

Regis University

ePublications at Regis University

Regis University Student Publications
(comprehensive collection)

Regis University Student Publications

Fall 2008

A Business Continuity Solution for Telecommunications Billing Systems

Andrew McCormack
Regis University

Follow this and additional works at: <https://epublications.regis.edu/theses>



Part of the [Computer Sciences Commons](#)

Recommended Citation

McCormack, Andrew, "A Business Continuity Solution for Telecommunications Billing Systems" (2008).
Regis University Student Publications (comprehensive collection). 110.
<https://epublications.regis.edu/theses/110>

This Thesis - Open Access is brought to you for free and open access by the Regis University Student Publications at ePublications at Regis University. It has been accepted for inclusion in Regis University Student Publications (comprehensive collection) by an authorized administrator of ePublications at Regis University. For more information, please contact epublications@regis.edu.

Regis University
College for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

Use of the materials available in the Regis University Thesis Collection ("Collection") is limited and restricted to those users who agree to comply with the following terms of use. Regis University reserves the right to deny access to the Collection to any person who violates these terms of use or who seeks to or does alter, avoid or supersede the functional conditions, restrictions and limitations of the Collection.

The site may be used only for lawful purposes. The user is solely responsible for knowing and adhering to any and all applicable laws, rules, and regulations relating or pertaining to use of the Collection.

All content in this Collection is owned by and subject to the exclusive control of Regis University and the authors of the materials. It is available only for research purposes and may not be used in violation of copyright laws or for unlawful purposes. The materials may not be downloaded in whole or in part without permission of the copyright holder or as otherwise authorized in the "fair use" standards of the U.S. copyright laws and regulations.

A business continuity solution for Telecommunications Billing Systems

by

Andrew McCormack

Submitted to the Graduate Faculty of
NUI Galway in partial fulfillment
of the requirements for the degree of
Master of Science in Software and Information Systems

National University of Ireland Galway

2008

NUI GALWAY
College of Engineering and informatics

This thesis was presented

by

Andrew McCormack

Regis University

School for Professional Studies Graduate Programs

MScSIS Program

Graduate Programs Final Thesis

Certification of Authorship

Print Student's Name Andrew McCormackTelephone +353 86 6082410 Email acmccorm@gmail.comDate of Submission 20 August 2008 Degree Program MScSISTitle of Submission **A business continuity solution for Telecommunications Billing Systems**Project Advisor/Faculty Name **Brad Blake**

Certification of Authorship:

I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for the purpose of partial fulfillment of requirements for the Master of Science in Software and Information Systems Degree Program.

Student's (Electronic) Signature:



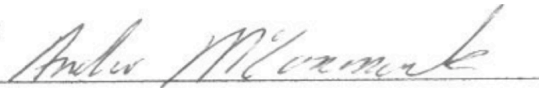
9-August-2008

Regis University
School for Professional Studies Graduate Programs
MScSIS Program
Graduate Programs Final Thesis
Authorization to Publish Student Work

I, Andrew McCormack the undersigned student, in the Master of Science in Software and Information Systems Degree Program hereby authorize Regis University to publish through a Regis University owned and maintained web server, the document described below ("Work"). I acknowledge and understand that the Work will be freely available to all users of the World Wide Web under the condition that it can only be used for legitimate, non-commercial academic research and study. I understand that this restriction on use will be contained in a header note on the Regis University web site but will not be otherwise policed or enforced. I understand and acknowledge that under the Family Educational Rights and Privacy Act I have no obligation to release the Work to any party for any purpose. I am authorizing the release of the Work as a voluntary act without any coercion or restraint. On behalf of myself, my heirs, personal representatives and beneficiaries, I do hereby release Regis University, its officers, employees and agents from any claims, causes, causes of action, law suits, claims for injury, defamation, or other damage to me or my family arising out of or resulting from good faith compliance with the provisions of this authorization. This authorization shall be valid and in force until rescinded in writing.

Print Title of Document(s) to be published: **A business continuity solution for Telecommunications Billing Systems**

Student's (Electronic) Signature: _____



9-August-2008

Check appropriate statement:

☒ The Work does not contain private or proprietary information.

Regis University
School for Professional Studies Graduate Programs
MScSIS Program
Graduate Programs Final Thesis
Project Advisor/ Faculty Approval Form

Student's Name: Andrew McCormack Program MScSIS

Thesis Title: **A business continuity solution for Telecommunications Billing Systems**

Project Advisor Name Brad Blake

Project Faculty Name _____

Advisor/Faculty Declaration:

I have advised this student through the Project/Thesis Process and approve of the final document as acceptable to be submitted as fulfillment of partial completion of requirements for the MScSIS Degree Program.



18-Aug-08

Original Advisor Signature

Date

Original Module/Class Facilitator Signature

Date

Degree Chair Approval if:

The student has received project approval from Faculty and has followed due process in the completion of the project and subsequent documentation.

Original Degree Chair/Designee Signature

Date

Copyright © by Andrew McCormack

2008

ACKNOWLEDGEMENTS

I would like to thank my wife Nora and children Hannah, Ben and Gerald for their support and patience.

ABSTRACT

The billing system is a critical component in a Telecommunications service provider's suite of business support systems – without the billing system the provider cannot invoice their customers for services provided and therefore cannot generate revenue. Typically billing systems are hosted on a single large Unix/Oracle system located in the company's data centre. Modern Unix servers with their redundant components and hot swap parts are highly resilient and can provide levels of availability when correctly installed in properly managed data centre with uninterruptible power supplies, cooling etc. High Availability clustering through the use of HP MC/ServiceGuard, Sun Cluster, IBM HACMP (High Availability Cluster Multi-Processing) or Oracle Clusterware/RAC (Real Application clusters) can bring this level of availability even higher.

This approach however can only protect against the failure of a single server or component of the system, it cannot protect against the loss of an entire data centre in the event of a disaster such as a fire, flood or earthquake. In order to protect against such disasters it is necessary to provide some form of backup system on a site sufficiently remote from the primary site so that it would not be affected by any disaster, which might befall the primary site.

This paper proposes a cost effective business continuity solution to protect a Telecommunications Billing system from the effects of unplanned downtime due to server or site outages. It is aimed at the smaller scale tier 2 and tier 3 providers such as Mobile Virtual

Network Operators (MVNOs) and startup Competitive Local Exchange Carriers (CLECs) who are unlikely to have large established IT systems with business continuity features and for whom cost effectiveness is a key concern when implementing IT systems.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	VIII
ABSTRACT	IX
1.0 INTRODUCTION.....	1
1.1 BACKGROUND	1
1.2 THESIS OUTLINE	4
1.2.1 Chapter 1 – Introduction	4
1.2.2 Chapter 2 – Review of literature and research	4
1.2.3 Chapter 3 - Design of proposed solution	5
1.2.4 Chapter 4 - Implement and test proof of concept architecture	6
1.2.5 Chapter 5 – Conclusion.....	6
1.2.6 Appendix A.....	6
1.2.7 Appendix B.....	6
2.0 REVIEW OF LITERATURE AND RESEARCH	8
2.1 BUSINESS CONTINUITY	8
2.1.1 The need for Business continuity	8
2.1.2 Aspects of Business continuity	12
2.1.2.1 High Availability	12

2.1.2.2	Continuous Operation.....	15
2.1.2.3	Disaster Recovery.....	16
2.1.3	Conclusion	20
2.2	OVERVIEW OF TELECOMMUNICATIONS BILLING	21
2.2.1	What Billing Systems do.	21
2.2.2	Billing Systems architecture.	25
2.3	BUSINESS CONTINUITY AND THE BILLING SYSTEM.	27
2.3.1	Business continuity requirements.....	27
2.3.2	Replication requirements.....	30
2.3.3	Business continuity requirements for a small tier 3 provider.....	31
2.4	BUSINESS CONTINUITY SOLUTION RESEARCH	33
2.4.1	Database replication.....	34
2.4.1.1	Oracle level replication	34
2.4.1.2	Oracle Data Guard.....	35
2.4.1.3	Oracle Streams.....	37
2.4.1.4	Oracle Advanced replication	38
2.4.1.5	Oracle Recovery Manager	38
2.4.1.6	Quest Software's SharePlex for Oracle.....	39
2.4.1.7	Ixion IxPropogator.....	39
2.4.2	Remote mirroring based database replication	39
2.4.2.1	OSCP Validated remote mirroring	40
2.4.3	Filesystem replication.....	40
2.4.3.1	Hardware based mirroring.....	41

2.4.3.2	Host based mirroring	44
3.0	DESIGN OF PROPOSED SOLUTION	47
3.1	REQUIREMENTS	47
3.2	SOLUTION OVERVIEW	48
4.0	IMPLEMENT AND TEST PROOF OF CONCEPT	54
4.1	SYSTEM HARDWARE CONFIGURATION	55
4.2	SYSTEM SOFTWARE CONFIGURATION	59
4.2.1	Oracle 10g installation	59
4.2.2	Oracle Primary Database creation	60
4.2.3	Oracle Standby Database creation	62
4.2.4	Generate rows in database	62
4.2.5	Verify data replication to standby database	66
4.2.6	Generate CDR files in input filesystem	69
4.2.7	Configure and test script for automatic zfs replication.	73
4.2.8	Carry out failover test	78
5.0	CONCLUSION	83
5.1	CONCLUSIONS	83
5.2	RECOMMENDATIONS FOR FURTHER RESEARCH	84
APPENDIX A	86
APPENDIX B	90
BIBLIOGRAPHY	110

LIST OF TABLES

Table 1. 2001 cost of downtime Survey – direct monetary costs.....	9
Table 2. 2001 cost of downtime Survey – factors critical to business survival	10
Table 3. Billing system filesystems	30
Table 4. Billing system hardware requirements	32
Table 5. Billing system hardware requirements	48
Table 6. Billing system replication.....	51
Table 7. ZFS filesystems.....	57
Table 7. CDR format.....	61

LIST OF FIGURES

Figure 1. Overview of proposed solution.....	3
Figure 2. Active/Standby cluster	13
Figure 3. Active/Active cluster.....	14
Figure 4. Disaster Recovery	18
Figure 5. Billing System Overview	24
Figure 6. Billing System Architecture	25
Figure 7. Overview of proposed solution.....	52

1.0 INTRODUCTION

1.1 BACKGROUND

The billing system is a critical component in a Telecommunications service provider's suite of business support systems – without the billing system the provider cannot invoice their customers for services provided and therefore cannot generate revenue. Typically billing systems are hosted on a single large Unix/Oracle system located in the company's data centre. Modern Unix servers with their redundant components and hot swap parts are highly resilient and can provide levels of availability approaching 99.99% when correctly installed in properly managed data centre with uninterruptible power supplies, cooling etc.

High Availability clustering through the use of HP MC/ServiceGuard, Sun Cluster, IBM HACMP or Oracle Clusterware/RAC can bring this level of availability even higher. This approach however can only protect against the failure of a single server or component of the system, it cannot protect against the loss of an entire data centre in the event of a disaster such as a fire, flood or earthquake. In order to protect against such disasters it is necessary to provide some form of backup system on a site sufficiently remote from the primary site so that it would not be affected by any disaster, which might befall the primary site.

There are a number of approaches that can be taken to implement a business continuity

solution for a billing system. The approach taken will depend on a number of factors, chief of which are:

- Recovery Time Objective (RTO) – what is the maximum amount of time allowed between the declaration of a disaster and having the system up and running on the standby site?
- Recovery Point Objective (RPO) – to what point must the system be recovered?
What if any data loss is acceptable?
- Cost – how much is budgeted for the provision of the business continuity system?

These factors are interrelated – the shorter the RTO and the closer the RPO to current time, the more expensive the solution will be to implement.

Large tier1 telecommunications providers typically have multi-million dollar budgets to implement comprehensive business continuity solutions for their billing systems using high end SAN (Storage Area Network) level solutions such as HP's Business Copy XP or EMC's SRDF (Symmetrix Remote Data Facility) , however for the smaller tier 3 telcos, such solutions are out of reach due to the considerable costs involved.

This paper will investigate how a cost effective business continuity solution can be provided for a small tier 3 telecommunications provider's billing system using standard Oracle and Unix operating system functionality without requiring the purchase of expensive SAN level

data replication products.

The diagram below provides an overview of the proposed solution:

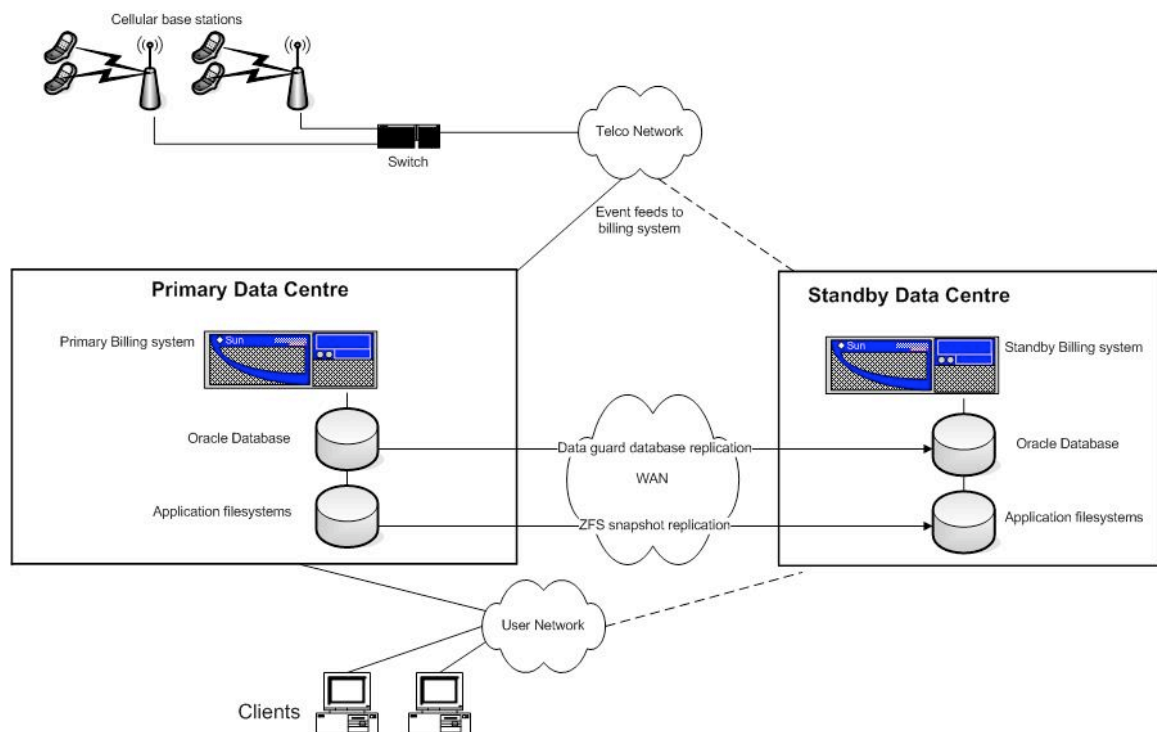


Figure 1. Overview of proposed solution

Oracle 10g DataGuard functionality will be used to replicate the database from the primary site to the remote site over a WAN (Wide Area Network) link.

Application filesystem data will be replicated by means of the snapshot facility of the new ZFS filesystem included in Sun's Solaris 10 Unix operating system.

Both of these technologies are standard functionality included with Oracle and Solaris so they do not require the purchase of additional licenses.

1.2 THESIS OUTLINE

This Thesis is composed of five chapters, each of which is introduced briefly here.

1.2.1 Chapter 1 – Introduction

This chapter contains a high level overview of the research carried out for this thesis.

1.2.2 Chapter 2 – Review of literature and research

This chapter describes the research and literature review carried out in order to evaluate the requirements and various options available for implementing a business continuity solution for a small telecommunications billing system. The advantages and disadvantages of each solution are identified as they apply to the requirements for the billing system business continuity solution. The areas covered are:

- Business Continuity - Discusses the importance of Business Continuity for an enterprise and describes at a high level how business continuity systems can be implemented to protect an enterprise from the effects of downtime. Concepts and terminology important to the understanding of business continuity systems are also outlined here.

- Overview of telecommunications billing – provides an overview of how telecommunications billing systems operate to translate telecommunications network events such as making a call or sending a text message into charges, which can then be billed, to a customer account, generating revenue for the business. The specific business continuity requirements for a billing system are also discussed, focusing on where the data is stored on the system and how it can be replicated to the remote location. Finally, constraints particular to a small, cost-conscious tier 3 telecommunications provider are described and a set of requirements for the business continuity solution are defined.
- Business continuity solutions - describes the research carried out in order to evaluate the various options available for implementing a business continuity solution for a billing system. The advantages and disadvantages of each solution are identified as they apply to the requirements for the billing system business continuity solution.

1.2.3 Chapter 3 - Design of proposed solution

Based on the research carried out in the previous chapter, a business continuity solution is proposed to meet the requirements of the billing system. This chapter describes the server hardware, operating system, database and replication technologies selected to implement the billing system in a disaster tolerant manner.

1.2.4 Chapter 4 - Implement and test proof of concept architecture

This chapter describes how the proposed solution was implemented as a proof of concept using small scale hardware in order to prove that the replication mechanisms function as required and that a production billing system can be replicated to a remote location so that it can assume billing responsibilities in the event of a failure of the primary site.

1.2.5 Chapter 5 – Conclusion

This chapter contains the conclusions arrived at through this research and recommendations for further research.

1.2.6 Appendix A

Appendix A contains pricing information on some of the hardware and software used in the proposed solution.

1.2.7 Appendix B

Appendix B contains details of how the PRIMARY and STANDBY database instances were created for the proof of concept tests in chapter 4.

2.0 REVIEW OF LITERATURE AND RESEARCH

2.1 BUSINESS CONTINUITY

Business Continuity refers to the ability of an organization to continue to function after a disruptive event.

2.1.1 The need for Business continuity

Today, a business's ability to function depends on its IT infrastructure and the data it contains. Depending on the type of business, the impact on the business caused by the loss of its IT infrastructure can range from significant – e.g. a small business that has to resort to manually processing orders on paper for the duration of the outage, to catastrophic – e.g. a bank or stock exchange which ceases to function without its IT systems.

There are very few businesses that would be able to carry on functioning as normal following a loss of their critical IT systems however such losses can and do occur all the time and regularly cause significant impacts to the bottom line of the business affected. The causes of such outages are wide ranging. Natural disasters such as fires, floods or earthquakes can damage or destroy the buildings containing the IT systems, as can terrorist acts such as the September

11th attack. On a smaller scale, local disruptions such as loss of power or communications into a building can put the IT systems beyond use. Even a disaster affecting a nearby location can affect a business, for example a gas leak or a fire nearby can result in an exclusion zone being set up that prevents employees from getting access to their place of work.

All these scenarios and countless others can result in a business losing access to their IT systems for an undetermined period of time. This downtime has a direct monetary effect on the business; a survey by Eagle Rock Alliance (Eagle Rock Alliance 2001) found that 46% of respondents estimated that 1 hour of downtime would cost them up to \$50k; while 8% estimated that 1 hour of downtime would cost them more than \$1 Million.

