temporal data.

### 2.2.2 Oracle Flashback Versions Query

Oracle Flashback Versions Query is an extension to SQL that can be used to retrieve the versions of rows in a given table that existed in a specific time interval. Oracle Flashback Versions Query returns a row for each version of the row that existed in the specified time interval. For any given table, a new row version is created each time the COMMIT statement is executed.

Flashback Versions Query is a powerful tool for the DBAs to run analysis to determine what happened. Additionally, application developers can use Flashback Versions Query to build customized applications for auditing purposes.

### 2.2.3 Oracle Flashback Transaction Query

Oracle Flashback Transaction Query provides a mechanism to view all changes made to the database at the transaction level.

When used with Flashback Versions Query, it offers fast and efficient means to recover from a user or application error. Flashback Transaction Query also increases the ability to perform online diagnosis of problems in the database by returning the user that changed the row, analysis of update and insert history and audits past transactions.

### 2.2.4 Oracle Flashback Table

By using Oracle Flashback Table, it is possible to recover a set of tables to a specific point in time without having to perform traditional point-in-time recovery operations. A Flashback Table operation is done in-place, while the database is online, by rolling back only the changes that are made to the given tables and their dependent objects. A Flashback Table statement is executed as a single transaction. All tables must be flashed back successfully or the entire transaction is rolled back. In most cases, this tool reduces the need for DBAs to perform more complicated point-in-time recovery operations. Even after a flashback, the data in the original table is not lost; it can later be reverted back to the original state.

### 2.2.5 Oracle Flashback Drop

Dropping objects by accident has always been a problem for users and DBAs alike. Historically, there is no easy way to recover dropped tables, indexes, constraints or triggers, but with Oracle Flashback Drop this fact has been tackled. When a user drops a table, Oracle automatically places it into the Recycle Bin. The Recycle Bin is a virtual container where all dropped objects reside. Users can continue to query data in a dropped table.

### 2.2.6 Oracle Flashback Database

Oracle Flashback Database is faster than the traditional point-in-time recovery that uses restored files and redo log files. As a database grows in size, the length of time required to restore all the data files to perform a traditional point-in-time recovery becomes prohibitive. With Flashback Database, the time to recover a database is now proportional to the number of changes that need to be backed out (and not to the size of the database) because you do not have to restore data files.

Flashback Database is implemented by using a type of log file called Flashback Database logs. The Oracle database periodically logs “before images” of data blocks in the Flashback Database logs. Block images can be reused to quickly back out the data file changes to any time at which flashback logs are captured. Then, changes from the redo log files are applied to fill in the gap. The Flashback Database logs are automatically created and managed in the flash recovery area.

With Oracle Flashback Database, the time to restore a backup when fixing human error that has a database-wide impact is eliminated.

### 2.2.7 Oracle Flashback Restore Points

When an Oracle Flashback recovery operation is performed on the database, the DBA must determine the point in time identified by the System Change Number (SCN) or timestamp to which the data can later be flashed back. These points of restore are user-defined labels that can be substituted by the SCN or transaction time used in Flashback Database, Flashback Table, and Recovery Manager (RMAN) operations. Furthermore, a database can be flashed back through a previous database recovery and open reset logs by using guaranteed restore points. Guaranteed restore points allow major database changes such as database batch jobs, upgrade, or patch to be quickly undone by ensuring that the undo required to rewind the database is retained.

## 2.3 Rman

Recovery Manager (RMAN) is an Oracle utility to manage the backup and, more importantly, to manage the recovery of the database. It eliminates operational complexity while providing superior performance and availability of the database.

RMAN determines the most efficient method of executing the requested backup, restoration or recovery operation and then submits these operations to the Oracle database server for processing. RMAN and the server automatically identify modifications to the structure of the database and dynamically adjust the required operation to adapt to the changes .

RMAN allows you to back up either to disk or directly to tape and it works with third-party products designed to support backup and recovery using offline storage media. It includes an API, known as the Media Management Layer (MML), which allows third-party vendors to integrate RMAN with their backup solution. The MML has intelligence about the hardware platform and can manipulate the tape drives, carousels, the tape library and so forth. RMAN reads and writes the data and the MML manipulates the storage system, such as tape retrieval, tape switching, and so on.

RMAN manages the creation of database backups for recovery purpose and can back up the entire database, individual tablespaces, and individual datafiles, as well as archived redo logs, control files, and server parameter files to disk or tape as mentioned previously. RMAN can also purge archived redo logs once they have been successfully written to the backup device.

In RMAN terminology, the database being backed up or restored is referred to as the target database. Backup files can be written either as backupsets or image copies. A backupset contains one or more files multiplexed together. It can contain one or more backup pieces. Backupsets are always used when writing to tape; they can be optionally used when writing to disk. RMAN only writes datafile blocks that have been initialized to the backupset. Image copies are only written to disk. An image copy of a datafile consists of all blocks in that file including any uninitialized blocks. RMAN backups are typically smaller than image copies since RMAN only backs up initialized blocks that are needed for a recovery.

RMAN maintains status information in the database control file. Optionally it is possible to create a recovery catalog, which is a set of tables, indexes and packages stored in a secondary database known as the catalog database. The catalog database contains information about RMAN backups for all target databases in the enterprise and the use of the recovery catalog is mandatory for complex configurations. The benefit of using a recovery catalog is that in a situation where all the copies of the control files for a database are lost, RMAN can re-create the control files for you from the information contained in the recovery catalog. This process is more efficient than recovering an older copy of a control file and then performing a forward recovery to bring the control file up to date.

RMAN can be used to create duplicated or cloned databases, which are copies of existing databases. It can also be used to create standby databases. In both cases, the new database is known as the auxiliary database.

When RMAN performs a backup, it reads blocks from the datafiles into a memory buffer. Blocks are verified in memory before they are written to the backup location. Therefore, RMAN provides early detection and some protection against block corruption. Corrupt blocks are reported and excluded from the backup.

RMAN can perform either full or incremental backups. With a full backup, all blocks are written to disk and with an incremental backup, RMAN inspects the block header to determine whether the block has been changed since the last time it was backed up and, if so, includes it in the incremental backup.

RMAN can perform backups more efficiently than most third-party tools, because it has access to the SGA and can therefore back up blocks that have been updated in the buffer cache but not necessarily written back to disk. The basic RMAN recovery process generally involves two steps. The first step is to restore datafiles and archived redo logs from the backup device. This may be a backupset on tape or disk. One of the main benefits of RMAN is that,in conjunction with the MML, it can identify which tapes contain the disk blocks that need to be recovered, making the process more efficient and less prone to human error than a manual recovery.

The second step is to perform media recovery to apply archived redo logs to the restored files. RMAN makes the recovery process very simple by identifying the object(s) that need recovery, retrieving the media that contains those objects, restoring only the required objects, and then automatically recovering the database [8].

## 2.4 Real Application Cluster

Oracle Real Application Clusters (RAC) allows the Oracle database to run any packaged or custom application unchanged across a set of clustered servers. This capability provides the highest levels of availability and the most flexible scalability. If a clustered server fails, the Oracle database will continue running on the surviving servers. When more processing power is needed, another server can be added without interrupting user access to data.