Percentile	Cost per 1hour of downtime
46%	Up to US \$51k
28%	Between US \$51k and US \$250k
18%	Between US \$251k and US \$1M
8%	More than US \$1M

Table 1. 2001 cost of downtime Survey – direct monetary costs

The main factors contributing to these costs are:

- Loss of clients (lifetime value of each) and market share
- Fines, penalties, and liability claims for failure to meet regulatory compliance
- Lost ability to respond to subsequent marketplace opportunities
- Cost of re-creation and recovery of lost data
- Salaries paid to staff unable to undertake billable work

- Salaries paid to staff to recover work backlog and maintain deadlines
- Employee idleness, labor cost, and overtime compensation
- Lost market share, loss of share value, and loss of brand image

(IBM, Brooks, Leung et al. 2007)

Aside from the direct monetary effects of IT systems downtime, there are also indirect effects that may not have an immediate monetary impact on the business but would prove critical to its survival in the mid to long term following a systems outage. These are summarized in table 2 below.

Percentile	factors critical to business survival
40%	Customer service or expectations
17%	Competitive advantage
16%	Public image
12%	Regulatory requirements
11%	Contractual obligations
2%	Share holder satisfaction
2%	Other

Table 2. 2001 cost of downtime Survey – factors critical to business survival

The principle impact on a business's survival following an outage would be due to loss of consumer confidence – if an existing customer perceives an enterprise as not being able to

handle their business, they will be inclined to take their business elsewhere. A loss of public image following a high profile outage would make it harder to win that business back or generate new business.

Contractual and regulatory requirements are another key factor. Recently introduced legislation such as Sarbanes-Oxley and Basel II have strict requirements for providing business continuity for IT systems. Often contracts between service providers and customers will define service level agreements that must be maintained in order to avoid breaching the contract terms. This can result in substantial penalties or fines in the event of breach of contract or non-compliance with regulations due to loss of critical IT systems or data.

For today's enterprise, IT systems and the data they contain are critical to the functioning of the enterprise. There are many threats that could potentially take these systems offline, however a properly planned business continuity solution can mitigate these risks and allow a business to continue functioning even in exceptional circumstances.

2.1.2 Aspects of Business continuity

There are three main aspects to providing business continuity for IT systems, they are:

- High Availability
- Continuous Operations
- Disaster Recovery

(IBM, Brooks, Leung et al. 2007)

2.1.2.1 High Availability

High Availability refers to the ability of a system to continue to provide service despite the failure of one of its component parts. This is typically provided by a highly reliable, redundant hardware platform, which has been designed to automatically handle the failure of a critical component without causing a loss of service.

An example of such a server is Sun Microsystems' Enterprise T2000, which has redundant disks, power supplies and fans, which can be replaced while the system is running. Failed processor cores can be dynamically de-allocated by the operating system and ECC (Error checking and correction) can dynamically detect and correct memory errors. Operating system features such as multi-path I/O (MPXIO) and IP Multipathing (IPMP) provide multiple redundant paths for storage and networking traffic. Using these and other technologies, Sun Microsystems claim that the T2000 server can approach 99.999% availability, or less than 5 minutes unplanned downtime per year. (Sun Microsystems 2007a)

Technologies such as clustering can also be used to increase availability. Here an application runs on two or more servers in either an active/standby or an active/active configuration. In active/standby, the application runs on one primary server. In the event of a failure of the primary server, the application is automatically switched to the standby server, thus maintaining the availability of the application.

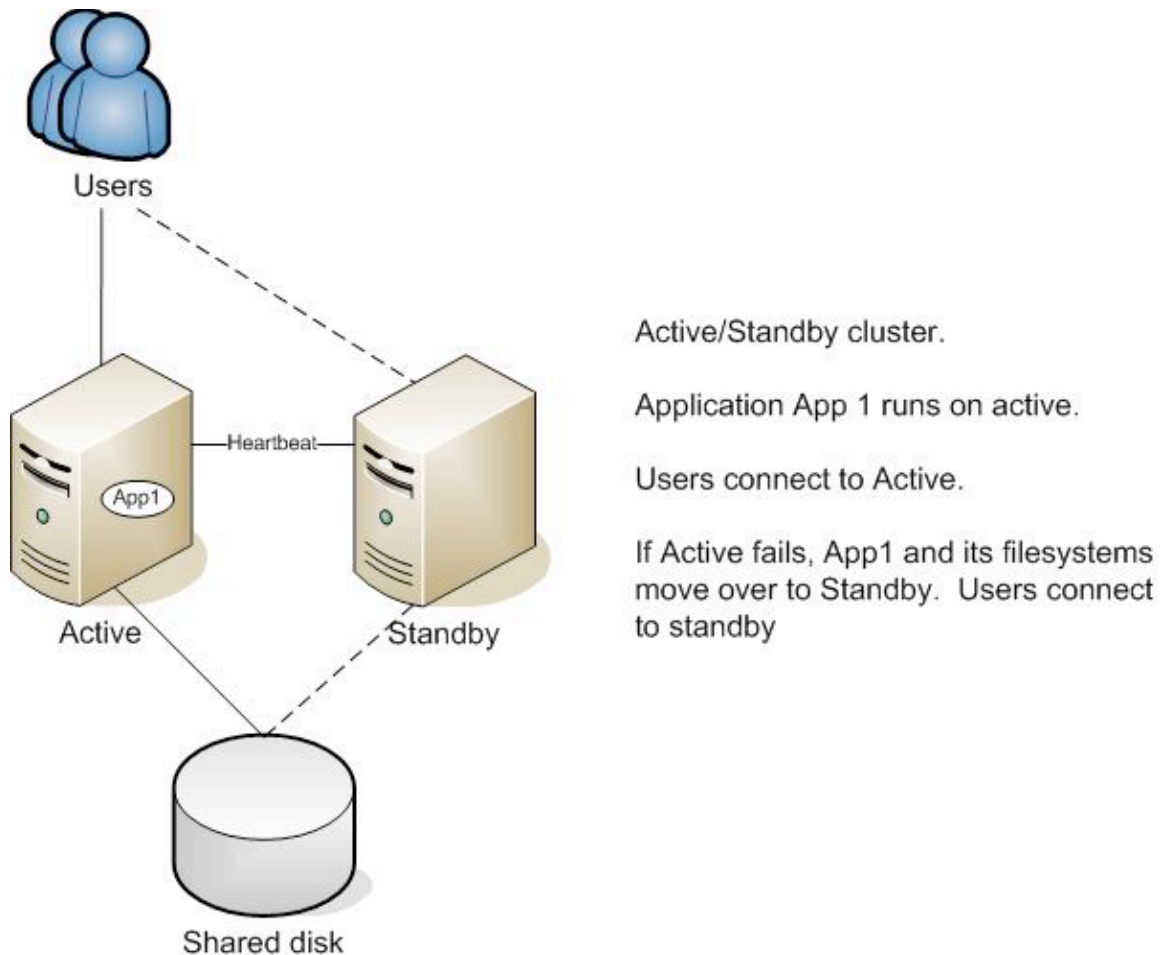
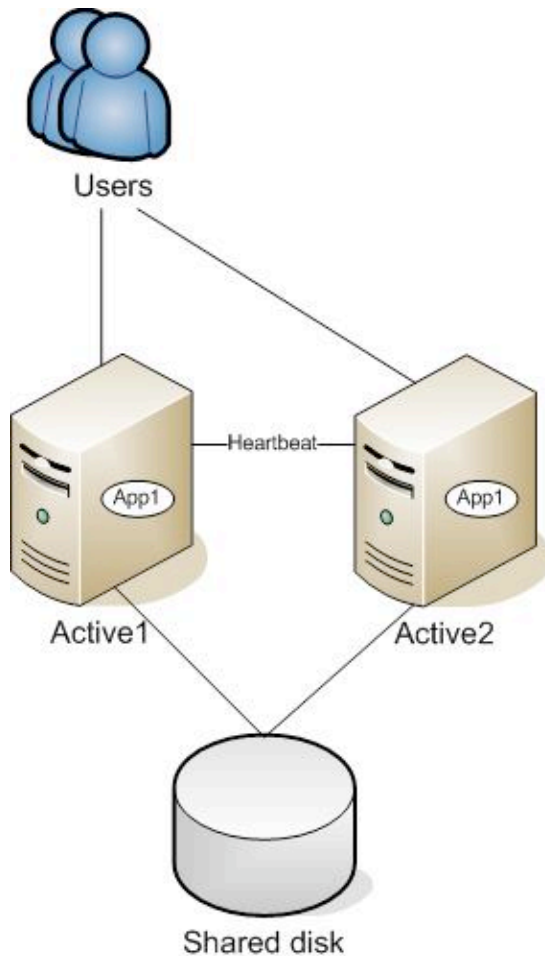


Figure 2. Active/Standby cluster

In an active/active configuration, the application runs on all servers in the cluster at the same time, each handling a portion of the load. In the event of a failure of one of the servers, the remaining servers in the cluster can assume the load of the failed server, maintaining the availability of the application for the users.



Active/Active cluster.

Application App 1 runs on both nodes. Each node carries a portion of the workload.

Users connect to both nodes.

If Active1 fails, users on affected node connect to remaining node.

Figure 3. Active/Active cluster

Resilient servers and high availability clusters can protect the application from the effects of unplanned downtime, such as a server or disk failure, however it is sometimes necessary to have planned downtime for situations such as Server Operating System upgrades, hardware upgrades, application patches etc. Generally these tasks are scheduled for times when the availability of the system is not as critical, such as outside of office hours, however in today's global, networked economy, 24 x 7 operation is often required, meaning there is no non-critical time for the system. In this case the system needs to be configured for Continuous Operation.

2.1.2.2 Continuous Operation

Continuous Operation allows a system to provide service at all times, even during scheduled application, OS or hardware upgrades. This can be accomplished using clustered servers as previously described. Here, what is known as a rolling upgrade can be performed on a cluster of servers running an application. One server has its workload transferred to other servers in the cluster, and then it is taken offline to allow the upgrade work to take place. Once it is upgraded, it is brought back online and begins to process its workload again. The process is repeated for all servers in the cluster until the entire cluster has been upgraded.

High Availability and Continuous Operation only address the availability of the system within a single site. Even with resilient hardware and high availability clustering it would still be possible for a problem affecting the location hosting the servers to make the application unavailable. Certain risks affecting the entire site, such as power outages, can be mitigated through the provision of uninterruptible power supplies (UPS) and backup generators, however there are other risks that can't be adequately mitigated on a single site – e.g. fire, earthquake,

flood, terrorist act etc. To mitigate these risks, the only solution is to duplicate the IT systems on a site sufficiently remote from the primary site not to be affected by any disaster that might befall the primary site. The provision of such a facility is known as Disaster Recovery.

2.1.2.3 Disaster Recovery

Disaster Recovery (DR) involves providing some form of facility for transferring the operations of a business's IT systems from its primary location to a remote standby location should the primary location be unable to support normal operations.

There are a number of approaches that can be taken to implement a business continuity solution for an IT system. The approach taken will depend on a number of factors, chief of which are:

- Recovery Time Objective (RTO) – what is the maximum amount of time allowed between the declaration of a disaster and having the system up and running on the standby site?
- Recovery Point Objective (RPO) – to what point must the system be recovered?
What if any data loss is acceptable?
- Cost – how much is budgeted for the provision of the business continuity system?

These factors are interrelated – the shorter the RTO and the closer the RPO to current time, the more expensive the solution will be to implement.

In some cases it may not be critical to have an IT system back up and operating within hours or minutes of losing the primary site. If the RTO is greater than 24 hours, meaning the application can be offline for greater than 24 hours, and the RPO is greater than the time since the last backup, it would be sufficient to implement a DR solution based on shipping the nightly backup tapes from the primary site to the standby site. Here, in the event of a disaster affecting the primary site, the IT systems could be restored onto standby hardware from the most recent set of backup tapes. This approach offers the lowest cost, however any data changes made since the most recent backup would be lost.

For some businesses it may be acceptable to be offline for more than 24 hours and to lose a day's worth of data, however for others, even minutes of downtime and seconds worth of lost data could result in significant monetary loss. For such situations some form of online replication is required between the primary and standby site. This involves configuring the standby site with similar servers and storage to the primary site and replicating any changes made to the storage on the primary site to the storage on the standby site. This means that any changes made to data on the primary site are automatically made to the data on the standby site, therefore keeping the data sets on both sites in sync. In the event of a failure of the primary site, the application can be quickly brought up on the standby site to resume service with minimal or no data loss. This is a significantly more costly option than tape backup, as it requires specialized storage hardware/software and a high capacity network link between the sites.

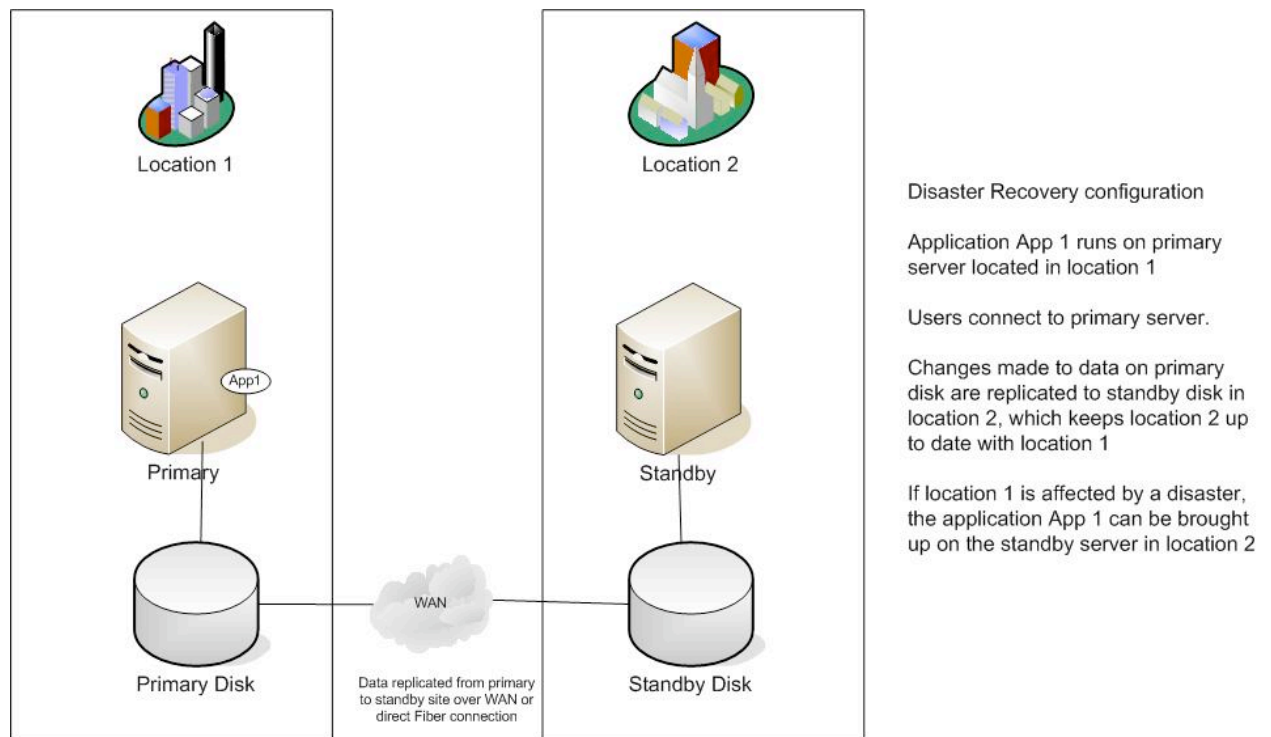


Figure 4. Disaster Recovery

For replication of data between the two sites there are two methods that need to be taken into account – Synchronous replication and Asynchronous replication.

- Synchronous replication means that when a data block is written to disk by an application on the primary site, it must also be confirmed written to disk on both the primary and the secondary site before the application is told that the write has

been completed. This approach guarantees that data will be consistent between the two sites, so in the event of a disaster there will be no data loss as every piece of data written would have been written to both sites. This allows the DR solution to have a RPO of 0, however it comes at a cost – the writing application must wait for both the local and remote writes to complete before proceeding. If the remote site is far away, or if there is network contention, this can have a significant performance impact on the application. Very high speed network links can offset this performance impact, however over a certain distance, latency or the time taken for a packet of data to travel between two points, begins to play a significant part, to the point where it begins to have an unacceptable impact on the performance of the application. A network outage between the two sites would have the undesired effect of halting the application at the primary site, as it could not complete its remote writes.

- Asynchronous replication means that when a data block is written to disk by an application on the primary site, it must only be confirmed written to disk on the primary site before the application is told that the write has been completed. The corresponding write to disk on the secondary site is allowed to take place in the background. This approach has no performance impact on the application and no limit on the distance between sites, however there is a potential for some data loss in the event of a disaster. e.g. if an application's write was confirmed on the primary site, then the site went down before the write to the standby site had been shipped across the network, then that write would be lost. Buffering technologies

can be used to prevent data loss in the event of momentary network outages, however there is still some potential for data loss.

2.1.3 Conclusion

There is no one solution that fits the business continuity requirements of every business. Each business is different and is affected by downtime in a different way. What may be an acceptable risk for one could be intolerable for another; a reasonable cost for one could be unfeasible for the other.

Because of this, each business must assess what is an acceptable level of downtime and of data loss, how much downtime and data loss will cost them, and weigh these costs up against the cost of providing a business continuity solution.

2.2 OVERVIEW OF TELECOMMUNICATIONS BILLING

This chapter gives an overview of the main processes carried out by a telecommunications billing systems in order to translate telecommunications network events such as making a call or sending a text message into charges, which can then be billed to a customer account, generating revenue for the business.

The specific business continuity requirements for a billing system are also discussed, focusing on where the data is stored on the system and how it can be replicated to the remote location. Finally, constraints particular to a small, cost-conscious tier 3 telecommunications provider are described and a set of requirements for the business continuity solution are defined.

2.2.1 What Billing Systems do.

In a telecommunications network, when a subscriber places a call the network element or switch handing the call produces a record called a 'Call Detail Record' (CDR). The CDR will contain information such as: the originating number, the terminating number, the billed number if different from the originating number, the time the call was connected, the time of hang up, whether operator assistance was used and other provider specific flags. (Hunter, Thiebaud 2003)

CDRs from the various network elements and switches are collected by a mediation system, possibly being translated into a common format for processing by the billing system.

Multiple CDRs are gathered into a CDR file and these are sent to the billing system. Additional CDR feeds from external systems such as TAP roaming feeds from partner networks also go through the rating process.

The incoming CDRs to the billing system are placed in an *input* filesystem on the billing server. A rating process opens the CDR files and performs the following tasks:

- Associates the CDR with a particular customer account based on the originating number, a process known as guiding.
- Determines the call length from the connect time and hang up time.
- Determines a rate for the call based on the number called, time of day and the company's rate plan table, along with any complex tariffs such as free minutes or friends and family discounts.
- Writes the charge information generated by the CDR into the database.

Once all the CDR records contained within a CDR file have been processed, the file is removed from the *input* filesystem and moved to an *archive* filesystem. This allows the CDR files to be moved to offline storage in case they need to be held for a period of time to meet regulatory requirements. Occasionally, CDRs fail to rate correctly, either due to errors in the CDR due to incorrect network element configuration or incorrect rate plan configuration on the billing system. Should this happen, the failed CDR files are moved to an *error* filesystem for manual exception analysis and eventual reprocessing once the error is rectified.

Rating is generally a continual process, with CDR files being received 24 x 7 and processed by the rating engine as they arrive.

In addition to the continual rating process, the billing system also runs periodic batch jobs known as 'bill runs', which are used to aggregate all the monthly usage charges for a particular customer into a single bill which is then sent to the customer for payment. Typically a number of bill runs are run each month, each catering for a subset of the customer base. The bill run performs the following tasks:

- Identify which customers are associated with this particular bill run
- Identify unbilled charges stored in the database, which are associated with customers attached to this bill run.
- Generate any recurring charges such as line or equipment rental that apply to the customers.
- Aggregate usage and fixed charges for each customer.
- Apply any necessary taxation charges
- Generate an invoice for each customer.
- Mark the charge records contained in the database as 'billed' so they are not billed for a second time.
- Update customer and accounting tables with credit/debit information.
- At the end of the bill run, the invoices produced are sent to a printing house for printing and distribution to the customers.

In addition to the rating and billing functions, the billing system is also responsible for providing management and financial reports to various departments within the business. It also needs to interface with various other Operations Support Systems (OSS) such as Web Self Care systems or Customer Relationship Management (CRM) systems so that customer service

representatives (CSRs) can query usage records and invoices on behalf of customers, or create new accounts and sell services to new customers.

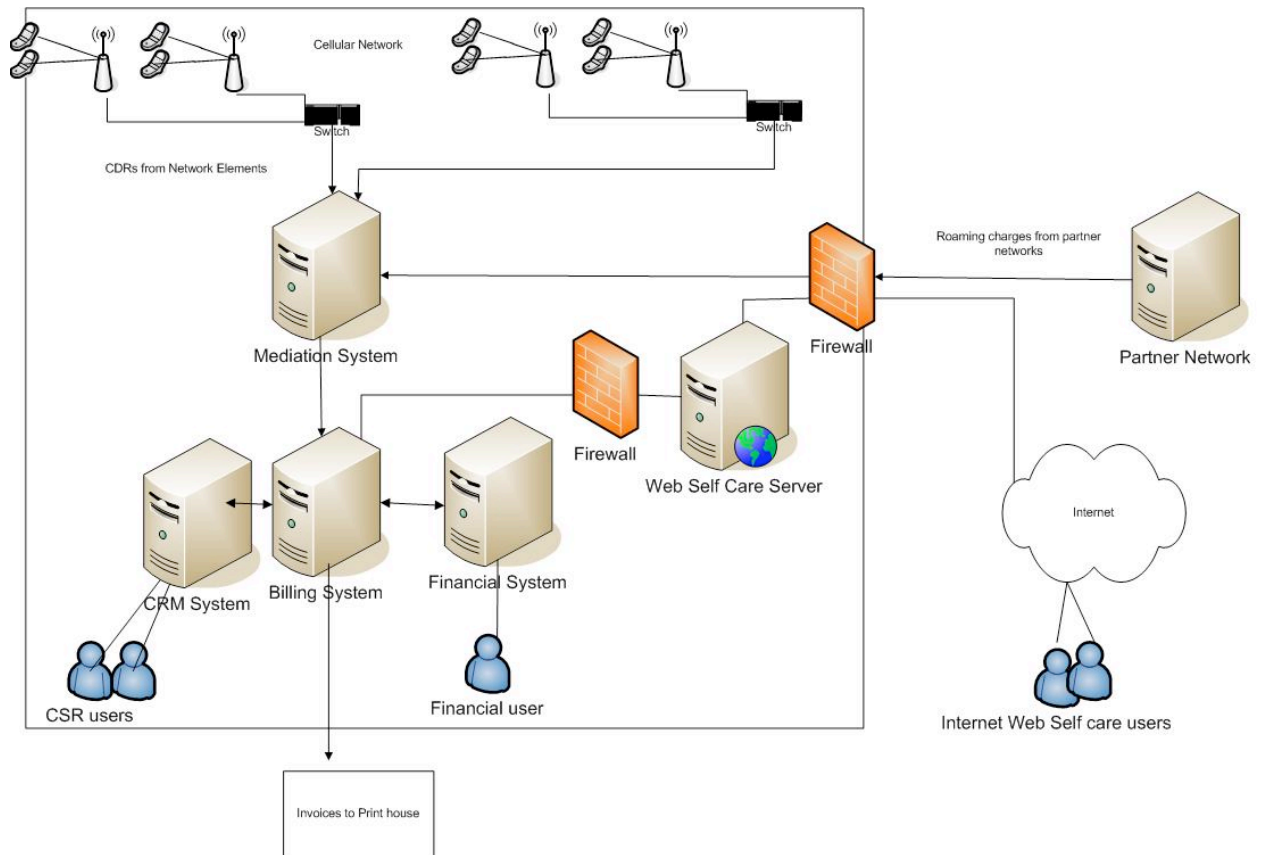


Figure 5. Billing System Overview

2.2.2 Billing Systems architecture.

Billing systems are generally implemented as a Unix application, which provides the rating and billing functionality and a back end database used to store the rated charges, customer and account details and generated invoices.

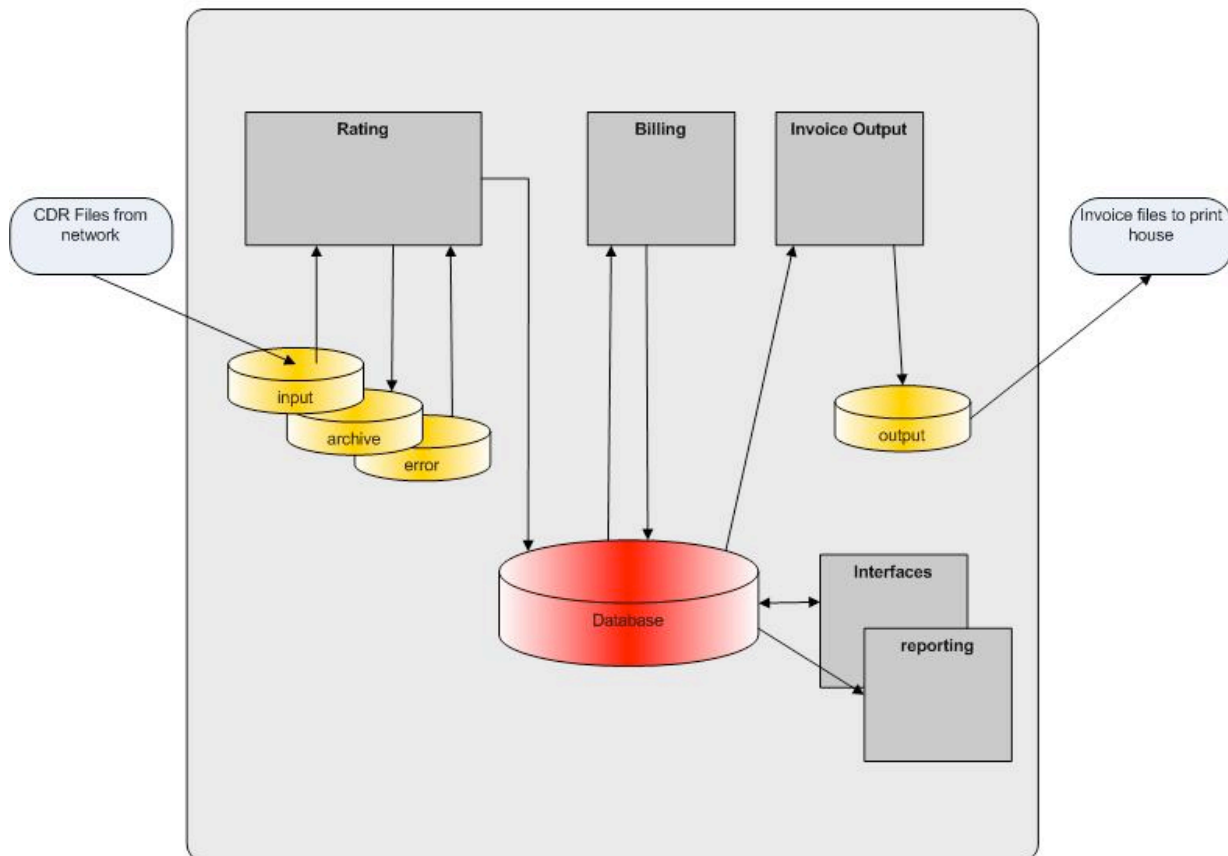


Figure 6. Billing System Architecture

Figure 6 above shows the major components of a billing system. The rating engine takes the incoming CDR files from the *input* filesystem, rates them then stores the charges in the database. Incoming files are then moved from the *input* filesystem to the *archive* or *error* filesystem depending on whether rating was successful.

The Billing engine reads charge and customer data from the database, generates invoices and writes them back into the database. The Invoice output process then extracts these invoices from the database to an *output* directory from where they are shipped to a print house for printing.

The interfaces module handles connections from operator graphical user interface (GUI) clients, which are used to control the configuration of the system, and the scheduling of batch operations. External interfaces such web services or EAI (Enterprise Application Integration) adapters to CRM and provisioning systems are also handled here.

The reporting module provides functionality for running reports against the system.

The database is an Oracle database used for storing all customer information, charges, invoices rate plans and configuration information for the billing system.

2.3 BUSINESS CONTINUITY AND THE BILLING SYSTEM.

2.3.1 Business continuity requirements

The billing system is critical for the functioning of a telecommunications service provider – without it they cannot generate revenue from the services they provide. Because of this it is necessary that the billing system should be protected from the effects of downtime.

In order to determine the Business continuity requirements for the billing system, we need to consider it from the standpoint of recovery time objective (RTO), recovery point objective (RPO) and cost.

The billing system performs essentially a back end function for the business. If it goes down for a short period of time, while inconvenient, it won't stop the business from functioning. The switches and network elements will continue to connect and process calls, the CDRs they generate will back up on the switches or mediation systems, so no revenue will be lost and no customer calls will be dropped. Once the billing system comes back on line it can begin to rate any backlog of CDR records. Billing is a periodic batch task, so it is scheduled to run within a particular window. If the billing system is unavailable for a part of this window it could delay the execution of the bill run, however it will be able to take place at a later stage without too much inconvenience. This gives the billing system a bit of leeway when it comes to availability – it can be down for a short period of time without any major negative impacts, however if it is down for an extended period of time, a backlog of processing tasks will begin to build up – CDR files

to be rated, bill runs to be executed and reports to be run. Billing systems are hosted on large, expensive servers so while they are generally configured to have some spare processing capacity to cope with backlogs and periodic usage spikes, they tend to be configured to run at high levels of utilization to maximize the company's hardware investment. If the backlog grows too large, it may not be possible for the billing system to clear the backlog while at the same time handling its normal daily workload. Specifying an RTO of 4 hours would require that the system be available for normal processing within 4 hours of an outage. This would ensure that the system would not be down for long enough to generate a backlog of processing that it could not clear within the day.

Assuming that during the network's peak hour, each subscriber will make up to three calls, then on a small billing system with 1 Million customers, this translates into up to 3 Million calls. Even at the lowest per minute rates, this translates into significant revenue per hour. If the billing system were to lose a couple of hours of rated data this would result in significant financial loss for the company. It may be possible to regenerate the lost CDRs from the mediation system and re-submit them for processing, however this may not be a straightforward procedure. For this reason a fairly recent RPO should be specified, ideally no more than a few minutes of data should be lost from the billing system in the event of a disaster.

The amount of cost that can be born for the implementation of a business continuity solution will depend very much on the company concerned. Tier 1 telecommunications providers such as state carriers and major multi-nationals can afford to pay much more than smaller regional carriers or Mobile Virtual Network Operators (MVNOs). Often the small tier 3 carriers are startups who are trying to compete with the bigger more established players by offering lower prices. For these companies, costs are a major concern as they have much lower amounts

of operating capital than the established companies. Another significant factor that affects small tier 3 providers is that of network bandwidth costs. Large telecommunications providers own high capacity networks between their equipment locations, – this is their core business, after all. So using some of their own network capacity for business continuity purposes is not a major cost factor. Smaller providers however may not necessarily have their own network; often they are virtual network operators, buying capacity on one of the larger provider's equipment. This means that network connectivity between sites needs to be purchased at market rates, which makes it a requirement that a business continuity solution should be economical in its use of network bandwidth.

2.3.2 Replication requirements

There are two major components that need to be replicated from the billing system on the primary site to the secondary site – filesystem data and database data.

The filesystems for the billing system contain both dynamic data and relatively static data. The key filesystems required for billing system operation are described in table 3 below.

Filesystem	Usage	Comment
/oracle	Oracle database binaries	Fairly static, only changes when database software is patched. Only needs to be replicated when changed.
/opt/billing	Billing software binaries	Fairly static, only changes when billing software is patched. Only needs to be replicated when changed.
/billing/bilprd1	Production billing instance	Dynamic, contains log files and configuration data for billing application instance.
/billing/bilprd1/input	Incoming CDR files for rating	Dynamic, needs to be periodically replicated.
/billing/bilprd1/archive	Archive directory for files that have been rated.	Dynamic, does not need to be replicated as files that are in this directory already exist as charges in the database.
/billing/bilprd1/error	Error directory for files that have not rated successfully	Dynamic, needs to be periodically replicated.
/billing/bilprd1/output	Output directory for invoices produced by bill runs.	Dynamic, does not need to be replicated, as invoices in this directory are also contained in the database.
/u01	Oracle database data files.	Dynamic – critical that their contents are continually replicated to the remote site.
/u02		
/u0n		

Table 3. Billing system filesystems

The Oracle database contains the majority of the data critical to the operation of the billing system. This needs to be continuously replicated to the standby site.

2.3.3 Business continuity requirements for a small tier 3 provider.

For a small, tier 3 telecommunications provider, the billing system business continuity requirements are as follows:

- Recovery Time Objective (RTO) – The billing system should be back up and running within 4 hours of any outage.
- Recovery Point Objective (RPO) – The billing system should lose no more than 5 minutes worth of data in the event of an outage.
- Cost – cost is an issue, so the business continuity solution should aim to minimize costs while not impacting the availability requirements above.

Disaster recovery capability is required to protect from an outage affecting the primary site and to comply with regulatory requirements. This should meet the RTO and RPO requirements detailed above.

High Availability clustering is not necessarily required given the 4-hour RTO. Redundant Unix server hardware is capable of reaching availability levels approaching 99.999%, which reduces the likelihood of an outage due to a hardware failure. Hot swap components allow certain hardware replacements without any downtime. Any extended outage due to hardware failure on the primary site can be managed by switching operations over to the standby site.

Continuous operation is not required as planned downtime for hardware and software upgrades can be scheduled to occur at periods of low system utilization. Major system upgrades

on the primary site can be facilitated by switching operations from the primary to the standby site, upgrading the primary site then resuming operations on the primary site.

The following hardware is specified as a sample hardware configuration capable of supporting up to 1 Million subscribers. The business continuity solution needs to be compatible with the hardware configuration below.

Requirement	Value	Comment
Subscribers	1 Million	
Calls/subscriber/day	8	Combination of Voice calls, SMS, data.
CDR History retention	6 months	Rated CDRs are stored for 6 months in the database before being archived off
Billing windows	6 x 8hr	6 x 8hr long bill runs per month
Platform	Sun T2000	
CPUs	8 x 1.2GHz cores	1 x Sun UltraSparc T1 processor
RAM	32GB	
Disk	3 TB	
Operating system	Solaris 10	

Table 4. Billing system hardware requirements

2.4 BUSINESS CONTINUITY SOLUTION RESEARCH

This chapter describes the research carried out in order to evaluate the various options available for implementing a business continuity solution for the billing system. The advantages and disadvantages of each solution are identified as they apply to the requirements for the billing system business continuity solution.

There are two main areas to consider when looking at technologies to replicate a billing system to a remote location. These are database replication, where the Oracle database contents are replicated to the remote site and filesystem replication, where the application and supporting filesystems are replicated to the remote site.

This section will evaluate the technologies available to perform both tasks and select the most suitable option for a small billing system.

2.4.1 Database replication

In order to provide disaster recovery for an Oracle database, it is necessary to find some mechanism to replicate the contents of the Oracle database from the primary location to the secondary location. There are a number of methods of replication available, but they fall broadly into two categories:

- Oracle database level replication
- Remote mirroring based replication. (Oracle n.d. a)

Oracle database level replication involves replicating the database through Oracle database technologies. The database instance itself is responsible for copying changed data from the primary database instance to the standby database instance.

In remote mirroring based replication, the Oracle database is unaware of the replication going on, instead the changed data blocks are replicated at a lower level such as when they are written to the filesystem, logical volume or disk.

2.4.1.1 Oracle level replication

At the database level there are several Oracle technologies that can be used to replicate data from a primary to a standby database. These are:

- Oracle Data Guard
- Oracle Streams

- Oracle Advanced replication
- Oracle Recovery Manager
- OSCP Validated remote mirroring.

(Oracle n.d. a)

In addition to the Oracle technologies listed above, there are a number of third party software vendors that produce Oracle database replication software. These include:

- Quest Software's SharePlex for Oracle
- Ixion's IxPropagator

2.4.1.2 Oracle Data Guard

Oracle Data Guard is a software solution for managing standby databases to protect the primary database from the effects of downtime. It operates by synchronously or asynchronously transmitting transactional redo data from the primary database to the standby database. The redo data is applied to the standby database, which ensures that it is kept up to date with the primary on a block-by-block basis. As data guard functions on the Oracle level, it can perform additional Oracle level checks on the data it applies to the standby database, which protects against accidental data corruption that may occur as data are shipped between the primary and standby sites. As Oracle data guard only copies across the database redo logs, it requires less network bandwidth than a remote mirroring solution which would need to copy across changes to data files and control files as well as online and archived redo logs. An Oracle internal analysis of their corporate email systems (Oracle n.d. b) found that 7 times more data was transmitted over

the network and 27 times more I/O operations were performed when using a remote mirroring solution compared to Oracle Data Guard.

Another major advantage of Oracle Data Guard is that it allows you to open the standby database on the remote site for reporting purposes even while it is being updated with redo data from the primary site. This means that the standby system can actually be used to offload certain tasks from the primary database, a factor that is attractive to CIOs who do not like to see low levels of utilization on their expensive IT hardware. Reporting and data warehouse extracts can impose significant loads on a billing system, often requiring them to be scheduled outside of bill runs or peak rating time. This may not always be suitable for the consumers of the reports who want to have up to date data on their schedule, not the billing system's schedule. By moving some of the reporting requirement to the standby system, better operational efficiency can be achieved. Data Guard has always had the facility to open a standby database in read-only mode for reporting purposes, however this sometimes meant that the redo apply had to be stopped for the duration of the reporting. Oracle 11g now offers a new feature called the 'snapshot standby' database, which allows a standby database to be opened read-write for reporting or testing purposes. When the snapshot standby database is converted back into a physical standby database, redo data that has been received from the primary is applied to the standby database and any changes made while it was a snapshot standby are discarded. This enables the production data to remain in a protected state at all times, while still allowing the standby database to be used for reporting or testing purposes. (Oracle 2007a)

Another new feature in Oracle 11g Data Guard is 'Transient logical standby'. This feature allows a DBA to perform an Oracle upgrade on a standby database, while the primary is still in operation. Once the upgrade is complete, it can be converted back into a physical standby

database. This feature will facilitate rolling upgrades of the production Oracle database with minimal downtime, a feature that will increase the overall availability of the billing system.