RAC enables multiple instances that are linked by an interconnect to share access to an Oracle database. In a RAC environment, the Oracle database runs on two or more systems in a cluster while concurrently accessing a single shared database. The result is a single database system that spans multiple hardware systems yet appears as a single unified database system to the application. This enables RAC to provide high availability, scalability and redundancy during failures within the cluster. RAC accommodates all system types, from read-only data warehouse (DSS) systems to update-intensive online transaction processing (OLTP) systems.

High availability configurations have redundant hardware and software that maintain operations by avoiding single points-of-failure. To accomplish this, the Oracle Clusterware is installed as part of the RAC installation process. Oracle Clusterware is a portable solution that is integrated and designed specifically for the Oracle database. In a RAC environment, Oracle Clusterware monitors all Oracle components (such as instances and listeners). If a failure occurs, Oracle Clusterware will automatically attempt to restart the failed component. Other non-Oracle processes can also be managed by Oracle

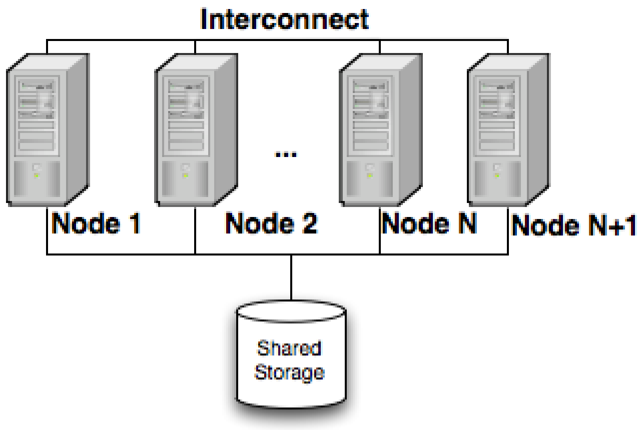

Figure 2.2: RAC design

Clusterware. During outages, Oracle Clusterware relocates the processing performed by the inoperative component to a backup component. For example, if a node in the cluster fails, Oracle Clusterware will cause client processes running on the failed node to reconnect and resume running on a surviving node.

The Oracle Clusterware requires two files: the Oracle Cluster Registry (OCR) and the voting disk. To avoid single points-of-failure, the Oracle Clusterware automatically maintains redundant copies of these files. Oracle Clusterware also enables the replacement of a damaged copy of the OCR online. Oracle's recovery processes quickly re-master resources, recover partial or failed transactions, and quickly restore the system.

### 2.4.1 Services

Services are a logical abstraction for managing workloads. Services divide the universe of work executing in the Oracle Database into mutually disjoint classes. Each service represents a workload with common attributes, service-level thresholds and priorities.

Services are built into the Oracle Database providing a single-system image for workloads, prioritization for workloads, performance measures for real transactions, alerts and actions when performance goals are violated. These attributes are handled by each instance in the cluster by using metrics, alerts, scheduler job classes and the resource manager. With RAC, services facilitate load balancing, allow end-to-end lights-out recovery and provide full location transparency.

A service can span one or more instances of an Oracle Database in a cluster and a single instance can support multiple services. The number of instances offering the service is transparent to the application. Services enable the automatic recovery of work. Following outages, the service is recovered automatically at the surviving instances. When instances are later repaired, services that are not running are restored automatically by Oracle Clusterware. Immediately, the service changes state, up, down or too busy; a notification is available for applications using the service to trigger immediate recovery and load-balancing actions. Listeners are also aware of services availability and they are responsible for distributing the workload on surviving instances when new connections are made. This architecture forms an end-to-end continuous service for applications.

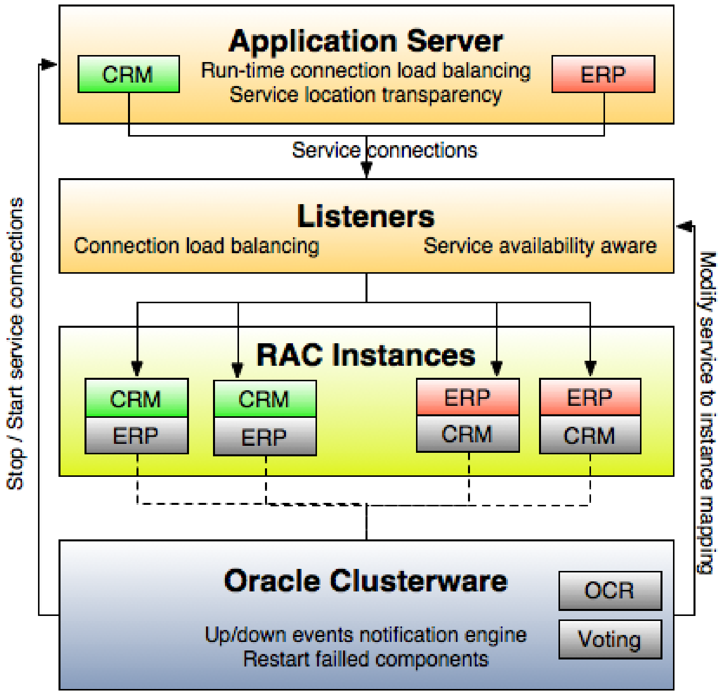
Application Server
CRM
Run-time connection load balancing
Service location transparency
ERP
Service connections
Listeners
Connection load balancing
Service availability aware
RAC Instances
CRM
ERP
CRM
ERP
ERP
CRM
ERP
CRM
Oracle Clusterware
Up/down events notification engine
Restart failled components
OCR
Voting
Stop / Start service connections
Modify service to instance mapping

Figure 2.3: RAC and Services (adapted from [14])

### 2.4.2 RAC Extended

Typically, RAC databases share a single set of storage and are located on servers in the same data center. With extended RAC, it is

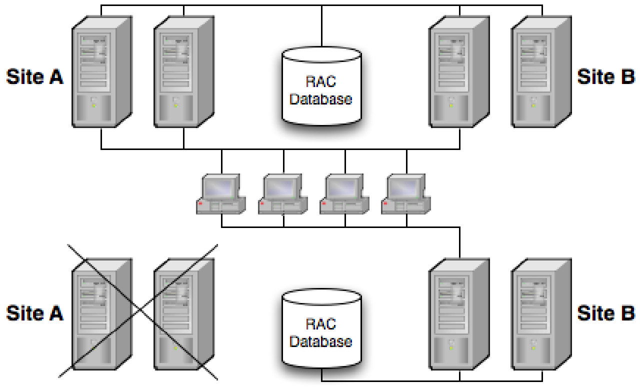

Figure 2.4: Extended RAC Overview

possible to use disk mirroring and Dense Wavelength Division Multiplexing (DWDM) equipment to extend the reach of the cluster. This configuration allows two data centers, separated by up to 100 kilometers, to share the same RAC database with multiple RAC instances spread across the two sites.

As shown in Figure 2.4, this RAC topology is very interesting, because the clients work gets distributed automatically across all nodes independently of their location, and in case one site goes down, the clients work continues to be executed on the remaining site. The types of failures that extended RAC can cover are mainly failures of an entire data center due to a limited geographic disaster. Fire, flooding and site power failure are just a few examples of limited geographic disasters that can result in the failure of an entire data center.

### Extended RAC Connectivity

In order to extend a RAC cluster to another site separated from your data center by more than ten kilometers, it is required to use DWDM over dark fiber to get good performance results. DWDM is a technology that uses multiple lasers and transmits several wavelengths of light simultaneously over a single optical fiber. DWDM enables the existing infrastructure of a single fiber cable to be dramatically increased. DWDM systems can support more than 150 wavelengths, each carrying up to 10Gbps. Such systems provide more than a terabit per second of data transmission on one optical strand that is thinner than a human hair.

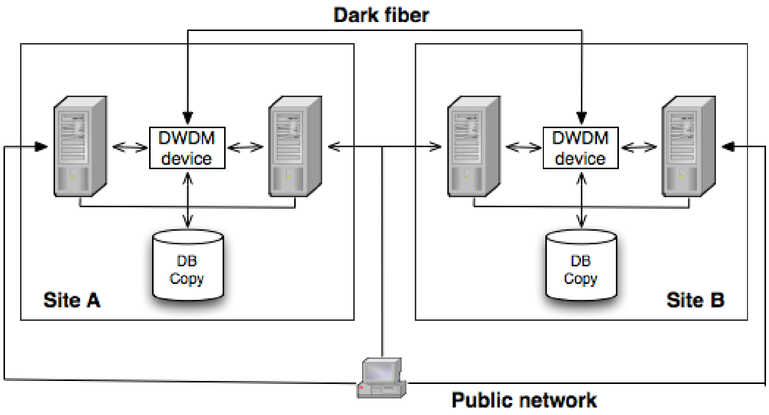

Figure 2.5: Extended RAC Connectivity

As shown in Figure 2.5, each site should have its own DWDM device connected together by a dark fiber optical strand. All traffic between the two sites is sent through the DWDM and carried on dark fiber. This includes mirrored disk writings, network and heartbeat traffic, and memory-to-memory data passage. Also shown in the graphic are sets of disks at each site. Each site maintains a copy of the RAC database.

It is important to note that depending on the site's distance, the minimum value of buffer credits needs to be determined and tuned in order to maintain the maximum link bandwidth. Buffer credit is a mechanism defined by the Fiber Channel standard that establishes the maximum amount of data that can be sent at one time.

### Extended RAC Disk Mirroring

Although there is only one RAC database, each data center has its own set of storage that is synchronously mirrored using either a cluster-aware host-based Logical Volume Manager (LVM) solution, such as SLVM with MirrorDiskUX, or an array-based mirroring solution, such as EMC SRDF. With host-based mirroring, shown in Figure 2.6, the disks appear as one set, and all I/Os get sent to both sets of disks. This solution requires closely integrated clusterware and LVM, which does not exist with the Oracle Database 10g clusterware.

With array-based mirroring, shown in Figure 2.6, all I/Os are sent to one site, and are then mirrored to the other. This alternative is the only option if we only have the Oracle Database 10g clusterware.

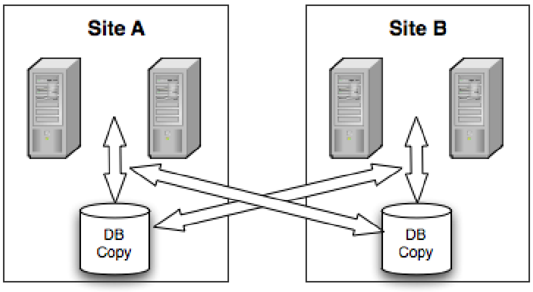

Figure 2.6: Extended RAC Disk Host-based mirroring

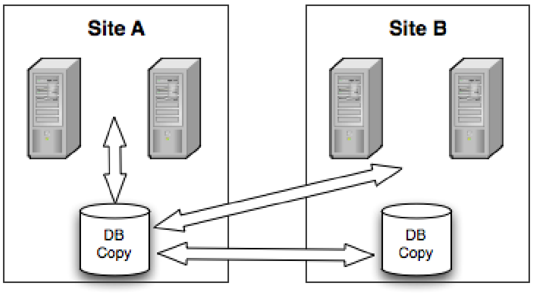

Figure 2.7: Extended RAC Disk Remote array-based mirroring

In fact, this solution is like a primary/secondary site setup. If the primary site fails, all access to primary disks is lost. An outage may be incurred before the switch to the secondary site.

## 2.5 Data Guard

Oracle Data Guard provides a comprehensive set of services that create, maintain, manage and monitor one or more standby databases to enable production databases to survive failures, disasters, errors and data corruption. Data Guard maintains these standby databases as transactionally consistent copies of the production database. Then, if the production database becomes unavailable due to a planned or an unplanned outage, Data Guard can switch any standby database to the production role, thus greatly reducing the downtime caused by the outage. The failover of data processing from the production to the standby database can be completely automatic and done without any human intervention, thereby reducing the management costs associated with the Data Guard configuration. Data Guard can be used with traditional backup, restored and clustering solutions to provide a high level of data protection and data availability [3].

Data Guard can ensure no data loss, even in the face of unforeseen disasters. A standby database provides a safeguard against data corruption and user errors. Storage level physical corruptions on the primary database do not propagate to the standby database. Similarly, logical corruptions or user errors that cause the primary database to be permanently damaged can be resolved. Finally, the redo data is validated when it is applied to the standby database.

The standby database tables that are updated with redo data received from the primary database can be used for other tasks such as backups, reporting, summations and queries, thereby reducing the primary database workload necessary to perform these tasks, saving valuable CPU and I/O cycles. With a logical standby database, users can perform normal data manipulation on tables in schemas that are not updated from the primary database. A logical standby database can remain open while the tables are updated from the primary database, and the tables are simultaneously available for read-only access. Finally, additional indexes and materialized views can be created on the maintained tables for better query performance and to suit specific business requirements.

### 2.5.1 Data Guard Configurations

A Data Guard configuration consists of one production database and one or more physical or logical standby databases. The databases in a Data Guard configuration are connected by Oracle Net and may be dispersed geographically. There are no restrictions on where the databases are located if they can communicate with each other. For example, it is possible to have a standby database in the same building as the primary database to help manage planned downtime and two or more standby databases in other locations for use in disaster recovery [2].

A standby database can be either a Physical standby database or a Logical standby database. A Physical standby database is kept in sync with the primary database by using media recovery to apply redo that was generated on the primary database. It provides a identical copy of the primary database on a block-for-block basis. A Logical standby database is kept in sync with the primary database

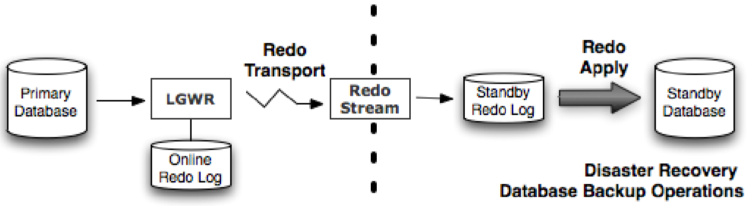

Figure 2.8: Typical Data Guard Configuration (based on [2])

by transforming redo data received from the primary database into logical SQL statements and then executing the SQL statements on the standby database.