Data Guard supports both synchronous and asynchronous replication between primary and standby database. It also provides a feature known as ‘delayed log apply’ whereby an artificial delay can be introduced when applying the redo logs, which allows the secondary database to be kept a certain period of time (e.g. 4 hours) behind the primary database. This offers additional protection against the effects of human error – if a DBA accidentally dropped a table on the primary database, that change would not be instantly propagated to the standby database, therefore allowing time to recover from the error.

Oracle Data Guard is included in the Oracle Database Enterprise Edition license so there is no additional cost on top of the Oracle licenses already purchased to implement a data guard based solution.

2.4.1.3 Oracle Streams

Oracle streams is a replication technology that is included in Oracle Enterprise Edition. It supports both unidirectional and bi-directional replication of all or a portion of an Oracle database, so it is suitable for use in distributed database configurations. As Streams can replicate data from one database to another, it can be used to implement high availability/disaster recovery features, however Streams is primarily intended for information sharing and Data Guard is designed primarily for Disaster Recovery. (Oracle n.d. c)

2.4.1.4 Oracle Advanced replication

Oracle Advanced Replication, like Streams, provides a way to replicate database objects between multiple databases. It can provide High Availability and Disaster recovery features due to the fact that it maintains multiple copies of database objects, however like streams, HA and DR are not its primary focus.

2.4.1.5 Oracle Recovery Manager

Oracle Recovery Manager (RMAN) is Oracle's tape backup and recovery solution, which allows the contents of a database to be backed up to tape while the database is open and operational. RMAN can provide DR facilities, in that periodic backups to tape can be shipped to the remote site to allow the production database to be restored in the event of a disaster, however as with any tape based backup solution it is incapable of being used to provide a short recovery time objective (RTO). If you loose your primary site and RMAN is your only DR option, you will only be able to recover the primary database to the point of the last tape backup. This makes it unsuitable for use as a primary DR option if a short RTO is required, however if a longer RTO is acceptable (up to 24 hours) then RMAN tape based backups provide a cost-effective, reliable solution. . (Oracle n.d. a)

2.4.1.6 Quest Software's SharePlex for Oracle

SharePlex operates in a similar manner to Oracle Data Guard in that it makes use of change information contained in the database redo logs to replicate data changes from the primary site to the secondary site. Unlike Oracle Data Guard, which comes included in an Oracle Enterprise Edition license, SharePlex is a separately licensed product and so will incur additional license costs to deploy. It claims to be easier to install and manage than data guard, however recent releases of data guard have become easier to configure and manage thanks to integration with Oracle's Enterprise manager. (Quest 2007)

2.4.1.7 Ixion IxPropogator

Like SharePlex, IxPropogator operates in a similar manner to Oracle Data Guard in that it makes use of change information contained in the database redo logs to replicate data changes from the primary site to the secondary site. It is also a third party commercial product, which means it adds extra cost to deployment compared to Data Guard. It claims to offer significant performance gains over Data Guard, however evidence of this is hard to find. (Ixion 2005)

2.4.2 Remote mirroring based database replication

Remote mirroring based replication refers to replicating the Oracle database at the host Operating System (OS) or hardware layer. This differs from database level replication in that the replication methods are not application aware – i.e. they just replicate any changes they see made to data blocks on the primary server across to the standby server.

Methods of remote mirroring can be grouped into two main categories – host based and hardware based. As the methods used to perform remote mirroring of an Oracle database can also be used to replicate filesystem data, they are looked at in the next section – Filesystem Replication.

2.4.2.1 OSCP Validated remote mirroring

The Oracle Storage Compatibility Program (OSCP) is a program that Oracle uses to validate storage hardware from various vendors for use with the Oracle database. There are a number of vendors that provide hardware-based remote mirroring solutions that can be used to replicate Oracle databases at the file level, and Oracle has certified some of them for use with the Oracle database. The disadvantage of remote mirroring technologies is that they are not application aware, so they can not perform the same level of checking on replicated data as Oracle Data Guard can, also as discussed in the section on Data Guard above, they can require significantly more network bandwidth than Data Guard. (Oracle n.d. a)

2.4.3 Filesystem replication

In addition to replicating the Oracle database containing the billing system data, it is also necessary to replicate any filesystems that contain application binaries, configuration files and input/output data files. This section will look at the various technologies that can be used to perform such replication.

The technologies available for filesystem replication fall under two main categories – hardware based, where the replication is done at a low hardware level e.g. by the disk array hardware, and host based, where the replication is carried out at a higher level, either by an Operating System (OS) component, or through some third party software application running on the host.

2.4.3.1 Hardware based mirroring

Hardware based replication, also known as Storage Array based mirroring, refers to the ability of a disk array to track changes made to the data blocks it stores, and transfer these changes to a remote disk array where they can be applied in order to keep the data synchronized on both arrays. Typically direct fiber-channel connections are used to carry mirroring data between primary and standby sites, however it is also possible to use an IP based network in situations when a direct fiber based link is not available. Hardware based mirroring supports both synchronous and asynchronous replication.

As hardware based replication is carried out at the storage array hardware level, each storage array vendor has their own hardware based replication solution for their mid-range to high-end enterprise storage arrays. Products from the main disk array hardware vendors, which support the Solaris platform as required by the billing system, are described below:

Hewlett Packard (HP) categorizes its storage array products as ‘entry level’, ‘mid-range’ and ‘high end’. The entry-level disk arrays such as the Modular Smart Array (MSA) family, do not support the Solaris Operating System so are not discussed further here. The mid-range Enterprise Virtual Array (EVA) and the high-end Storage Works XP disk arrays support Solaris

and offer capacities from 56TB to 745TB. Hardware based remote mirroring software known as ‘Continuous Access EVA’ and ‘XP Continuous access’ is available for the mid-range and high-end disk arrays. (HP 2008)

Sun Microsystems categorizes its storage array products as ‘workgroup’, ‘midrange’ and ‘data center’ disk arrays. Apart from the mainframe compatible Shared Virtual Array, all arrays support the Solaris operating system and capacities range from 292GB at the low end, to 247PB at the high end. StorageTek 9990 Universal Replicator is a hardware based remote mirroring solution for the data center class 9985 and 9990 arrays. The workgroup and midrange class disk arrays do not have a hardware based remote mirroring product. (Sun Microsystems 2008d)

IBM categorize their storage array products as ‘entry level’, ‘mid-range’ and ‘high end’. The entry-level DS3000 series disk arrays do not support the Solaris Operating System so are not discussed further here. (IBM 2008a). The mid-range DS4000 class disk arrays support Solaris and offer capacities ranging from 500GB to 89.6TB and support remote mirroring through the ‘enhanced remote mirroring’ option. This provides Metro mirroring (asynchronous) and Global Mirroring (asynchronous), although synchronous mirroring is only supported up to a maximum distance of 10km. (IBM 2004). The high-end DS6000 and DS8000 class disk arrays support Solaris and offer capacities ranging from 292GB to 512TB and support remote mirroring through the Metro mirror (asynchronous) and Global Mirror (asynchronous) software. (IBM 2008d).

EMC storage products fall into two main categories – the high-end Symmetrix and the mid-range Clariion products. The Clariion arrays range in capacity from 38TB to 353TB (EMC 2008a) and can support remote mirroring through the MirrorView (EMC 2008d) product, which

supports both synchronous and asynchronous replication. The high-end Symmetrix systems range in capacity from 180TB to 1054TB (EMC 2008b) and can support remote mirroring through the SRDF (Symmetrix remote data facility) product, which supports both synchronous and asynchronous replication. (EMC 2008e).

2.4.3.2 Host based mirroring

There are two kinds of host based mirroring, volume mirroring and host based replication. Volume mirroring is where the disks on a host are mirrored using a host based logical volume manager such as Veritas Volume manager (VxVM) or Solaris Volume manager (SVM), with one side of the mirrored disks located remotely from the local set of disks. Every write that is made to the local disk is also made to the remote disk over a fiber channel connection. This approach is generally limited to synchronous operation, so the remote set of disks cannot be located more than a few tens of kilometers away from the local disks. (Oracle 2005a)

Host based replication is where specialized filesystem drivers or volume manager components in the primary server intercept local writes, package them up in logical messages and send them synchronously or asynchronously over an IP network to the remote hosts, where they are applied to the remote disks. There are a number of commercially available products that can perform host-based replication for the Solaris platform. These are described below:

Sun StorageTek Availability Suite is a Sun product that offers both point in time copy and remote mirroring features. (Sun 2008a) A point in time copy allows you to capture a consistent snapshot of data at a particular time. This snapshot can then be used for a number of different purposes, e.g. it can be mounted by another system for test purposes, or can be used to provide a rollback facility in the event that it was necessary to back out some upgrade or change. Point in time copies can also be used to facilitate tape backups – a point in time copy can be taken on one server then mounted on another server to be written out to tape without impacting the performance of the primary server during the backup period. This approach also eliminates the possibility of an inconsistent backup, which could occur if a tape backup was taken of a live

filesystem with constantly changing data. StorageTek Availability Suite also provides both synchronous and asynchronous remote mirroring over an IP WAN link. Pricing for the StorageTek Availability Suite is approximately US\$ 10,000 per TB of master disk. (Appendix A1.3)

Veritas Volume replicator from Symantec is an extension to the Veritas Storage foundation product, which contains the Veritas Filesystem (VxFS) and Volume Manager (VxVM) logical volume manager products. It allows VxVM volumes on one system to be replicated to identically sized volumes on a remote system over an IP network. Both synchronous and asynchronous replication modes are provided, along with a facility to provide zero data loss replication over any distance. It accomplishes this by means of a third site known as a 'bunker' site. Data are replicated synchronously to a Storage Replicator Log (SRL) volume on the bunker site and asynchronously to the standby site. This feature means that asynchronous replication can be used to avoid any performance problems with replicating to a distant standby site, while at the same time ensuring that no data is lost should the primary site go down. (Symantec 2006)

ZFS is a new filesystem introduced by Sun in the Solaris 10 Unix Operating System. It offers a number of advanced features compared to existing filesystems such as UFS and VxFS , giving it improved scalability, performance manageability and data integrity. ZFS departs from the traditional model of having a filesystem created on a single disk or a concatenation of disks provided by a volume manager – instead it adapts a pooled storage model where disk devices are aggregated into a storage pool from which they can be allocated to filesystems. Adding disk devices to the storage pool immediately makes them available to all filesystems created from that pool in much the same way as adding memory to a PC allows it to be used by all processes that

need it. This approach results in a storage model that is significantly simpler to configure and administer when compared to other models such as the Veritas volume manager (VxVM) and Veritas filesystem (VxFS). As an example, creating a filesystem striped across 48 physical disks using ZFS takes 2 simple commands and less than 20 seconds, whereas achieving the same task in VxVM/VxFS takes 8 complex commands and an elapsed time of 30 minutes. (Sun Microsystems 2007b) ZFS is a 128-bit filesystem so it offers massive scalability; it also operates using a transactional model and includes data checksums, which protect data from possible corruption. In addition to these impressive features, ZFS also offers the ability to take a point in time copy or snapshot of a filesystem and replicate this copy to a remote system. Incremental snapshots are also supported; meaning once an initial replication of a filesystem has taken place it is possible to keep the remote copy up to date by taking incremental snapshots and sending only the changes since the last snapshot. This approach allows for a remote replica to be kept reasonably up to date with the primary copy in a bandwidth efficient manner. A further advantage of the ZFS filesystem is that it comes bundled as part of the Solaris 10 OS, so no additional license fees are required to deploy it.

3.0 DESIGN OF PROPOSED SOLUTION

3.1 REQUIREMENTS

In section 2.3 the business continuity requirements for a small telecommunications service provider were defined. The key requirements are summarized below:

- Recovery Time Objective (RTO) – The billing system should be back up and running within 4 hours of any outage.
- Recovery Point Objective (RPO) – The billing system should lose no more than 5 minutes worth of data in the event of an outage.
- Cost – cost is an issue, so the business continuity solution should aim to minimize costs while not impacting the availability requirements above.
- HA clustering and Continuous operation are not required.

The following hardware is specified as a sample hardware configuration capable of supporting up to 1 Million subscribers. The business continuity solution needs to be compatible with the hardware configuration below.

Requirement	Value	Comment
Subscribers	1 Million	
Calls/subscriber/day	8	Combination of Voice calls, SMS, data.
CDR History retention	6 months	Rated CDRs are stored for 6 months in the database before being archived off
Billing windows	6 x 8hr	6 x 8hr long bill runs per month
Platform	Sun T2000	
CPUs	8 x 1.2GHz cores	1 x Sun UltraSparc T1 processor
RAM	32GB	
Disk	3 TB	
Operating system	Solaris 10	

Table 5. Billing system hardware requirements

3.2 SOLUTION OVERVIEW

In order to meet the RTO and RPO requirements for the billing system while at the same time minimizing the overall solution costs, the following solution is proposed:

Sun's low cost, energy efficient T2000 servers will be used to host the billing system and its associated Oracle database on both the primary and standby sites. The T2000 server has been benchmarked to support a billing system with over 1 Million subscribers (Intec 2008) and as it requires only 0.25 Oracle licenses per core (Oracle 2008) it is very cost effective to deploy as an Oracle database server – one UltraSparc T1 8 core processor only requires 2 Oracle licenses (8 cores x 0.25).

Disk storage on each site will be provided by an entry-level SAN array such as the Sun StorageTek 2540 array (Sun 2007f). The 2540 disk array is capable of similar performance to

the mid-range 6140 disk array - array - 735MB/s vs. 790MB/s in the SPC-2 benchmark (Sun 2007g) however being an entry level array it is significantly cheaper than the mid-range 6140. A 2540 disk array containing 24 x 146GB disk drives has a list price of \$23,540, whereas the slightly higher performing 6140 disk array with 21 x 146GB disk drives costs \$67,885. (Appendix A) This is a significant difference in price between the entry-level and mid-range arrays for very little difference in performance, which makes the 2450 an attractive option as a cost-effective storage array, however the 2540 disk array does not support any hardware based remote mirroring features, which means that some form of host based replication will be required to replicate data from the primary to the secondary server.

Oracle Data Guard will be used to replicate the database contents from the primary location to the standby location. Data Guard was chosen for a number of reasons:

- It is included as standard with the Oracle Enterprise Edition license, so no additional costs are incurred to deploy it
- As data guard only ships the redo logs to the remote site, it requires significantly less network bandwidth between primary and standby sites. Data Guard requires 1/7th the network bandwidth required by a remote mirroring solution (Oracle 2005) - a factor that is important to a cost-conscious service provider.
- Data Guard operates over a standard TCP/IP network, so it does not need additional storage networking hardware such as fiber channel to IP converters, again helping to minimize costs.

- Data Guard allows the standby database to be used for reporting purposes, adding value to the standby hardware, which might otherwise be considered to be underutilized.
- Data Guard allows the insertion of an artificial delay between the receipt of a redo log and its application to the standby database. This could allow the DR system to be configured to be a certain amount of time behind the primary system, which could allow for the recovery from human errors such as accidentally dropping a table.

Filesystem data will be replicated using the snapshot and send/receive functionality of Sun's ZFS filesystem. The ZFS filesystem was chosen for a number of reasons:

- ZFS is an integral part of the Solaris 10 operating system, so it does not require any additional expenditure to deploy. Support for ZFS is also included as part of a standard Solaris support contract.
- ZFS offers impressive performance when compared to competing filesystems such as Veritas VxFS. In particular, its performance when handling many small files is much faster than VxFS and consumes significantly less CPU resources than VxFS to achieve the same task. (Sun 2007b). This means better performance for the billing system, as the CPU resources can go to processing CDRs rather than to maintaining a filesystem.

- ZFS is significantly easier to administer and configure than Veritas filesystem (VxFS) with Veritas Volume manager (VxVM), which reduces the administrative overhead of looking after a complex Unix system. (Sun 2007b).
- ZFS has built in block integrity checking which prevents silent data corruption

The proposed billing system will utilize two T2000 servers, each attached to a StorageTek 2540 SAN array which will provide the following filesystems:

Filesystem	Usage	Replication
/oracle	Oracle database binaries	Manual ZFS replication when changed
/opt/billing	Billing software binaries	Manual ZFS replication when changed
/billing/bilprd1	Production billing instance	Automatic ZFS replication every 5 minutes
/billing/bilprd1/input	Incoming CDR files for rating	Automatic ZFS replication every 5 minutes
/billing/bilprd1/archive	Archive directory for files that have been rated.	Does not need to be replicated, as files that are in this directory already exist as charges in the database.
/billing/bilprd1/error	Error directory for files that have not rated successfully	Automatic ZFS replication every 5 minutes
/billing/bilprd1/output	Output directory for invoices produced by bill runs.	Does not need to be replicated, as invoices in this directory are also contained in the database.
/u01	Oracle database data files.	Replicated by Oracle Data Guard
/u02		
/u0n		
/archive		

Table 6. Billing system replication

Oracle data Guard will be configured to replicate the Oracle database using ‘maximum performance mode’ – this means that changes on the primary database are replicated asynchronously to the standby database. This method has minimal impact on the performance of

the primary database, but does leave the potential for some data loss in the event of a failure of the primary site before a write was committed on the standby site, however this is acceptable for the billing system case as an RPO of up to 5 minutes is allowed.

ZFS will be configured to take snapshots of the non-database filesystems according to the requirements in Table 6 above. These snapshots will be replicated from the primary system to the standby system.

The proposed solution is illustrated in Figure 7 below.