The Figure 2.8 shows a typical Data Guard configuration that contains a primary database that transmits redo data to a standby database.

## 2.5.2 Data Guard Services

Data Guard manages the transmission of redo data, the application of redo data and changes to the database roles using several services.

### Redo Transport Services

Redo transport services control the automated transfer of redo data from the primary database to one or more archival destinations.

This service manages the process of resolving gaps, so if connectivity is lost between the primary and one or more standby databases (for example, due to network problems), redo data being generated on the primary database cannot be sent to those standby databases. Once a connection is reestablished, the missing archived redo log files (referred to as a gap) are automatically detected, which then automatically transmits the missing archived redo log files to the standby databases and the standby databases are synchronized with the primary database automatically [13].

Also, it enforces the database protection mode and automatically detects missing or corrupted archived redo log files on a standby system and automatically retrieves replacement archived redo log files from the primary database or from another standby database.

### Log Apply Services

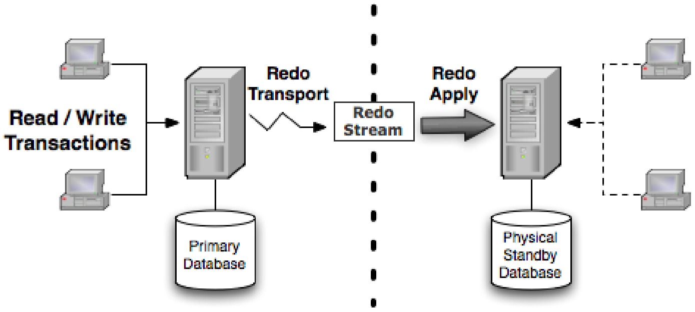

Figure 2.9: Automatic Updating of a Physical Standby Database (based on [2])

The redo data transmitted from the primary database is written on the standby system into standby redo log files, if configured, and then archived into archived redo log files. Log apply services automatically apply the redo data on the standby database to maintain consistency with the primary database. It also allows read-only access to the data.

The main difference between physical and logical standby databases is the manner in which log apply services apply the archived redo data. For a Physical standby databases, Data Guard uses Redo Apply technology, which applies redo data on the standby database using standard recovery techniques of an Oracle Database, as shown in Figure 2.9.

For a Logical standby database, Data Guard uses SQL Apply technology, which first transforms the received redo data into SQL statements and then executes the generated SQL statements on the logical standby database, as shown in Figure 2.10.

### Role Transitions

An Oracle Database operates in one of two roles: primary or standby. Using Data Guard, it is possible to change the role of a database using either a switchover or a failover operation.

A switchover is a role reversal between the primary database and one of its standby databases. A switchover ensures no data loss. This is typically used for planned maintenance of the primary database. During a switchover, the primary database moves to a standby role, and the standby database moves to the primary role. The transition occurs without having to re-create either database.

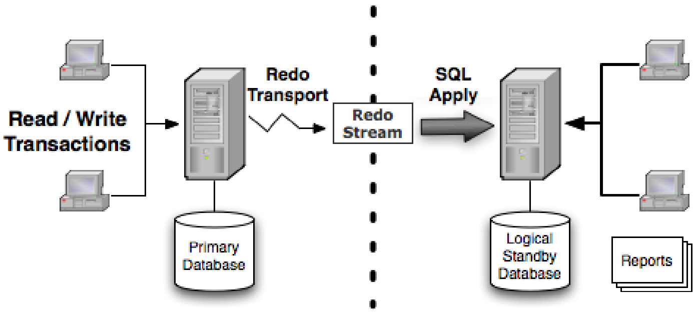

Figure 2.10: Automatic Updating of a Logical Standby Database (based on [2])

A failover is when the primary database is unavailable. Failover is performed only in the event of a catastrophic failure of the primary database in a timely manner. During a failover operation, the failed primary database is removed from the Data Guard environment, and a standby database assumes the primary database role. You invoke the failover operation on the standby database that you want to fail over to the primary role.

### 2.5.3 Data Guard Protection Modes

In some situations, it is not possible to afford to lose data. In other situations, the availability of the database may be more important than the loss of data. Some applications require maximum database performance and can tolerate some small amount of data loss. The following descriptions summarize the three distinct modes of data protection [2].

#### Maximum protection

This protection mode ensures that no data loss will occur if the primary database fails. To provide this level of protection, the redo data needed to recover each transaction must be written to both the local online redo log and to the standby redo log on at least one standby database before the transaction commits. To ensure data loss cannot occur, the primary database shuts down if a fault prevents it from writing its redo stream to the standby redo log of at least one transactionally consistent standby database. For multiple-instance Real Application Clusters (RAC) databases, Data Guard shuts down the primary database if it is unable to write the redo records to at least one properly configured database instance.

### Maximum availability

This protection mode provides the highest level of data protection that is possible without compromising the availability of the primary database. Like maximum protection mode, a transaction will not commit until the redo needed to recover that transaction is written to the local online redo log and to the standby redo log of at least one transactionally consistent standby database. Unlike maximum protection mode, the primary database does not shut down if a fault prevents it from writing its redo stream to a remote standby redo log. Instead, the primary database operates in maximum performance mode until the fault is corrected, and all gaps in redo log files are resolved. When all gaps are solved, the primary database automatically changes to maximum availability mode.

This mode ensures that no data loss will occur if the primary database fails, but only if a second fault does not prevent a complete set of redo data from being sent from the primary database to at least one standby database.

### Maximum performance

This protection mode (the default) provides the highest level of data protection that is possible without affecting the performance of the primary database. This is accomplished by allowing a transaction to commit as soon as the redo data needed to recover that transaction is written to the local online redo log. The primary database redo data stream is also written to at least one standby database, but that redo stream is written asynchronously with respect to the transactions that create the redo data.

When network links with sufficient bandwidth are used, this mode provides a level of data protection that approaches that of maximum availability mode with minimal impact on primary database performance.

The maximum protection and maximum availability modes require that standby redo log files are configured on at least one standby database in the configuration.

All three protection modes require that specific log transport attributes be specified on the LOG_ARCHIVE_DEST_n initialization parameter to send redo data to at least one standby database.

### 2.5.4 Alternative to Data Guard

Data guard is a high availability feature only available with Enterprise edition. This section is dedicated to discuss an alternative to create a duplicate (clone) database of production database in standard edition which will support the following features of (physical) standby in Enterprise Edition:

- Keep the clone database in synchronized state with primary by applying the archivelogs from primary.
- Open the database in read only mode for reporting purpose.

To create an alternative to Data Guard, the archivelogs of the primary database need to be copied to the standby database, then they need to be registered and the recovery process started, bringing the standby database to the last SCN that are on the archivelogs.

This alternative standby database doesn't support advance data guard features like switchover, automatic log shipping and apply services (Managed recovery).

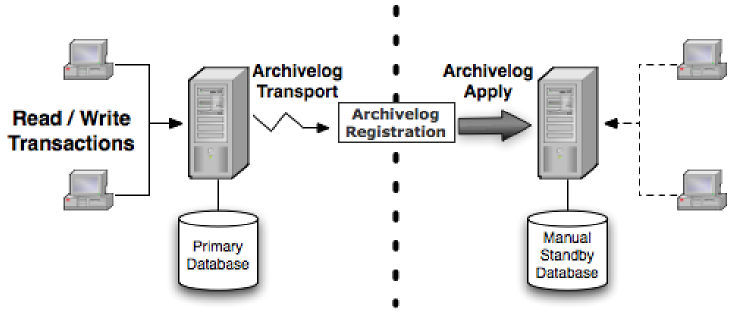

Figure 2.11: Manual Updating of a Physical Standby Database

# Chapter 3

# RAC and Data Guard

RAC and Data Guard together provide the benefits of system-level, site-level and data-level protection, resulting in high levels of availability and disaster recovery without loss of data:

- RAC addresses system failures by providing quick and automatic recovery from failures at the node and instance level.
- Data Guard addresses site failures and data protection through transactionally consistent primary and standby databases that do not share disks, enabling recovery from site disasters and data corruption.

RAC and Data Guard provide the basis of the database Maximum Availability Architecture (MAA) solution. MAA provides the most comprehensive architecture for reducing down time for scheduled outages and preventing, detecting and recovering from unscheduled outages. The recommended MAA has two identical sites: the primary site contains the RAC database; the secondary site contains both a physical standby database and a logical standby database on RAC. Identical site configuration is recommended to ensure that performance is not sacrificed after a failover or switchover [4].

Symmetric sites also enable processes and procedures to be kept the same between sites, making operational tasks easier to maintain and execute. The Figure 3.1 illustrates identically configured sites. Each site consists of redundant components and redundant routing mechanisms, so that services are always able to process requests even in the event of a failure. Client requests are always routed to the site playing the production role.

After a failover or switchover due to a serious outage, client requests are routed to another site that assumes the production role. Each

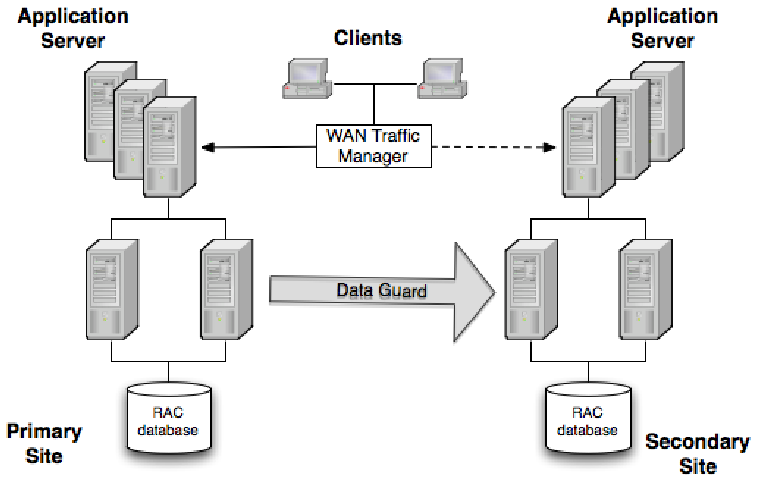

Figure 3.1: Maximum Availability Architecture (based on [14])

site contains a set of application servers or mid-tier servers. The site playing the production role contains a production database using RAC to protect from host and instance failures. The site playing the standby role contains one standby database and one logical standby database managed by Data Guard. Data Guard switchover and failover functions allow the roles to be traded between sites.

## 3.1 RAC and Data Guard Topologies

With Data Guard it is possible to configure a standby database to protect a primary database in a RAC environment. Basically, all kinds of combinations are supported. For example, it is possible to have a primary database running under RAC and a standby database running as a single-instance database. It is also possible to have both the primary and standby databases running under RAC.

**Symmetric configuration with RAC at all sites:**

- Same number of instances
- Same service preferences

**Asymmetric configuration with RAC at all sites:**

- Different number of instances

- Different service preferences

**Asymmetric configuration with mixture of RAC and single instance:**

- All sites running under Oracle Clusterware
- Some single-instance sites not running under Oracle Clusterware

However, to benefit from the tight integration of Oracle Clusterware and Data Guard Broker, the primary site and the secondary site should be running under Oracle Clusterware, and that both sites should have the same services defined.

## 3.2 RAC and Data Guard Architecture

As mentioned before, different topologies can be developed to archive high availability, while adding its underlying particularities in terms of implementation and performance. As example, it is equally possible to use a “RAC-to-Single-Instance Data Guard” configuration or a RAC-to-RAC Data Guard configuration. However in this last mode, although multiple standby instances can receive redo from the primary database, only one standby instance can apply the redo generated by the primary instances.

A RAC-to-RAC Data Guard configuration can be set up in different ways, and Figure 3.2 shows one possibility with a symmetric configuration where each primary instance sends its redo stream to a corresponding standby instance using standby redo log files. It is also possible for each primary instance to send its redo stream to only one standby instance that can also apply this stream to the standby database. However, it is possible to get performance benefits by using the configuration shown in Figure 3.2. For example, assuming that the redo generation rate on the primary is too great for a single receiving instance on the standby side to handle. Suppose further that the primary database is using the SYNC redo transport mode. If a single receiving instance on the standby cannot keep up with the primary, then the primary’s progress is going to be throttled by the standby. If the load is spread across multiple receiving instances on the standby, then this is less likely to occur.

If the standby can keep up with the primary, another approach is to use only one standby instance to receive and apply the redo. For

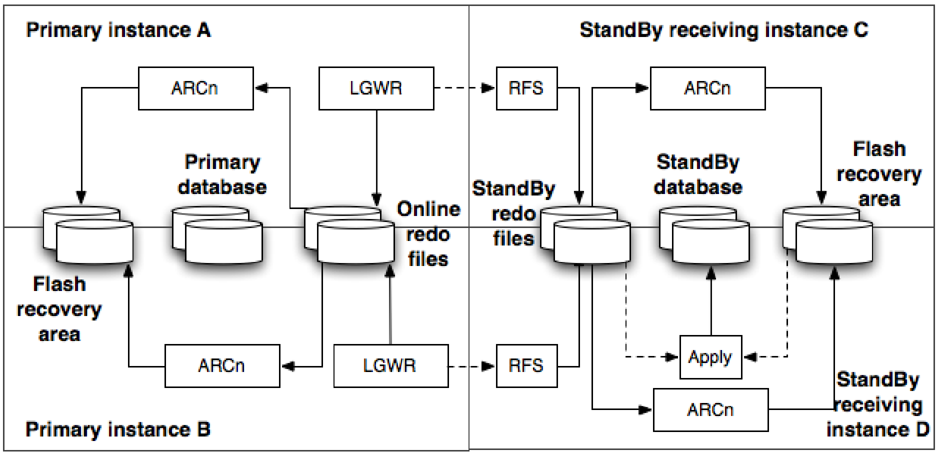

Figure 3.2: RAC and Data Guard Architecture (based on [14])

example, setting up the primary instances to remotely archive to the same Oracle Net service name. It is possible then to configure one of the standby nodes to handle that service and this instance will then receive and apply redo from the primary. If necessary to do maintenance on that node, then it is possible to stop the service on that node and start it on another node. This approach allows the primary instances to be more independent of the standby configuration because they are not configured to send redo to a particular instance.