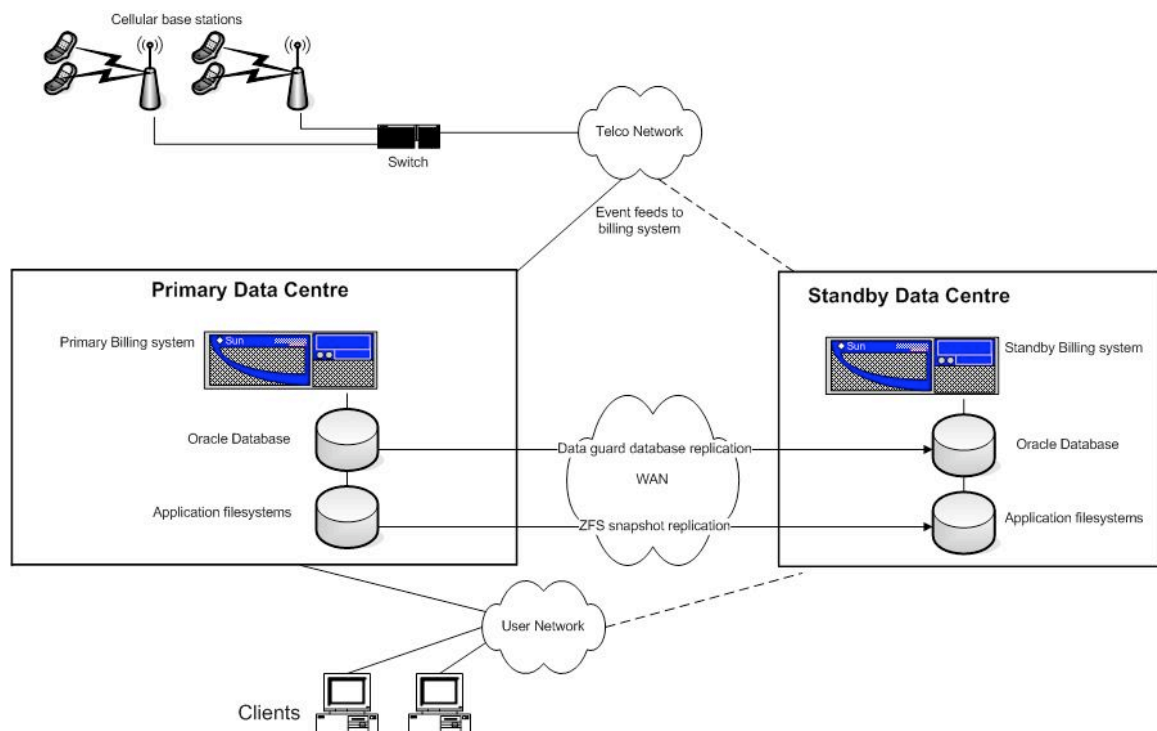


Figure 7. Overview of proposed solution

In the event of the primary site suffering an outage, operations staff will promote the standby database on the secondary site to the primary role and open it for read/write access. An 'ACTIVE' flag will be set to 1 on the standby billing system to prevent it from receiving any more snapshots from the primary server should it come back into operation. The billing system can then be started from the replicated filesystems and resume normal operations.

Upstream and Downstream systems interfacing to the billing system will need to be redirected to the standby billing system once it becomes active. This can be accomplished by altering the DNS entry for the billing system to point to the IP address of the standby system or by pre-configuring primary and standby IP addresses for the billing system into the interfacing systems.

Once the outage affecting the primary site is resolved, normal operations can be resumed on the primary site by configuring the standby site to replicate back to the old primary site. Once both sites are back in sync, a failover can be initiated back to the primary site.

4.0 IMPLEMENT AND TEST PROOF OF CONCEPT

This section contains details the implementation of a proof of concept system, and the tests carried out on that system to assess its suitability for use as a disaster recovery solution for a small billing system.

The proof of concept test involved the following main steps:

- Install and configure the base Solaris 10 operating system on two servers
- Configure a ZFS pool on both servers.
- Install Oracle 10g on primary server and replicate the Oracle installation to the secondary server
- Create an Oracle 10g database on the primary server and configure DataGuard to replicate this database to the standby server.
- Insert a large number of rows into the test database and verify that they are replicated across to the standby server
- Create a billing input filesystem structure on the primary server and configure a script to replicate this structure at 5-minute intervals to the secondary server.
- Create a test script to simulate incoming CDRs being placed in the input filesystem and moved to the archive filesystem. Verify that these incoming CDR files are replicated correctly to the standby system.

- Observe the snapshot replication time and network bandwidth utilization at varying rates of incoming EDR files.
- Perform a failover from the primary system to the standby system.

The results of this proof of concept test indicate that the proposed DR system is capable of providing DR capability from a functional point of view however further testing at full production volumes will be required to verify functionality at full production volumes.

4.1 SYSTEM HARDWARE CONFIGURATION

The proof of concept system was implemented on a pair of relatively old Sun systems – a Sun E420 with 4 x 450MHz UltraSparcII processors and 4GB RAM, and a Sun Ultra5 with 1 x 333MHz UltraSparcII processor and 1GB RAM. These systems were deemed suitable for the proof of concept, as despite their lack of processing power they were still capable of running the current versions of the 64-bit Solaris operating system (Solaris 10) and the Oracle 10g database. For the purposes of the proof of concept it was not considered necessary to handle full production volumes. A full install of the Solaris 10 Update 4 operating system was performed on both servers.

The E420 server was configured as the primary server (node1). 2 x 9GB disks were configured as a ZFS pool called **zfs1**.

The Solaris 'format' command was used to identify the device names of available disks on the system:

```
format> disk
```

```
AVAILABLE DISK SELECTIONS:
```

0. c0t0d0 <ST320414A cyl 38790 alt 2 hd 16 sec 63>
/pci@1f,0/pci@1,1/ide@3/dad@0,0
1. clt10d0 <SEAGATE-ST39175LC-HP05 cyl 11697 alt 2 hd 5 sec 304>
/pci@1f,0/pci@1/scsi@3/sd@a,0
2. clt11d0 <SEAGATE-ST39173WC-HP11 cyl 7499 alt 2 hd 10 sec 237>
/pci@1f,0/pci@1/scsi@3/sd@b,0
3. clt13d0 <SEAGATE-ST39173WC-HP11 cyl 7499 alt 2 hd 10 sec 237>
/pci@1f,0/pci@1/scsi@3/sd@d,0
4. clt14d0 <SEAGATE-ST39173WC-HP11 cyl 7499 alt 2 hd 10 sec 237>
/pci@1f,0/pci@1/scsi@3/sd@e,0
5. clt15d0 <SEAGATE-ST39175LC-HP05 cyl 11697 alt 2 hd 5 sec 304>
/pci@1f,0/pci@1/scsi@3/sd@f,0

```
Specify disk (enter its number) [4]:
selecting clt14d0
[disk formatted]
format> q
```

Disks clt13d0 and clt14d0 were verified to be not currently in use and so were selected for use in the ZFS pool.

The ‘zpool create’ command was used to create a ZFS pool called zfs1 using the two disks identified above. For the purposes of this test, no redundancy was configured into the ZFS pool, however in a production system it would be necessary to provide RAID protection either at the hardware level by using a RAID array or in software by using ZFS’s built in RAIDZ feature.

```
bash-3.00# zpool create zfs1 /dev/dsk/clt13d0 /dev/dsk/clt14d0
```

The ‘zpool status’ command was then used to verify that the pool was created correctly.

```
bash-3.00# zpool status
pool: zfs1
state: ONLINE
scrub: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
zfs1	ONLINE	0	0	0
clt13d0	ONLINE	0	0	0
clt14d0	ONLINE	0	0	0

```
errors: No known data errors
```

The 'zfs list' command confirms that there is a 16.6GB zfs filesystem mounted as /zfs1.

```
bash-3.00# zfs list
```

```
NAME      USED    AVAIL    REFER  MOUNTPOINT
zfs1      87K    16.6G    24.5K   /zfs1
```

The following filesystems were created on the zfs1 pool on node 1

Filesystem	Mountpoint	Comment
zfs1/archive1	/archive1	Oracle Archivelog filesystem
zfs1/oracle	/oracle	Oracle binaries
zfs1/oradata	/oradata	Oracle datafiles
zfs1/input	zfs1/input	Input files

Table 7. ZFS filesystems

The following commands were used to create each filesystem:

Create the ZFS filesystem

```
# zfs create zfs1/archive1
```

Set the mountpoint of the filesystem

```
# zfs set mountpoint=/archive1 zfs1/archive1
```

Verify filesystem is mounted correctly

```
# df -k /archive1
Filesystem      kbytes    used    avail capacity  Mounted on
zfs1/archive1   52383744     24 36657346     1%   /archive1
```

Change ownership of the filesystem

```
# chown oracle:dba /archive1
```

These steps were repeated for each filesystem in table 7 above.

The Ultra5 system was configured as the standby server(node2). 2 x 9GB disks were configured as a ZFS pool called zfs1 using the same procedure as for node1. Only the /archive1 filesystem was created initially, the remaining filesystems were replicated across from node1.

Both systems were networked together using a 100BaseT fast Ethernet network.

4.2 SYSTEM SOFTWARE CONFIGURATION

4.2.1 Oracle 10g installation

Secure Shell (ssh) public key authentication was configured on both servers to allow the root user on each server to execute commands on the other server without needing to supply a password. This was configured in accordance with the Oracle clusterware installation guide (Oracle 2007b)

Oracle 10g 10.2.0.3 Enterprise Edition was installed on the primary server under the /oracle filesystem following the installation instructions contained in the Oracle 10g installation guide for the 64 bit Solaris Sparc platform (Oracle 2005b). The /oracle filesystem was created as a ZFS filesystem zfs1/oracle, so the ZFS snapshot and send features were used to replicate the Oracle installation across to the standby server. This was accomplished by taking a snapshot of the /oracle filesystem then piping the output of the 'zfs send' command on node1 via ssh to the 'zfs receive' command on node2. This resulted in the zfs1/oracle filesystem being replicated to node2. The following commands were used:

On node 1 create a snapshot of zfs1/oracle called snap0:

```
zfs snapshot zfs1/oracle@snap0
```

On node1 send that snapshot to node2 by piping it through ssh:

```
zfs send zfs1/oracle@snap0 | ssh -C node2 zfs receive zfs1/oracle
```

On node2, set the mountpoint for zfs1/oracle to be /oracle:

```
zfs set mountpoint=/oracle zfs1/oracle
```

On node2 verify that the snapshot sent has been received and mounted:

```
bash-3.00# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
zfs1	8.48G	8.13G	24.5K	/zfs1
zfs1/oracle	4.65G	8.13G	4.65G	/oracle
zfs1/oracle@snap0	82K	-	4.65G	-

```
df -k /oracle
```

Filesystem	kbytes	used	avail	capacity	Mounted on
zfs1/oracle	17418240	4875998	7495757	40%	/oracle

Once the oracle filesystem was replicated across to node2 it was necessary to create the Oracle user and the dba group as per the Oracle 10g installation guide, then the Oracle root.sh script was executed to perform the root specific parts of the Oracle installation.

```
bash-3.00# /oracle/app/oracle/product/10.2.0.3/root.sh
```

```
Running Oracle10 root.sh script...
```

```
The following environment variables are set as:
```

```
ORACLE_OWNER= oracle
```

```
ORACLE_HOME= /oracle/app/oracle/product/10.2.0.3
```

```
Enter the full pathname of the local bin directory: [/usr/local/bin]:
```

```
Copying dbhome to /usr/local/bin ...
```

```
Copying oraenv to /usr/local/bin ...
```

```
Copying coraenv to /usr/local/bin ...
```

```
Creating /var/opt/oracle/oratab file...
```

```
Entries will be added to the /var/opt/oracle/oratab file as needed by
```

```
Database Configuration Assistant when a database is created
```

```
Finished running generic part of root.sh script.
```

```
Now product-specific root actions will be performed.
```

At this point we had an identical Oracle 10g installation on both node1 and node2

4.2.2 Oracle Primary Database creation

Using the Oracle dbca utility, an OLTP database was created on the node1 server using the transaction-processing template. This database was given the database name PRIMARY.

A user called 'billing' was created to contain the schema to be used to contain the billing system data. A table called CDR was created to simulate the loading of call detail records into the billing

system database. This table was created as per table 7 below to contain sample CDR records using a Cisco Switch format (Cisco 1998):

Time	Qualifier	Calling Number	Called Number	PRI number	ID	B Channel Number	Time	Text
DATE	VARCHAR2(11)	NUMBER(21)	NUMBER(21)	NUMBER(5)		NUMBER(3)	DATE	VARCHAR2(35)
Time of event	Type of event – <ul style="list-style-type: none"> • Call rqst • Call disc • Setup Fail • DiscFail 	Calling party phone number (APARTY)	Called party phone number (BPARTY)			Bearer channel number	Disconnect time	Disconnect cause code.

Table 8. CDR format

The table was created using the following SQL:

```
CREATE TABLE CDR (
  ID                NUMBER(20) NOT NULL,
  STIME             DATE,
  Qualifier         VARCHAR2(11),
  APARTY            NUMBER(21) NOT NULL,
  BPARTY            NUMBER(21) NOT NULL,
  PRIID             NUMBER(5) NOT NULL,
  BCH_NUM           NUMBER(3) NOT NULL,
  ETIME             DATE,
  TEXT              VARCHAR2(35),
  CONSTRAINT CDR_PRIMARY_KEY PRIMARY KEY (ID)
);
```

A sequence called CDR_SEQUENCE was created to generate a unique ID number for each row in the CDR table. This ID was used as the table's primary key.

```
CREATE SEQUENCE CDR_SEQUENCE START 1 INCREMENT 1;
```

4.2.3 Oracle Standby Database creation

Once the primary database was created, a physical standby database was implemented on the standby server using DataGuard by following the instructions contained in section 3 of the Oracle 10g Data Guard Concepts and Administration manual (Oracle 2006a). The standby database was given the database name of STANDBY. Detailed steps for the primary and standby database configuration are contained in Appendix B

4.2.4 Generate rows in database.

The following perl script was developed to insert a large number of unique rows into the CDR table to simulate the database activity generated by rating events into the billing system. It uses the perl DBI and DBDOracle modules to connect to the PRIMARY database and insert rows into the CDR table. An Oracle sequence called CDR_SEQUENCE is used to generate a unique ID number for each row inserted into the table. This ID number is used as the table's primary key. Each time the script is executed it adds 5000 rows to the CDR table.


```
#!/usr/bin/perl

use strict;
use DBI;

#
# Prepare connection to PRIMARY database
#
my $dbh = DBI->connect('dbi:Oracle:host=localhost;sid=PRIMARY;port=1521',
    'billing', 'billing', { RaiseError => 1, AutoCommit => 0 });

#
# Prepare SQL insert statement
#

my $sth = $dbh->prepare("insert into CDR
(ID, TIME, QUALIFIER, APARTY, BPARTY, PRIID, BCH_NUM, ETIME, TEXT)
values (CDR_SEQUENCE.NEXTVAL, TO_DATE(?, 'yyyymmddHH24MISS'), ?, ?, ?, ?, TO_DATE(?,
'yyyymmddHH24MISS'), ?)")
or die "Can't prepare statement: $DBI::errstr";

#
# Loop 5000 times, generating and inserting a unique row for
# every iteration of the loop
#

for (my $i=0 ; $i <5000; $i++) {

    my ($sec,$min,$hour,$mday,$mon,$year,$yday,$isdst)=localtime(time);

    $year = $year + 1900 ;
    $mon = sprintf("%0.2d" , $mon + 1) ;
    $mday = sprintf("%0.2d" , $mday) ;
    $hour = sprintf("%0.2d" , $hour) ;
    $min = sprintf("%0.2d" , $min) ;
    $sec = sprintf("%0.2d" , $sec) ;

    my $starttime = $year.$mon.$mday.$hour.$min.$sec;
    my $endtime = $year.$mon.$mday.$hour.$min.$sec;
    my $anum = 3333333333 + $i ;
    my $bnum = 1234567890 + $i ;

    $sth->bind_param(1, $starttime) ;
    $sth->bind_param(2, 'Call Rqst');
    $sth->bind_param(3, $anum);
    $sth->bind_param(4, $bnum);
    $sth->bind_param(5, 25);
    $sth->bind_param(6, 2);
    $sth->bind_param(7, $endtime) ;
    $sth->bind_param(8, 'xxxxx xxx xxx');

    #
    # Execute insert statement
    #
    $sth->execute() or die "Can't execute statement: $DBI::errstr";

}

#
# Commit rows and close database connection
#
$dbh->commit or die $dbh->errstr;
$dbh->disconnect or warn $dbh->errstr;
```

To verify that data was being replicated from the primary to the standby database, the genrows.pl script was used to populate the database with 35,000 rows.

Verify CDR table is empty:

```
root.oracle.nodel>sqlplus billing

SQL*Plus: Release 10.2.0.3.0 - Production on Sat Jul 26 17:01:40 2008
Copyright (c) 1982, 2006, Oracle. All Rights Reserved.
Enter password:

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bit Production
With the Partitioning, OLAP and Data Mining options


SQL> select count(*) from CDR ;
      COUNT(*)
-----
           0
```

Run genrows.pl to add 5000 rows to the table:

```
root.oracle.nodel>./genrows.pl
```

Verify CDR table contains rows:

```
root.oracle.nodel>sqlplus billing

SQL*Plus: Release 10.2.0.3.0 - Production on Sat Jul 26 17:01:40 2008
Copyright (c) 1982, 2006, Oracle. All Rights Reserved.
Enter password:

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bit Production
With the Partitioning, OLAP and Data Mining options


SQL> select count(*) from CDR ;
      COUNT(*)
-----
        5000

SQL>
```

Run genrows.pl some more times to add more rows to the table:

```
root.oracle.nodel>./genrows.pl
root.oracle.nodel>./genrows.pl
root.oracle.nodel>./genrows.pl
root.oracle.nodel>./genrows.pl
root.oracle.nodel>./genrows.pl
root.oracle.nodel>./genrows.pl
```

Verify CDR table contains rows:

```
root.oracle.nodel>sqlplus billing
```

```
SQL*Plus: Release 10.2.0.3.0 - Production on Sat Jul 26 17:01:40 2008  
Copyright (c) 1982, 2006, Oracle. All Rights Reserved.  
Enter password:
```

```
Connected to:
```

```
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bit Production  
With the Partitioning, OLAP and Data Mining options
```

```
SQL> select count(*) from CDR ;
```

```
      COUNT(*)  
-----  
          35000
```

```
SQL> select max(ID) from CDR ;
```

```
      MAX(ID)  
-----  
          73419
```

This confirmed that there were 35000 rows in the PRIMARY database and the highest unique ID number was 73419.

4.2.5 Verify data replication to standby database.

In order to verify that the rows generated above were being replicated across to the standby database, a switchover was performed to make the STANDBY database active and the PRIMARY database a standby. The STANDBY database was then queried to ensure it contained the same number of rows as the PRIMARY and that the highest CDR table ID number matched that of the PRIMARY, thus confirming that all data had been successfully replicated across.

On node 1 the following steps were performed to prepare the database for switchover from the primary database role to the standby database role:

Verify the switchover status of the database:

```
SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;

SWITCHOVER_STATUS
-----
SESSIONS ACTIVE
```

Identify any active sessions:

```
SQL> SELECT SID, PROCESS, PROGRAM FROM V$SESSION WHERE TYPE = 'USER'
2 AND SID <> (SELECT DISTINCT SID FROM V$MYSTAT);
```

SID	PROCESS	PROGRAM
100	5250	oracle@node1 (J000)

This indicated that the only active session was my current session - this could be terminated, so the following command was issued to turn the PRIMARY database to a physical standby and to close any active sessions:

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PHYSICAL STANDBY WITH SESSION SHUTDOWN;

Database altered.
```

The database was then shutdown and restarted in mount mode:

```
SQL> STARTUP MOUNT ;
ORACLE instance started.
Total System Global Area 167772160 bytes
Fixed Size                2028624 bytes
Variable Size             134220720 bytes
Database Buffers          25165824 bytes
Redo Buffers              6356992 bytes
Database mounted.
SQL>
```

Querying V\$DATABASE indicated that the database was now a physical standby.

```
SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
SWITCHOVER_STATUS
-----
TO PRIMARY
```

On node 2 the following steps were performed to prepare the database for switchover from the standby database role to the primary database role:

Verify the switchover status of the database:

```
SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
SWITCHOVER_STATUS
-----
TO PRIMARY
```

Initiate switchover to primary role:

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;
Database altered.
```

Verify the switchover status of the database:

```
SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE ;
SWITCHOVER_STATUS
-----
TO STANDBY
```

Open the database, then shutdown and restart the database:

```
SQL> ALTER DATABASE OPEN;
Database altered.
```

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

```
SQL> SQL> startup

ORACLE instance started.

Total System Global Area 163577856 bytes
Fixed Size                2028624 bytes
Variable Size             79694768 bytes
Database Buffers          79691776 bytes
Redo Buffers              2162688 bytes
Database mounted.
Database opened.
SQL>
```

At this point, Node 2 is now the active database.

Check to see if all the data from the PRIMARY is present by connecting as the billing user and comparing the number of rows and the maximum CDR ID value with those from the PRIMARY instance :

```
SQL> connect billing/billing
Connected.

SQL> select count(*) from CDR ;
COUNT(*)
-----
35000

SQL> select max(ID) from CDR ;
MAX(ID)
-----
73419

SQL>
```

This confirmed that all data had been replicated successfully from the primary to the standby database.

On node 1 the following command was issued to allow the PRIMARY instance in the standby role to receive redo log updates from the STANDBY instance:

```
SQL> alter database recover managed standby database disconnect from session ;
```

The steps in this section were then repeated with the nodes switched to restore the PRIMARY instance on node1 back to the primary role and the STANDBY instance on node2 back to the standby role.

4.2.6 Generate CDR files in input filesystem

The following perl script was used to generate files containing Call Detail Records (CDRs) using the same format as was used to populate the database in section 4.2.4 above.

genfiles.pl listing

```
#!/usr/bin/perl

for ($filename = 1; $filename <= 27; $filename++) {
    my ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time);
    $timestamp=($year+1900).($mon+1).$mday.$hour.$min.$sec ;

    $FILENAME="cdrfile_" . $filename . "_" . $timestamp . ".evt" ;

    open(CDRFILE, ">$FILENAME") ;
    for ($i = 1; $i <= 5000; $i++) {
        ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst)=localtime(time);
        printf CDRFILE "%4d-%02d-%02d %02d:%02d:%02d", $year+1900, $mon+1, $mday, $hour, $min, $sec;
        print CDRFILE "| " ;
        print CDRFILE "Call Rqst" ;
        print CDRFILE "| " ;
        print CDRFILE "3537865432"+$i ;
        print CDRFILE "| " ;
        print CDRFILE "353786549999"+$i ;
        print CDRFILE "| " ;
        print CDRFILE "25" ;
        print CDRFILE "| " ;
        print CDRFILE "2" ;
        print CDRFILE "| " ;
        printf CDRFILE "%4d-%02d-%02d %02d:%02d:%02d", $year+1900, $mon+1, $mday, $hour, $min+3, $sec+22;
        print CDRFILE "| " ;
        print CDRFILE "xxxxxx xxx xxx\n" ;
    }
    close (CDRFILE) ;
}
```

Each time it is run, the script generates 27 files, each containing 5000 CDR rows – a total of 135,000 CDRs. This approximates 5 minutes worth of call data during the busy hour based on an assumption of 1 Million subscribers each generating an average of 8 calls per day, with 20% of that usage occurring during the busy hour.

Running the genfiles.pl script once on node1 took 19 seconds and generated 13.8MB of data in the zfs1/input filesystem.

```
bash-3.00# time ./genfiles.pl
```

```
real    0m19.811s
user    0m19.523s
sys     0m0.259s
```

```
bash-3.00# du -sk
13894   .
```

```
bash-3.00# ls
cdrfile_10_2008730204042.evt    cdrfile_17_2008730204047.evt    cdrfile_23_2008730204051.evt
cdrfile_4_2008730204037.evt    cdrfile_18_2008730204048.evt    cdrfile_24_2008730204052.evt
cdrfile_11_2008730204042.evt   cdrfile_19_2008730204048.evt    cdrfile_25_2008730204053.evt
cdrfile_5_2008730204038.evt    cdrfile_1_2008730204035.evt     cdrfile_26_2008730204053.evt
cdrfile_12_2008730204043.evt   cdrfile_20_2008730204049.evt    cdrfile_27_2008730204054.evt
cdrfile_6_2008730204039.evt    cdrfile_21_2008730204050.evt    cdrfile_2_2008730204036.evt
cdrfile_13_2008730204044.evt   cdrfile_22_2008730204050.evt    cdrfile_3_2008730204037.evt
cdrfile_7_2008730204039.evt
cdrfile_14_2008730204045.evt
cdrfile_8_2008730204040.evt
cdrfile_15_2008730204045.evt
cdrfile_9_2008730204041.evt
cdrfile_16_2008730204046.evt
genfiles.pl
```

With 5 minutes worth of CDR records in the zfs1/input filesystem on node1, a snapshot of the filesystem was taken and replicated across to node2.

```
bash-3.00# zfs snapshot zfs1/input@initial_snap
```

```
bash-3.00# time zfs send zfs1/input@initial_snap | ssh node2 zfs receive zfs1/input
```

```
real    0m8.313s
user    0m2.512s
sys     0m0.552s
```

The 13.8MB of data was replicated across in 8.31 seconds.

On node2 the 'zfs list' and 'df -k' commands were used to verify that the filesystem had been replicated across correctly and that it contained the same size and number of files as on node1.

```
bash-3.00# df -k zfs1/input
Filesystem            kbytes    used    avail capacity  Mounted on
zfs1/input            17418240  13911  7481437      1%    /zfs1/input

bash-3.00# zfs list
NAME                                USED    AVAIL    REFER  MOUNTPOINT
zfs1                                9.48G   7.13G   26.5K   /zfs1
zfs1/archive1                      716M    7.13G   716M    /archive1
zfs1/input                        13.6M    7.13G   13.6M    /zfs1/input
zfs1/input@initial_snap           24.5K    -       13.6M    -
zfs1/oracle                       4.65G    7.13G   4.65G    /oracle
zfs1/oracle@snap0                 913K    -       4.65G    -
zfs1/oradata                     4.11G    7.13G   1.14G    /oradata
zfs1/oradata@snap0               2.97G    -       3.83G    -

bash-3.00# cd /zfs1/input
bash-3.00# ls
cdrfile_10_2008730204042.evt  cdrfile_1_2008730204035.evt  cdrfile_3_2008730204037.evt
cdrfile_11_2008730204042.evt  cdrfile_20_2008730204049.evt  cdrfile_4_2008730204037.evt
cdrfile_12_2008730204043.evt  cdrfile_21_2008730204050.evt  cdrfile_5_2008730204038.evt
cdrfile_13_2008730204044.evt  cdrfile_22_2008730204050.evt  cdrfile_6_2008730204039.evt
cdrfile_14_2008730204045.evt  cdrfile_23_2008730204051.evt  cdrfile_7_2008730204039.evt
cdrfile_15_2008730204045.evt  cdrfile_24_2008730204052.evt  cdrfile_8_2008730204040.evt
cdrfile_16_2008730204046.evt  cdrfile_25_2008730204053.evt  cdrfile_9_2008730204041.evt
cdrfile_17_2008730204047.evt  cdrfile_26_2008730204053.evt  genfiles.pl
cdrfile_18_2008730204048.evt  cdrfile_27_2008730204054.evt
cdrfile_19_2008730204048.evt  cdrfile_2_2008730204036.evt

bash-3.00# du -sk
13894  .
```

Next the 'sum' command was used on the CDR files on both nodes to generate a checksum for the 27 CDR files. This checksum was used to conform the integrity of the data replicated across.

On node1:

```
bash-3.00# sum *
25933 909 cdrfile_10_2008730204042.evt
34593 909 cdrfile_11_2008730204042.evt
41103 909 cdrfile_12_2008730204043.evt
47367 909 cdrfile_13_2008730204044.evt
55933 909 cdrfile_14_2008730204045.evt
63627 909 cdrfile_15_2008730204045.evt
4358 909 cdrfile_16_2008730204046.evt
9670 909 cdrfile_17_2008730204047.evt
40933 909 cdrfile_18_2008730204048.evt
47475 909 cdrfile_19_2008730204048.evt
50431 909 cdrfile_1_2008730204035.evt
41021 909 cdrfile_20_2008730204049.evt
15933 909 cdrfile_21_2008730204050.evt
25115 909 cdrfile_22_2008730204050.evt
31507 909 cdrfile_23_2008730204051.evt
37887 909 cdrfile_24_2008730204052.evt
```

```
45933 909 cdrfile_25_2008730204053.evt
54333 909 cdrfile_26_2008730204053.evt
60713 909 cdrfile_27_2008730204054.evt
56777 909 cdrfile_2_2008730204036.evt
398 909 cdrfile_3_2008730204037.evt
40824 909 cdrfile_4_2008730204037.evt
34369 909 cdrfile_5_2008730204038.evt
40933 909 cdrfile_6_2008730204039.evt
6983 909 cdrfile_7_2008730204039.evt
11897 909 cdrfile_8_2008730204040.evt
18227 909 cdrfile_9_2008730204041.evt
1525 2 genfiles.pl
```

On node2:

```
bash-3.00# sum *
25933 909 cdrfile_10_2008730204042.evt
34593 909 cdrfile_11_2008730204042.evt
41103 909 cdrfile_12_2008730204043.evt
47367 909 cdrfile_13_2008730204044.evt
55933 909 cdrfile_14_2008730204045.evt
63627 909 cdrfile_15_2008730204045.evt
4358 909 cdrfile_16_2008730204046.evt
9670 909 cdrfile_17_2008730204047.evt
40933 909 cdrfile_18_2008730204048.evt
47475 909 cdrfile_19_2008730204048.evt
50431 909 cdrfile_1_2008730204035.evt
41021 909 cdrfile_20_2008730204049.evt
15933 909 cdrfile_21_2008730204050.evt
25115 909 cdrfile_22_2008730204050.evt
31507 909 cdrfile_23_2008730204051.evt
37887 909 cdrfile_24_2008730204052.evt
45933 909 cdrfile_25_2008730204053.evt
54333 909 cdrfile_26_2008730204053.evt
60713 909 cdrfile_27_2008730204054.evt
56777 909 cdrfile_2_2008730204036.evt
398 909 cdrfile_3_2008730204037.evt
40824 909 cdrfile_4_2008730204037.evt
34369 909 cdrfile_5_2008730204038.evt
40933 909 cdrfile_6_2008730204039.evt
6983 909 cdrfile_7_2008730204039.evt
11897 909 cdrfile_8_2008730204040.evt
18227 909 cdrfile_9_2008730204041.evt
1525 2 genfiles.pl
```

These tests confirmed that the ZFS replication had successfully replicated the input CDR data from the primary to the standby node without any data loss or corruption.

4.2.7 Configure and test script for automatic zfs replication.

The following shell script – `zfs_replicate.sh`, was developed to automatically replicate the contents of a ZFS filesystem from the primary to the standby system.

It is configured to run every 5 minutes under the control of the Unix cron scheduler. Each time it runs, it performs the following tasks:

- Check to see if a lockfile exists on the local machine, if it does then exit. Else-
- Create the lock file to signify script is running.
- Generate a list of the snapshots for the input filesystem on the local node
- If snapshot `snap2` exists on local or remote nodes then delete it.
- If snapshot `snap1` exists on local or remote nodes then rename it to `snap2`
- If snapshot `snap0` exists on local or remote nodes then rename it to `snap1`
- On the local node, take a snapshot of the input filesystem, naming it `snap0`
- If this is not the first snapshot, send an incremental snapshot containing the differences between `snap0` and `snap1` to remote system using `zfs send`
- If it is the first snapshot, send an initial snapshot `snap0` to remote system.
- Remove lockfile before exiting.

zfs_replicate.sh listing

```
#!/usr/bin/ksh

# set filesystem to replicate
FILESYSTEM="zfs1/input"
# set destination node
destination_node=node2
LOCKFILE="/var/tmp/zfs_replicate.lck"
DATE=`date`

echo zfs_replicate start at $DATE

if [[ -f $LOCKFILE ]]
then
    echo "zfs_replicate in progress. exiting ...."
else
    # create lock file to indicate replicate in progress
    touch $LOCKFILE

    index=0

    # obtain list of local snapshots
    for snapshot in `zfs list -r -H -t snapshot -s name $FILESYSTEM | awk '{print $1}'`
    do
        SNAPSHOTS[$index]=$snapshot
        (( index=index+1 ))
    done

    # delete oldest snapshot on local and remote nodes
    if [[ -n ${SNAPSHOTS[2]} ]]
    then
        echo "Deleting local Snapshot snap2: " ${SNAPSHOTS[2]}
        zfs destroy ${SNAPSHOTS[2]}
        rsnap2=`ssh $destination_node zfs list -H -t snapshot -s name ${SNAPSHOTS[2]}
2>/dev/null | awk '{print $1}'`
    fi

    if [[ -n ${rsnap2} ]]
    then
        echo "Deleting remote snapshot " $rsnap2 on " $destination_node
        ssh $destination_node zfs destroy $rsnap2
    fi

    # rename second oldest snapshot on local and remote nodes
    if [[ -n ${SNAPSHOTS[1]} ]]
    then
        echo "renaming local snapshot snap1 to snap2"
        zfs rename ${SNAPSHOTS[1]} $FILESYSTEM@snap2
        rsnap1=`ssh $destination_node zfs list -H ${SNAPSHOTS[1]} 2>/dev/null | awk '{print
$1}'`
    fi

    if [[ -n ${rsnap1} ]]
    then
        echo "renaming remote snapshot snap1 on $destination_node to snap2"
        ssh $destination_node zfs rename $rsnap1 $FILESYSTEM@snap2
    fi

    # rename most recent snapshot on local and remote nodes, take new snapshot
    if [[ -n ${SNAPSHOTS[0]} ]]
    then
        echo "renaming local snapshot snap0 to snap1"
        zfs rename ${SNAPSHOTS[0]} $FILESYSTEM@snap1

        echo "Taking new snapshot snap0"
        zfs snapshot $FILESYSTEM@snap0
    fi

```

```

    rsnap0=`ssh $destination_node zfs list -H ${SNAPSHOTS[0]} 2>/dev/null | awk '{print $1}'`

    else
        echo "Taking initial snapshot snap0"
        zfs snapshot $FILESYSTEM@snap0
    fi

    if [[ -n ${rsnap0} ]]
    then
        echo "renaming snap0 on $destination_node to snap1"
        ssh $destination_node zfs rename $rsnap0 $FILESYSTEM@snap1
        echo "sending incremental snapshot to node " $destination_node
        zfs send -i $FILESYSTEM@snap1 $FILESYSTEM@snap0 | ssh -C $destination_node zfs
receive -v -F $FILESYSTEM
    else
        echo "sending initial snapshot to node " $destination_node
        zfs send $FILESYSTEM@snap0 | ssh -C $destination_node zfs receive -v -F $FILESYSTEM

    fi
    # remove lock file to indicate replicate is finished
    rm $LOCKFILE
    echo zfs_replicate end at `date`
    echo "#####"
fi

```

This script was configured to run every 5 minutes by adding the following crontab entry to root's crontab on node 1:

```

0,5,10,15,20,25,30,35,40,45,50,55 * * * * /zfs1/test/zfs_replicate.sh >>
/tmp/zfs_replicate.log 2>&1

```

The script's standard error and standard output were sent to a logfile `/tmp/zfs_replicate.log` for monitoring purposes.

When the script was running at 5-minute intervals, the `genfiles.pl` script from section 4.2.6 was run a number of times to simulate incoming files, and the output of the monitor script was observed. Each time `genfiles.pl` was executed, 135,000 CDR events were added to the input filesystem. Simulating 5 minutes worth of call usage.

The automatic replication script picked up these usage files and replicated across to the standby node. Due to the incremental snapshots, only data that had changed between executions of the `zfs_replicate.sh` script was sent.

Log extract after running `genfiles.pl` twice to simulate 10 minutes of usage:

```
zfs_replicate start at Sat Aug 2 18:05:00 IST 2008
Deleting local Snapshot snap2: zfs1/input@snap2
Deleting remote snapshot zfs1/input@snap2 on node2
renaming local snapshot snap1 to snap2
renaming remote snapshot snap1 on node2 to snap2
renaming local snapshot snap0 to snap1
Taking new snapshot snap0
renaming snap0 on node2 to snap1
sending incremental snapshot to node node2
receiving incremental stream of zfs1/input@snap0 into zfs1/input@snap0
received 27.1Mb stream in 10 seconds (2.71Mb/sec)
zfs_replicate end at Sat Aug 2 18:05:29 IST 2008
#####
```

Here, 10 minutes of usage – 27.1 MB was transferred across in 10 seconds. In the next execution of `zfs_replicate.sh` there were no new files, so no data was sent across.

```
zfs_replicate start at Sat Aug 2 18:10:00 IST 2008
Deleting local Snapshot snap2: zfs1/input@snap2
Deleting remote snapshot zfs1/input@snap2 on node2
renaming local snapshot snap1 to snap2
renaming remote snapshot snap1 on node2 to snap2
renaming local snapshot snap0 to snap1
Taking new snapshot snap0
renaming snap0 on node2 to snap1
sending incremental snapshot to node node2
receiving incremental stream of zfs1/input@snap0 into zfs1/input@snap0
received 695b stream in 1 seconds (695b/sec)
zfs_replicate end at Sat Aug 2 18:10:19 IST 2008
#####
```

To test whether the replication mechanism was capable of coping with spikes in input volumes or a backlog of files to be replicated, genfiles.pl was run 5 times to generate 5 times the normal peak CDR volume. This produced 67.8MB of data, which was replicated across in 22 seconds.

```
zfs_replicate start at Sat Aug 2 20:00:00 IST 2008
Deleting local Snapshot snap2: zfs1/input@snap2
Deleting remote snapshot zfs1/input@snap2 on node2
renaming local snapshot snap1 to snap2
renaming remote snapshot snap1 on node2 to snap2
renaming local snapshot snap0 to snap1
Taking new snapshot snap0
renaming snap0 on node2 to snap1
sending incremental snapshot to node node2
receiving incremental stream of zfs1/input@snap0 into zfs1/input@snap0
received 67.8Mb stream in 22 seconds (3.08Mb/sec)
zfs_replicate end at Sat Aug 2 20:00:41 IST 2008
#####
```

These tests indicate that the automatic replication mechanism is capable of automatically replicating the contents of the /zfs1/input filesystem from the primary to the standby node.