# Chapter 4

# Evaluation

The following tests that will be presented were made using a Desktop PC with Ubuntu 9.10 as the operative system. The virtualization manager used was the Kernel-based Virtual Machine (KVM). The operation System used by each virtual machine was the Oracle Enterprise Linux 5.4 32 bits. The method used consisted in the installation of first virtual machine with all the requisites to install Oracle software and then clone each virtual machine to have a similar environment on all the tests.

## 4.1 Virtual Machine Details

Each virtual machine was configured with 640MB of memory under a disk of 10GB. Also to each operating system two more disks were presented to be used by ASM, one with 2GB for data and the other with 1Gb for the flash recovery area.

The installation of the Oracle Enterprise Linux was a minimal installation and then all the packages needed were installed. There was also the need to change several kernel parameters to fulfill all the prerequisites for the installation of the Oracle software.

On some virtual machines a third disk was configured to be used as a cluster filesystem. This disk was configured with Oracle Clusterware File System (OCFS2).

## 4.2 Initial Oracle Setup

For each architecture an initial setup was needed. This initial setup consisted on the installation of the Oracle Database software, ASM configuration and the creation of a General Oracle Database. All the installations described were made using the Oracle interactive installation method.

Table 4.1 shows the time to setup the initial configuration for a Single Instance Database.

| Operation Type | Time |
|---|---|
| Install Database Software 10.2.0.1 | 10 Minutes |
| ASM Creation | 5 Minutes |
| Database Creation | 10 Minutes |

Table 4.1: Single Instance Setup

Table 4.2 shows the time to setup the initial configuration for a Real Application Cluster Database.

| Operation Type | Time |
|---|---|
| Install Clusterware Software 10.2.0.1 | 30 Minutes |
| Install Database Software 10.2.0.1 | 20 Minutes |
| ASM Creation | 10 Minutes |
| Database Creation | 15 Minutes |

Table 4.2: RAC Database Setup

Table 4.3 shows the time to setup the initial configuration for a Single Instance Database and table 4.4 shows the Oracle Data Guard configuration time needed.

| Operation Type | Time |
|---|---|
| Install Database Software 10.2.0.1 | 10 Minutes |
| ASM Creation | 5 Minutes |
| Database Creation | 10 Minutes |

Table 4.3: Initial Data Guard Setup

## 4.3 Test Results

Several tests were chosen to verify the downtime of each architecture. Included on these tests was the Patch Set 10.2.0.4 (PatchSet 3), the

| Operation Type | Time |
|---|---|
| Install Database Software 10.2.0.1 on Standby | 10 Minutes |
| ASM Creation | 5 Minutes |
| Physical StandBy Setup | 40 Minutes |
| Logical StandBy Setup | 20 Minutes |

Table 4.4: Data Guard Setup

Critical Patch Update (CPU) of July 2009 (CPU patch 8534387) and the CPU patch of January 2010 (CPU Patch 9119226). The reason for applying two CPU patches was that the first time that a CPU patch is applied, the database needs to be recompiled and to do that, the database needs to be in restricted mode. After the first CPU, this recompilation is not need anymore.

Table 4.5 shows the times to apply each patch and also the downtime associated with each patch a Single Instance.

| Outage Type | Operation | Total Time | Downtime |
|---|---|---|---|
| PatchSet 3 | Software Upgrade | 10 Minutes | 10 Minutes |
| | Database Upgrade | 80 Minutes | 80 Minutes |
| CPU Patch 8534387 | Patch Upgrade | 10 Minutes | 10 Minutes |
| | Compile Objects | 10 Minutes | 5 Minutes |
| CPU Patch 9119226 | Software Upgrade | 8 Minutes | 5 Minutes |

Table 4.5: Single Instance Downtime

Table 4.6 shows the times to apply each patch and also the downtime associated with each patch a RAC Database.

| Outage Type | Operation | Total Time | Downtime |
|---|---|---|---|
| CRS PatchSet 3 | CRS Upgrade | 25 Minutes | 0 Minutes |
| PatchSet 3 | Software Upgrade | 15 Minutes | 15 Minutes |
| | Database Upgrade | 80 Minutes | 80 Minutes |
| CPU Patch 8534387 | Patch Upgrade | 20 Minutes | 0 Minutes |
| | Compile Objects | 15 Minutes | 5 Minutes |
| CPU Patch 9119226 | Software Upgrade | 15 Minutes | 0 Minutes |

Table 4.6: RAC Database Downtime

The downtime associated with the application of CPU patch in a RAC Database was reduced in comparison with the application of the same patch on the Single Instance Database. The reason for that was that RAC Database allows rolling upgrades. The image 4.1

shows how the rolling upgrade method works for a RAC Database architecture.

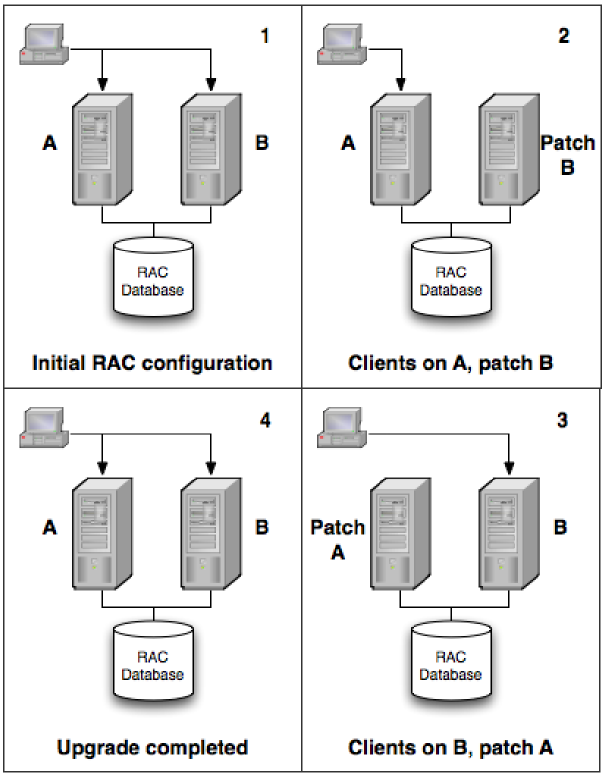

Figure 4.1: Rac Apply Patch

Table 4.7 shows the times to apply each patch and also the downtime associated with each patch when a Data Guard Database is in use.

| Outage Type | Operation | Total Time | Downtime |
|---|---|---|---|
| PatchSet 3 | Software Upgrade | 20 Minutes | 1 Minutes |
| | Database Upgrade | 160 Minutes | 0 Minutes |
| CPU Patch 8534387 | Patch Upgrade | 20 Minutes | 1 Minutes |
| | Compile Objects | 20 Minutes | 0 Minutes |
| CPU Patch 9119226 | Software Upgrade | 15 Minutes | 1 Minutes |

Table 4.7: Data Guard Downtime

The downtime associated with the application of the PatchSet and of the CPU patches in a Data Guard architecture was reduced in comparison with the application of the same patches on the Single Instance Database and on the RAC Database. This happens because

Data Guard allows rolling upgrades, even in the case of a PatchSet or major release. Figure 4.2 shows how the rolling upgrade method works for a Data Guard architecture [7].