4.2.8 Carry out failover test.

A test was carried out to confirm that the proposed DR solution could protect against a simulated failure of the primary node. This involved simulating normal usage of the billing system then simulating a failure of the primary node and a recovery to the second node. Measurements were taken of the amount of data lost in the failover process and the amount of time taken to make the database available on the secondary node.

A shell script was configured to simulate an operational load with CDR files being added and removed from the zfs1/input directory and rows being added to the database. The test script performs the following tasks:

- Look for CDR files older than 10 minutes and removes them.
- Generates 10 new CDR files each containing 500 CDR records.
- Adds 5000 rows to the CDR table in the Oracle database.
- Waits for 5 minutes and then repeats the steps above.

This simulates event files being added to the input directory and then to the database at a rate of 5000 records in 5 minutes. Removing files older than 10 minutes simulates the CDR files being removed to the archive directory after rating.

test1.sh listing

```
#!/usr/bin/ksh
cd /zfs1/input
while true
do
# remove files older then 10 mins
/usr/local/bin/find /zfs1/input -name "*.evt" -mmin +10 -exec rm {} \;

# generate 10 CDR files with 5000 records each
/zfs1/test/genfiles.pl

# add 50000 rows to the database
/zfs1/test/oracle/genrows.pl

# wait 5 minutes
sleep 300
done
```

This shell script was run for a period of 20 minutes to simulate normal operation of the billing system.

After 20 minutes a failover test was carried out to simulate loss of the PRIMARY system and to verify that both filesystem and database data was replicated correctly.

First check to see the timestamp of the most recent row in the primary database

```
1* select to_char(max(stime),'hh24:mi:ss yyyy-mm-dd') as last_rowtime from cdr
SQL> /

LAST_ROWTIME
-----
16:43:22 2008-08-06
```

Next kill the test1.sh and zfs_replicate.sh scripts and abort the PRIMARY Oracle instance to simulate failure of node1.

```
SQL> connect / as sysdba
Connected.
SQL> shutdown abort
ORACLE instance shut down.
SQL>
```

```
bash-3.00# tail /oracle/admin/PRIMARY/bdump/alert_PRIMARY.log
Wed Aug  6 11:00:03 2008
Thread 1 advanced to log sequence 61
  Current log# 2 seq# 61 mem# 0: /oradata/db_files/PRIMARY/redo02.log
Wed Aug  6 16:10:24 2008
Thread 1 advanced to log sequence 62
  Current log# 3 seq# 62 mem# 0: /oradata/db_files/PRIMARY/redo03.log
Wed Aug  6 16:48:15 2008
Shutting down instance (abort)
License high water mark = 4
Instance terminated by USER, pid = 9252
```

Checking the instance alert log indicated that the database was aborted at 16:48:15

On node 1 check to see the timestamp of the most recent CDR file

```
bash-3.00# ls -lrt /zfs1/input | tail
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_18_20088616432.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_19_20088616432.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_20_20088616433.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_21_20088616434.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_22_20088616435.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_23_20088616435.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_24_20088616436.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_25_20088616437.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_26_20088616438.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_27_20088616438.evt
```

This indicated that the most recent row in the PRIMARY database instance and most recent CDR file were created at 16:43

Begin failover to the STANDBY instance on node2 following section 7.2.2 of the ‘Oracle Data Guard Concepts and Administration 10g Release 2’ manual (Oracle 2006a).

Step 1 Identify and resolve any gaps in the archived redo log files.

```
amccorma.oracle.node2> (/oracle)
$ . oraenv
ORACLE_SID = [amccorma] ? STANDBY
amccorma.oracle.node2> (/oracle)
$
amccorma.oracle.node2> (/oracle)
$ sqlplus / as sysdba

SQL*Plus: Release 10.2.0.3.0 - Production on Wed Aug 6 16:54:49 2008

Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bit Production
With the Partitioning, OLAP and Data Mining options

SQL> SELECT THREAD#, LOW_SEQUENCE#, HIGH_SEQUENCE# FROM V$ARCHIVE_GAP;

no rows selected
```

Step 3 Copy any other missing archived redo log files.

```
SQL> SELECT UNIQUE THREAD# AS THREAD, MAX(SEQUENCE#)
2 OVER (PARTITION BY thread#) AS LAST from V$ARCHIVED_LOG;

  THREAD          LAST
-----
1          61
```

Step 4 Initiate a failover on the target physical standby database.

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE FINISH FORCE;

Database altered.
```

Step 5 Convert the physical standby database to the primary role.

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;

Database altered.
```

Step 6 Finish the transition of the standby database to the primary database role.

```
SQL> ALTER DATABASE OPEN;
```

```
Database altered.
```

The target physical standby database has now undergone a transition to the primary database role.

```

SQL> connect billing/billing
Connected.
SQL> ! date
Wed Aug  6 17:02:21 IST 2008

SQL> select to_char(max(stime),'hh24:mi:ss yyyy-mm-dd') as last_rowtime from cdr ;

LAST_ROWTIME
-----
17:38:00 2008-10-26

```

Database available at 17:02:28 after being aborted at 16:48.15 - 14 minutes 13 seconds to complete failover.

The most recent rows in the standby database instance were dated 17:38 – 5 Minutes 22 seconds behind the primary database at time of failover.

Listing the contents of the input directory on node 2 indicated that it was up to date with the primary system.

```

amccorma.oracle.node2> (/oracle)
$ ls -lrt /zfs1/input | tail
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_18_20088616432.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_19_20088616432.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_20_20088616433.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_21_20088616434.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_22_20088616435.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_23_20088616435.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_24_20088616436.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_25_20088616437.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_26_20088616438.evt
-rw-r--r-- 1 oracle oinstall 465000 Aug  6 16:43 cdrfile_27_20088616438.evt

```

This test indicated that the standby node could return to full operation in less than fifteen minutes after the failure of the primary node with a loss of less than 6 minutes worth of data.

5.0 CONCLUSION

This chapter contains the conclusions of the project and recommendations on areas for future research.

5.1 CONCLUSIONS

The goal of this project was to design a cost effective business continuity solution to allow the billing system of a small telecommunications service provider to continue operations on a standby site in the event of a natural disaster or other incident affecting their primary location.

The research carried out in chapter 4 identified the business continuity requirements for a small billing system and investigated the methods that could be used to implement a suitable solution. From this research, a combination of Oracle's Data Guard and Sun's ZFS filesystem replication were deemed suitable to provide the required level of disaster protection while still keeping the costs sufficiently low to appeal to a small telecommunications service provider with a limited IT budget.

A proof of concept test was carried out as detailed in chapter 6. This indicated that the proposed solution was capable of providing business continuity for the billing system by replicating all necessary data from the primary to the standby system and that in the event of a

failure of the primary system, the standby system could be made operational in less than 15 minutes (RTO) with less than 6 minutes worth of data lost (RPO).

5.2 RECOMMENDATIONS FOR FURTHER RESEARCH

The ZFS filesystem that is used for the proposed solution is a relatively new filesystem so there are areas that are still being developed and features still to be added. While it possesses impressive performance and capabilities now, future improvements should make it even more attractive an option on which to base a business continuity system on. Some areas which require future research are detailed below.

The proposed business continuity solution utilized ZFS snapshot replication for billing system files and Oracle DataGuard replication for the contents of the billing system database. It should also be possible to use ZFS snapshots to replicate the Oracle database files to the remote system by temporarily placing the database into hot backup mode and then taking a ZFS snapshot of the filesystem hosting the Oracle datafiles - e.g.

Place database into hot backup mode

```
SQL> ALTER DATABASE BEGIN BACKUP ;
```

Take snapshot of filesystem hosting Oracle datafiles

```
zfs snapshot zfs1/oradata@snap0
```

Take database out of hot backup mode

```
SQL> ALTER DATABASE END BACKUP ;
```

Replicate snapshot to standby system

```
zfs send zfs1/oradata@snap0 | ssh node2 zfs receive zfs1/oradata
```

As taking a snapshot only takes a few seconds, the amount of time the database would spend in backup mode would be minimal. By sending incremental snapshots, only changed data would need to be replicated across to the standby node.

The advantages of this approach is that it would mean that no Oracle licenses would need to be purchased for the standby system, providing it was only intended to be used in the event of a disaster affecting the primary site. This would however mean that the standby node could not be used as a reporting system. There are a number of disadvantages with this approach though – as this would result in replicating the entire Oracle database rather than just the redo logs, more network bandwidth would be required between primary and standby sites to replicate the same data. (Oracle Corporation n.d.b) This approach would also require that the Oracle datafiles be hosted on a ZFS filesystem rather than on the higher performing raw devices/ASM or UFS filesystem with direct I/O. While ZFS exhibits impressive performance gains over other filesystems such as VxFS and ext3 for most workloads, it currently lags both raw devices and UFS with direct I/O when it comes to Oracle OLTP database performance, making them a better choice for high performance databases. (Nadgir 2006), (Sun Microsystems 2007b). Future enhancements to ZFS are expected to significantly improve ZFS performance in this area, making the use of the free ZFS replication a potentially attractive option for use in business continuity solutions.

APPENDIX A

HARDWARE PRICING

This section contains list price quotations for the hardware and software specified in the proposed billing disaster recovery solution. List prices correct as of April 2008.

A.1.1 Sun StorageTek 2540 FC Array pricing

Your Configuration			
Item Description		Qty	Price (U.S.)
 <p>Sun StorageTek 2540 FC Array</p> <p>1 XTA2540R01A2E1752 \$ 14,320.00 Sun StorageTek 2540 FC Array, Rack-Ready Controller Tray, 1752 GB, 12 x 146 GB 15000 rpm SAS drives, 2 512 MB cache FC HW RAID controllers, 2 redundant AC power supplies, 2 redundant cooling fans, 4 shortwave SFPs, Includes Sun StorageTek Common Array Manager software and 2 storage domains using Sun StorageTek Storage Domains software, RoHS-5 Compliant Ships Within: 5 business days</p> <p>2 X311L \$ 0.00 Power Cord Kit, North American/Asian, RoHS Compliant Ships Within: 5 business days</p> <p>1 XTA2501R01A1E1752 \$ 9,220.00 Sun StorageTek 2501 SAS Expansion Array, Rack-Ready Expansion Tray, 1752 GB, 12 x 146 GB 15000 rpm SAS drives, 1 SAS I/O Module, 2 redundant AC power supplies and 2 redundant cooling fans, Includes 1 x 1m SAS host cable, RoHS-5 Compliant Ships Within: 6 business days</p> <p>2 X311L \$ 0.00 Power Cord Kit, North American/Asian, RoHS Compliant Ships Within: 5 business days</p>		1	\$ 23,540.00
			\$ 23,540.00

Copyright 1994-2007 Sun Microsystems, Inc.

A.1.2 Sun StorageTek 6140 Array pricing

Your Configuration

Item Description



Sun StorageTek 6140 Array

1	XTC6140R11D2C2336Z	\$ 53,415.00
RoHS-5, Sun StorageTek (tm) 6140 array with 4GB cache and 8 host ports, Rack-Ready Controller Tray, 2336GB, 16 * 146GB 15Krpm 4Gb FC-AL Drives, 2 * 2GB-cache memory FC RAID Controller cards, 2 * redundant DC power supplies and cooling fans, 2 * copper FC ports for expansion trays and 8 * host ports with shortwave SFPs, 2 * 5M fibre optic cables, 2 * 6M ethernet cables and management software, 3 yr on-site warranty included (Standard Configuration)		
Ships Within: 8 business days		
1	XTCCSM2R01D0C730Z	\$ 14,470.00
Sun StorageTek CSM200, Rack-Ready Expansion Tray, 730 GB, 5 x 146 GB 15000 rpm 4Gb FC-AL Drives, 2 I/O Modules, 2 redundant DC power supplies and cooling fans, 2 FC ports for expansions, 4 shortwave SFPs with 2 LC-LC FC cables, RoHS-5 Compliant		
Ships Within: 8 business days		
1	XTC6140-DRE-ARY	\$ 22,000.00
Sun StorageTek Enhanced Data Replicator Software Right-to-Use License Key for Sun StorageTek 6140 Array, Up to 64 Replication Mirrors, No Storage Capacity Limit		
Ships Within: Call Sun		
2	X311L	\$ 0.00
Power Cord Kit, North American/Asian, RoHS Compliant		
Ships Within: 5 business days		

Qty	Price (U.S.)	Total
1	\$ 89,885.00	\$ 89,885.00

A.1.3 Sun StorageTek Availability Suite pricing

Sun StorageTek Availability Suite Software



Sun StorageTek Availability Suite software helps reduce disruptions to your critical business applications and data, provide rapid recovery, and enable data repurposing, providing a solid foundation for a data continuance environment. The software features a point-in-time copy capability as well as a remote mirroring capability.

[Learn More](#)


[System Requirements](#)

Other Data Protection

- > Sun StorageTek Availability Suite Software
 - » Sun StorageTek Enterprise Backup
 - » Veritas NetBackup Server
 - » Veritas NetBackup Enterprise Server
 - » Veritas NetBackup Capacity Model

Select Your Software	
Make your choices using the menus below	
Software & Documentation	Sun StorageTek Availability Suite software media kit and documentation. Purchase version 4.0 for Solaris 10 support, or version 3.2 for Solaris 8 and 9 support. <div> <input type="text" value="Software & Documentation, V4.0"/> [+] \$150.00] </div>
License	License only. Software media kit and documentation must also be selected. Quantity <input type="text" value="3"/> <div> <input type="text"/> Volume License for up to 5 TB [+] \$37,500.00 each] <input type="text"/> Volume License for up to 50 TB [+] \$300,000.00 each] <input type="text"/> License per 1 TB of Master Online Storage [+] \$10,000.00 each] </div>
Your Selection	
Tax and shipping charges will be calculated when you check out.	
Ships Within	4 Business Days
Subtotal	<div> <input type="button" value="Update"/> </div> \$30,150.00 List Price
<div> <input type="button" value="Save / Send Quote"/> <input type="button" value="Add To Cart"/> </div>	

A.1.4 Sun T2000 server pricing

Your Configuration			
Item Description	Qty	Price (U.S.)	Total
 <p>Sun Fire T2000 Server</p> <p>1 T20Z108B-32GA2G \$ 20,495.00 Sun Fire T2000 Server, 8 Core, 1.2 GHz UltraSPARC T1 processor, 32 GB DDR2 memory (16 x 2 GB DIMMs), 2 x 73 GB 2.5-inch 10000 rpm SAS disks, 1 DVD-ROM/CD-RW slimline drive, 2 (N+1) power supplies, 4 x 10/100/1000 Ethernet ports, 1 Serial port, 3 PCIe slots, 2 PCI-X slots, Solaris 10 Operating System and Java Enterprise System software preinstalled, RoHS-5 Compliant Ships Within: 5 business days</p> <p>2 X311L \$ 0.00 Power Cord Kit, North American/Asian, RoHS Compliant Ships Within: 5 business days</p> <p>2 SG-XPCI1FC-QF4 \$ 1,100.00 4 Gb Single FC PCI-X Host Bus Adapter, includes Standard and Low Profile Brackets, RoHS-6 Compliant Ships Within: 5 business days</p> <p>1 SOLZ9-100C9A7M \$ 30.00 Solaris 10 (latest release) DVD Multilingual Media, No hardcopy documentation or license included, (Pricing per kit) Ships Within: 5 business days</p>	1	\$ 22,725.00	\$ 22,725.00

Copyright 1994-2007 Sun Microsystems, Inc.

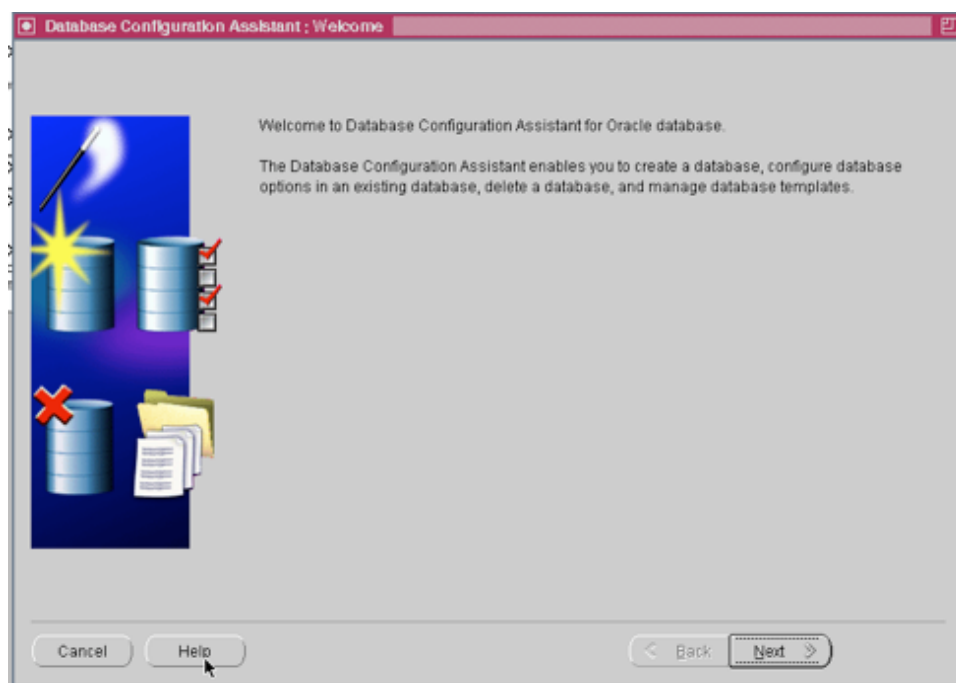
APPENDIX B

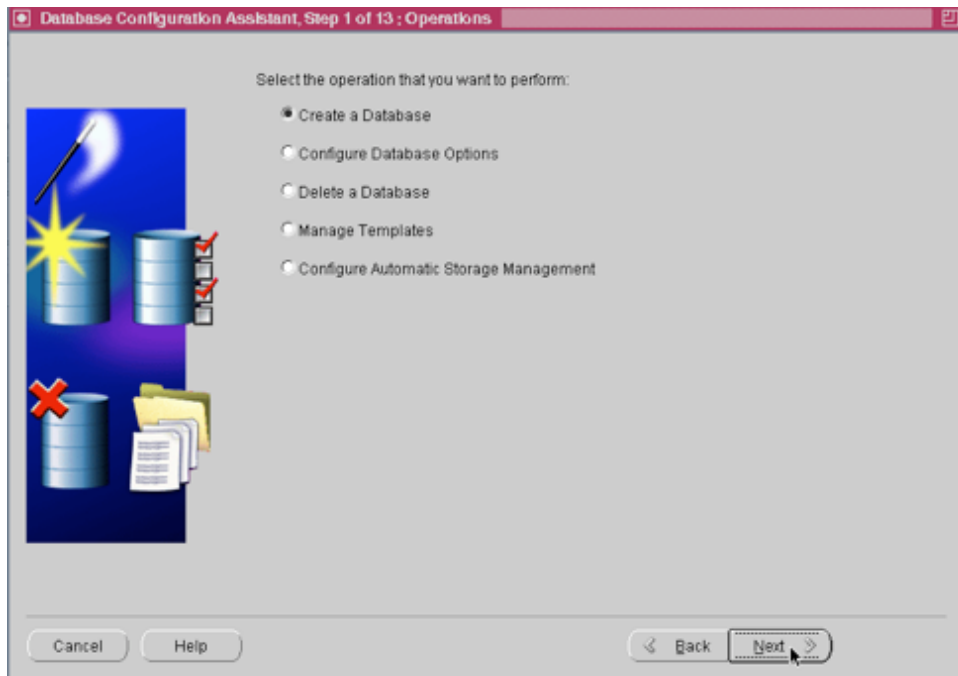
PRIMARY AND STANDBY DATABASE CONFIGURATION

This appendix contains details of how the primary and standby databases were created for the proof of concept test. Section B1 documents the creation of the PRIMARY database instance on node1. Section B2 documents the process of configuring a standby database on node2.