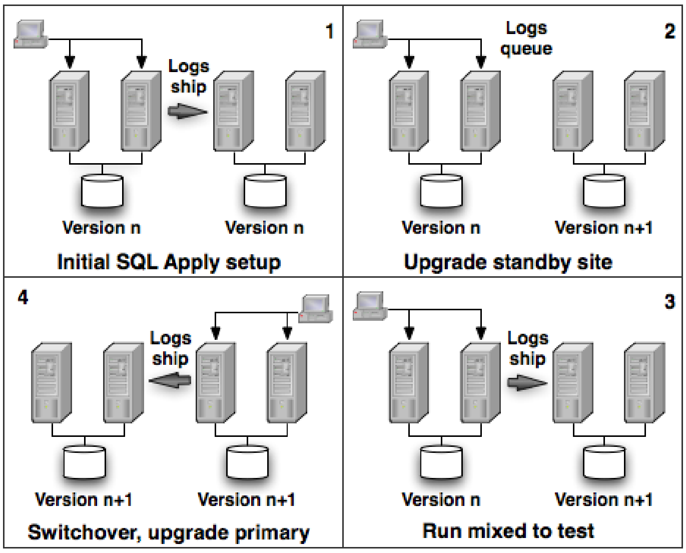

Figure 4.2: Data Guard Apply Patch

## 4.4 Conclusions

According to the previous results, RAC based architectures provide several benefits to Single Instances, such as:

- the ability to tolerate and quickly recover from computer and instance failures;
- fast, automatic, intelligent connection and service relocation and failover;
- rolling patch upgrades for qualified one-off patches;
- rolling release upgrades of Oracle Clusterware;
- load balancing advisory; runtime connection load balancing;
- flexibility to scale up processing capacity using commodity hardware without downtime or changes to the application

- comprehensive manageability integrating database and cluster features.

As shown by Figure 4.3, the downtimes using a RAC database compared with the Single Instance Database is more significant when a simple patch is applied. When a PatchSet is applied, the downtimes are practically the same.

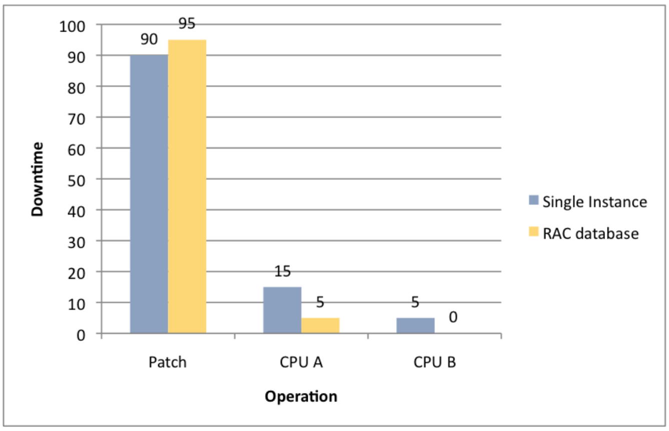

Figure 4.3: Downtime Single Vs RAC

The previous test indicate that Data Guard provides several benefits, such as:

- maintaining real-time, transactionally consistent database copies to provide protection against unplanned downtime and disaster;

- complete data protection against computer failures, human errors, data corruption and site failures;

- reduces planned downtime for hardware, system upgrades, Oracle patch set, database upgrades and multiple levels of data protection

- enhanced performance to balance data availability against system performance requirements.

As shown by Figure 4.4, the downtimes using a Data Guard architecture compared with the RAC Database and the Single Instance Database are significantly reduced, even in the case of a PatchSet or main release is upgraded.

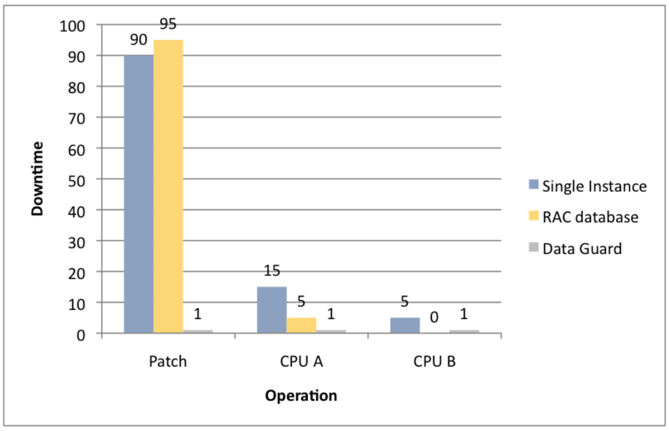

Figure 4.4: Downtime Single Vs RAC Vs Data Guard

Table 4.8 shows the summary of the Oracle Solutions for Unplanned Downtime and the schema show on Figure 4.5 points out the Oracle Solutions to prevent Planned Downtime and Unplanned Downtime.

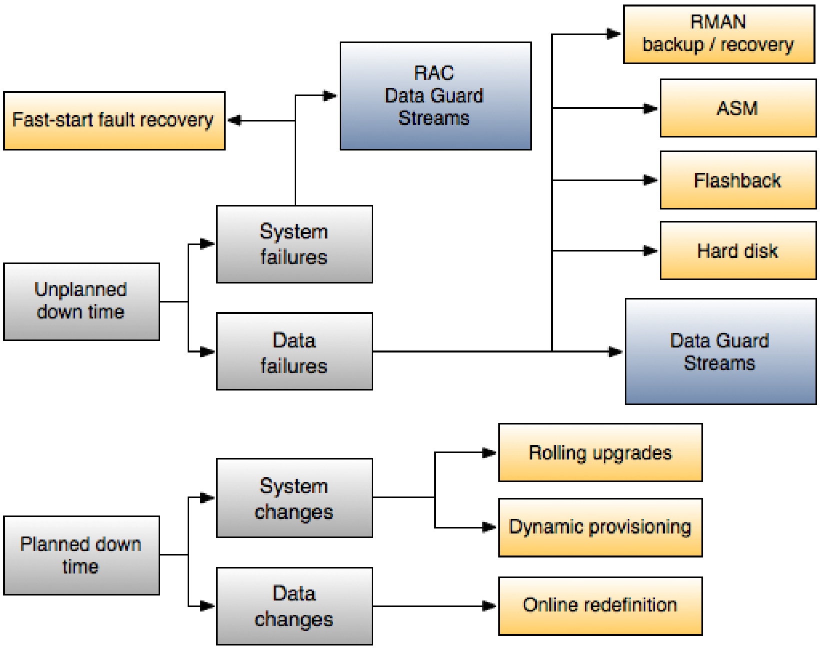

Figure 4.5: Oracle Solutions

| Outage Type | Solution | Benefits | Recovery Time |
|---|---|---|---|
| Computer failures | RAC | Automatic recovery of failed nodes and instances, fast connection failover, and service failover | No downtime |
| | Data Guard | Fast Start Failover and fast connection failover | Seconds to 5 minutes |
| Storage failures | ASM | Mirroring and online automatic rebalance | No downtime |
| | RMAN with FRA | Fully managed database recovery and managed disk-based backups | Minutes to hours |
| | Data Guard | Fast Start Failover and fast connection failover | Seconds to 5 minutes |
| Human errors | Flashback | Fine-grained and database-wide rewind capability | $<$ 30 Minutes |
| Data corruption | HARD | Corruption prevention within a storage array | No downtime |
| | RMAN with FRA | Online block media recovery and managed disk-based backups | Minutes to hours |
| | Data Guard | Automatic validation of redo blocks before they are applied, execute fast failover to an uncorrupted standby database | Seconds to 5 minutes |
| Site failures | RMAN | Fully managed database recovery and integration with tape management vendors | Hours to days |
| | Data Guard | Fast Start Failover and fast connection failover | Seconds to 5 minutes |

Table 4.8: Solution for Unplanned Downtime (based on [3])

# Chapter 5

# Conclusions and Future Work

The purpose of this dissertation was the study of the Oracle Database Suite, install each of these architectures as best technical practices and test each of them to be able to select the most suitable architecture as means towards improving the availability of the overall system. This has allowed me to learn and deploy the best practices of each architecture. By following this approach, the implementation of them and also the time to upgrade the system is much easier. It was proved that Oracle 10gR2 has the potential to simplify some critical aspects in providing a high availability system despite the usage of elementary RDBMS features. Inside these features there is the ASM, which demonstrates unique capabilities for database applications as it provides a higher level of availability, online disk reconfiguration and dynamic rebalancing, significantly less work to provide and manage database storage and elimination of the installation and maintenance of specialized storage products and subsequent costs. The tests that where run proved that ASM allows the reorganization of the storage system, from the simple task of increasing the capacity up to the replacement of entire storage, thanks to the ability to add and remove disks online. This feature becomes particularly useful in systems with large volumes of information, like systems with TeraByte of information, where this type of operation without a storage system such as the ASM becomes impractical.

Reorganizing tables online also provides a substantial increase in availability compared to traditional methods of redefining tables. This capability improves data availability, query performance, response time and disk space usage all of which are important in a critical environment because they make the process of application upgrade easier, safer and faster. Online reorganization is supported for advanced queues, clustered tables, materialized views and abstract data types.

The use of the flash recovery area provides a unified storage location of related recovery files, a management of the disk space allocated for recovery files in order to simplify database administration tasks and simultaneously fast and reliable disk-based backup/restore procedures.

The Recovery Manager (RMAN) utility is a command-line client for advanced functions. It has powerful control and scripting language. RMAN has a published API that enables interface with most popular backup software, it backs up data, control, archived log and server parameter files to the disk or tape.

The Oracle Database 10g architecture leverages the unique technological advances in the area of database recovery from the loss of data due to human errors. The Flashback technology provides a set of new features to view and rewind data back and forth in time. This technology represents a revolution to recovery by simply operating on the changed data. The time it takes to recover from the error is equal to the amount of time it takes to make the error. When used, the Flashback provides significant benefits over media recovery in terms of ease of use, availability and restoration time.

With the use of Real Application Cluster it is possible to tolerate and quickly recover from computer or instance failures. This has been confirmed simulating a critical problem with one of the nodes. Also some patches have been applied to test the impact on the availability this kind of task could have. Once again the best practices have been followed and it was proved that it is possible to apply some patches with zero downtime when using rolling patch upgrades for qualified one-off patches and rolling release upgrades of Oracle Clusterware. This type of architecture can be found in many systems that need to be running up to 24X24 7X7 such web shops and Hospitals.

Data Guard provides a greater disaster protection as it eliminates the distance limitation without performance hit. It provides additional protection against corruptions because it uses a separate database in stand-by and it has the option to delay a user operation to protect the system against human errors. Data Guard also provides better planned maintenance capabilities by supporting full rolling upgrades. This architecture inherently allows efficient use of system resources by diverting reporting and backup operations from the production database to standby databases. Data Guard is frequently found in insurance companies, banks or telecommunication companies where the protection of the information is mandatory and to be prepared to quickly recover the system even in case of site-level disaster.

RAC and Data Guard together provide the benefits of system-level, site-level, and data-level protection, resulting in high levels of availability and disaster recovery without loss of data or performance to the end-user. Due to the resources needed to implement it I was unable to implement and test this type of architecture. However, it is easy to realize the potential of this one, as it consists of the functionality of Real Application Cluster and Data Guard. Normally this kind of solution is implemented in big groups, such insurance companies, banks and telecommunication companies.

Some future work could be done in this area, such as the study of the Oracle Streams and some of the new features on Oracle 11g. Among the most interesting features, Active Data Guard stands out as it provides the management, monitoring, and automation software to create and maintain one or more synchronized replicas (standby databases) of a production database (primary database). An Active Data Guard standby database is an exact copy of the primary that is opened read-only while it continuously applies changes transmitted by the primary database.

Other important feature that could be studied in future work is the Oracle Automatic Storage Management Cluster File System (ACFS). ACFS can work with Single Instance Installations as well as Clustered architectures. As a matter of fact, it is strongly integrated with the Oracle Clusterware solution. This new feature allows storing in ASM the application file data, Oracle Cluster Registry (OCR), the Cluster Voting Disk and Oracle Binaries. With ACFS it is possible to have centralized and detailed trace files, alert logs, reports and other analysis tools. These characteristics make ASM a complete storage management system for both database and non-database files, completely eliminating the need for any third party cluster file systems.

Another interesting line of research would be to perform some tests comparing other databases software for high-availability. Studies comparing in depth the tested architecture with Microsoft SQL Server or MySQL could also be an interesting line of research.

# Bibliography

[1] Automatic storage management overview and technical best practices.

[2] Oracle data guard concepts and administration.

[3] Oracle database high availability solutions.

[4] Oracle maximum availability architecture.

[5] I. Abramson, M. Abbey, and M. Corey. *Oracle Database 10g: A Beginner's Guide.* McGraw-Hill, Inc., New York, NY, USA, 2004.

[6] T. Best and M. J. Billings. *Oracle Database 10g: Administration Workshop.* Oracle, 2005.

[7] R. Dutcher. Database rolling upgrade using data guard sql apply. Technical report, Oracle, 2009.

[8] J. Dyke and S. Shaw. *Pro Oracle Database 10g RAC on Linux: Installation, Administration, and Performance (Expert's Voice in Oracle).* Apress, Berkely, CA, USA, 2006.

[9] R. Dyke, L. Haan, C. Jeal, J. Stern, and J.-F. Verrier. Oracle database 10g: New features for administrators. Technical report, Oracle, 2004.

[10] R. Dyke and D. Keesling. *Oracle Database 10g: Data Guard Administration.* Oracle, 2005.

[11] K. Gopalakrishnan. *Oracle Database 10g Real Application Clusters Handbook.* McGraw-Hill, Inc., New York, NY, USA, 2007.

[12] M. Hart and R. G. Freeman. *Oracle Database 10g RMAN Backup & Recovery.* McGraw-Hill, Inc., New York, NY, USA, 2007.

[13] M. Hart and S. Jesse. *Oracle Database 10g High Availability with RAC, Flashback & Data Guard*. McGraw-Hill, Inc., New York, NY, USA, 2004.

[14] J. Spiller. *Oracle Database 10g: RAC for Administrator*. Oracle, 2006.