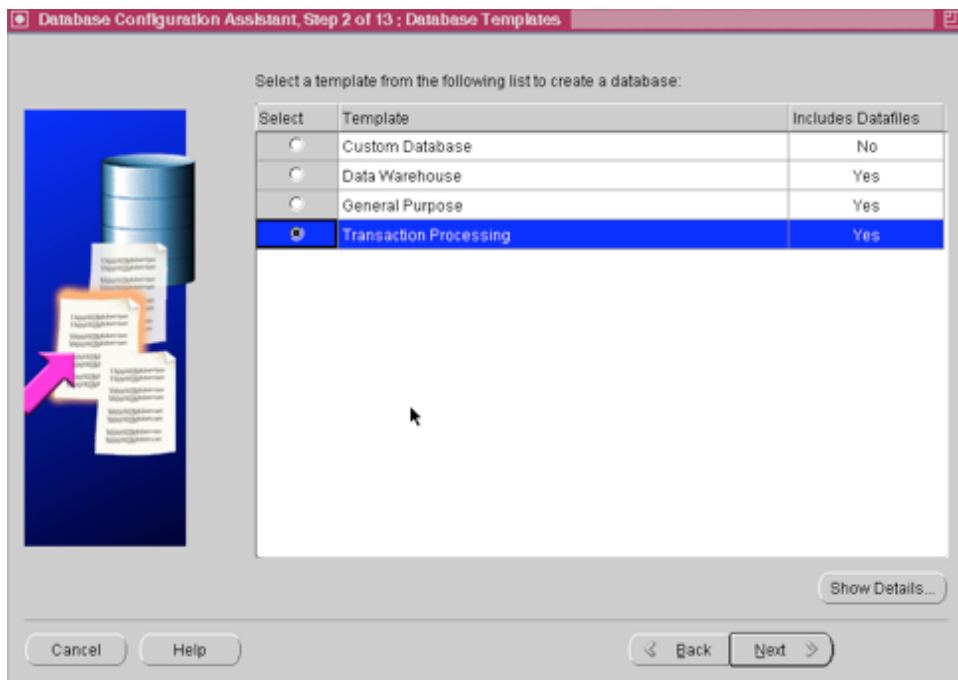
B.1 PRIMARY DATABASE CREATION

The Oracle database configuration assistant (DBCA) was launched on the primary node as the Oracle user by executing the command 'dbca'.

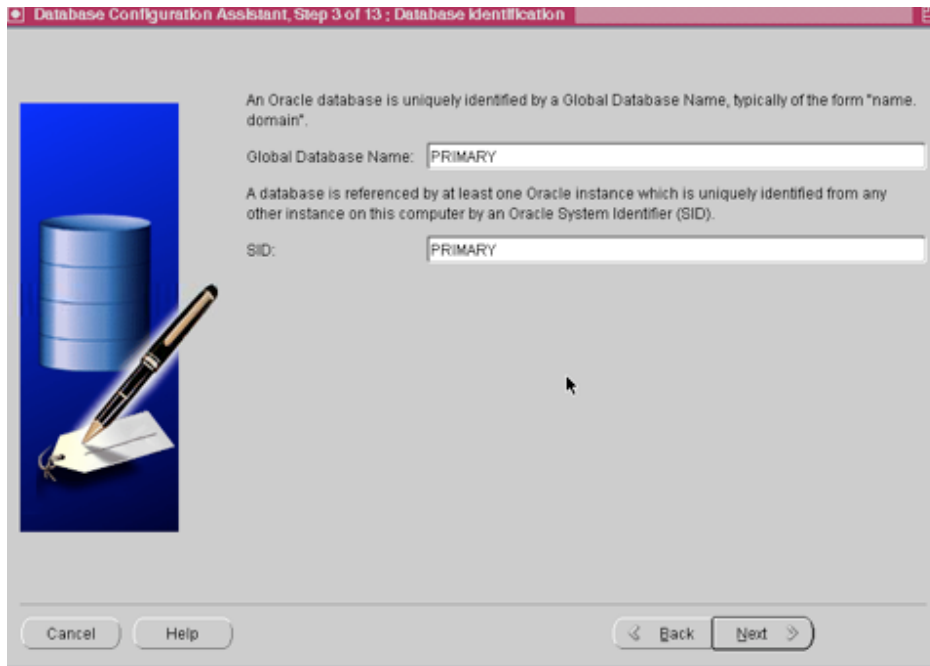




The 'create a database' option was selected



The 'Transaction Processing' template was selected.



Database Configuration Assistant, Step 3 of 13 : Database Identification

An Oracle database is uniquely identified by a Global Database Name, typically of the form "name.domain".

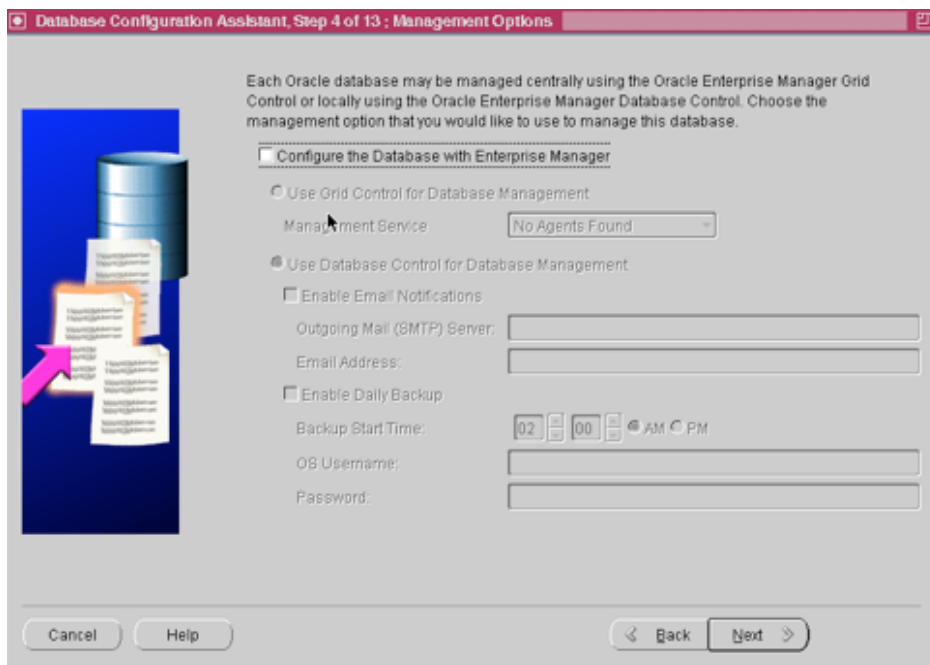
Global Database Name: PRIMARY

A database is referenced by at least one Oracle instance which is uniquely identified from any other instance on this computer by an Oracle System Identifier (SID).

SID: PRIMARY

Cancel Help Back Next

'PRIMARY' was chosen for the database name and SID.



Database Configuration Assistant, Step 4 of 13 : Management Options

Each Oracle database may be managed centrally using the Oracle Enterprise Manager Grid Control or locally using the Oracle Enterprise Manager Database Control. Choose the management option that you would like to use to manage this database.

☐ Configure the Database with Enterprise Manager

☐ Use Grid Control for Database Management

Management Service: No Agents Found

☒ Use Database Control for Database Management

☐ Enable Email Notifications

Outgoing Mail (SMTP) Server:

Email Address:

☐ Enable Daily Backup

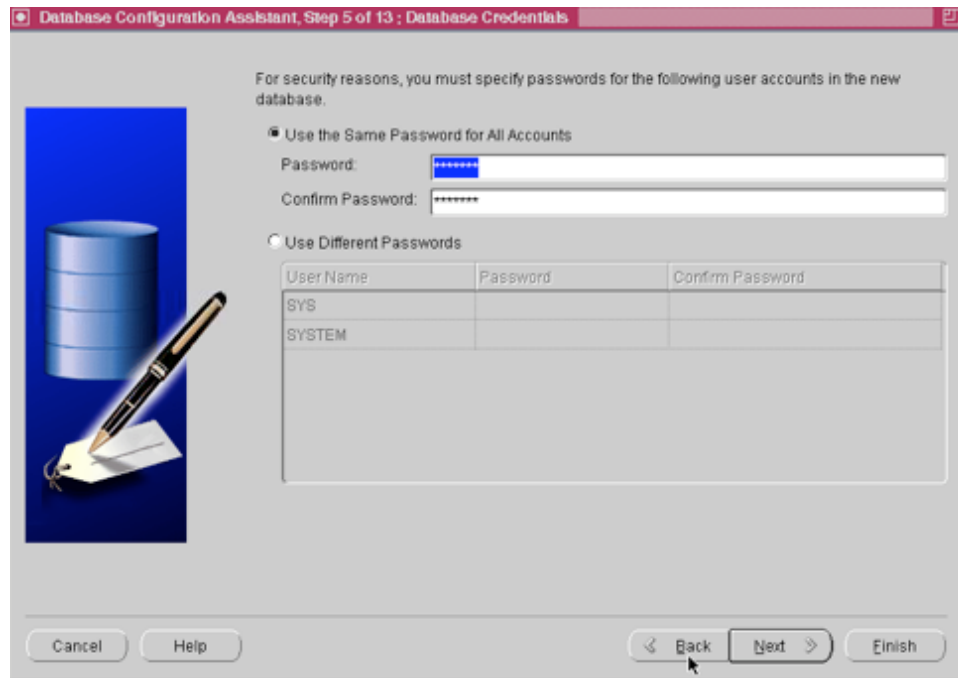
Backup Start Time: 02:00 AM PM

OS Username:

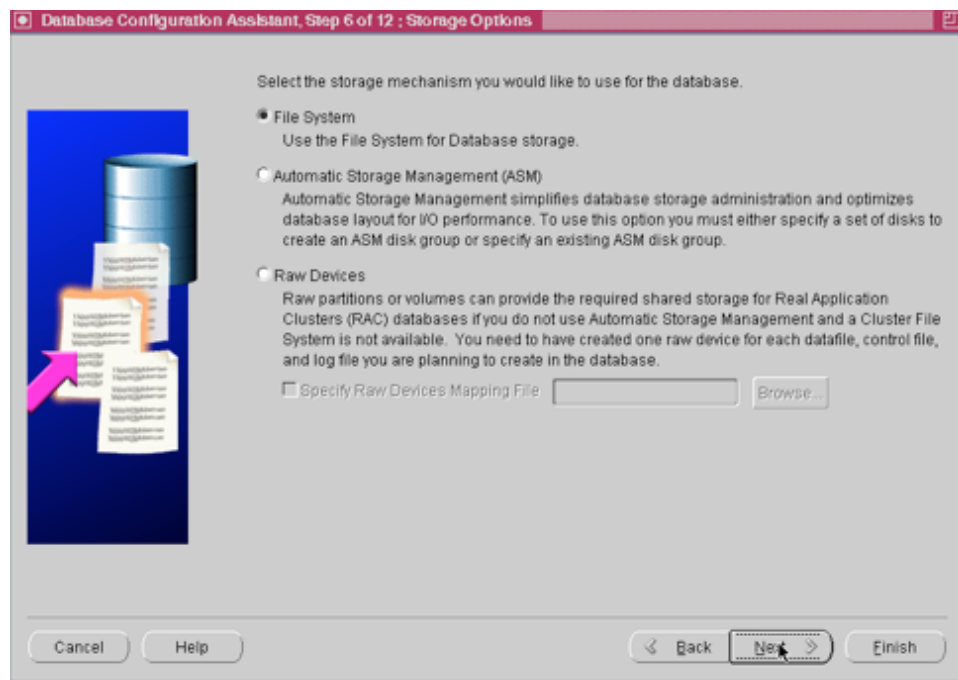
Password:

Cancel Help Back Next

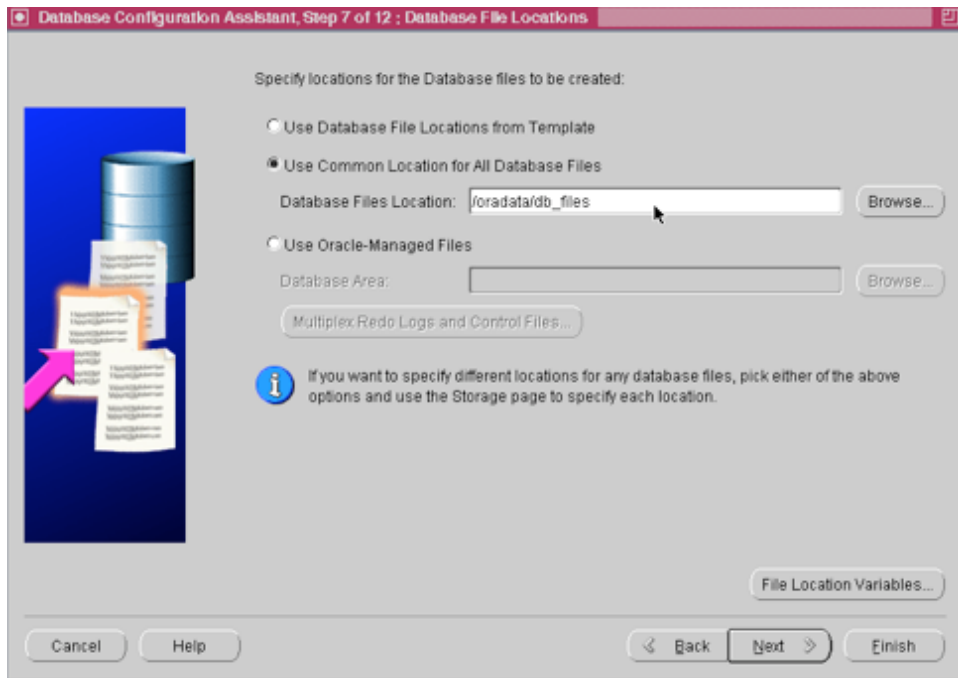
Enterprise manager was not selected for this database due to the limited memory resources of the test systems.



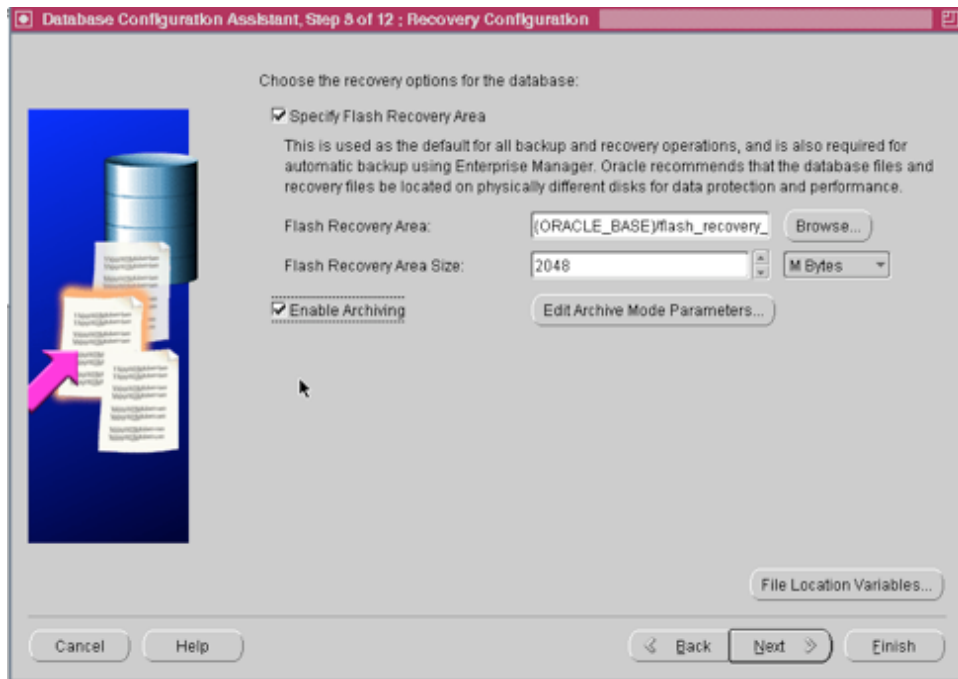
A password was entered for the SYS and SYSTEM accounts.



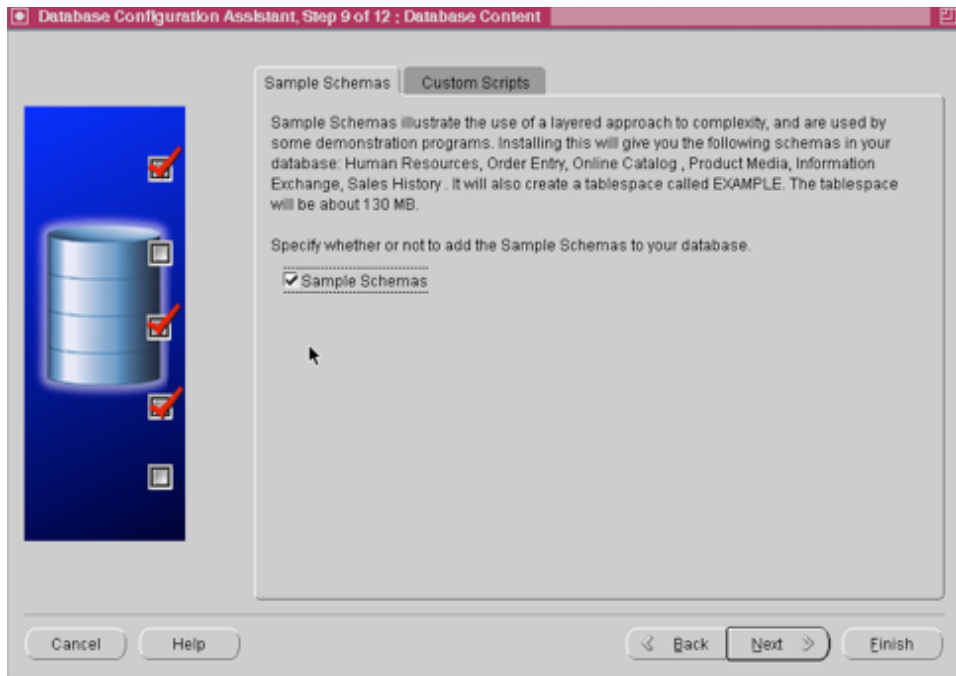
File System was chosen as the storage mechanism for the data files.



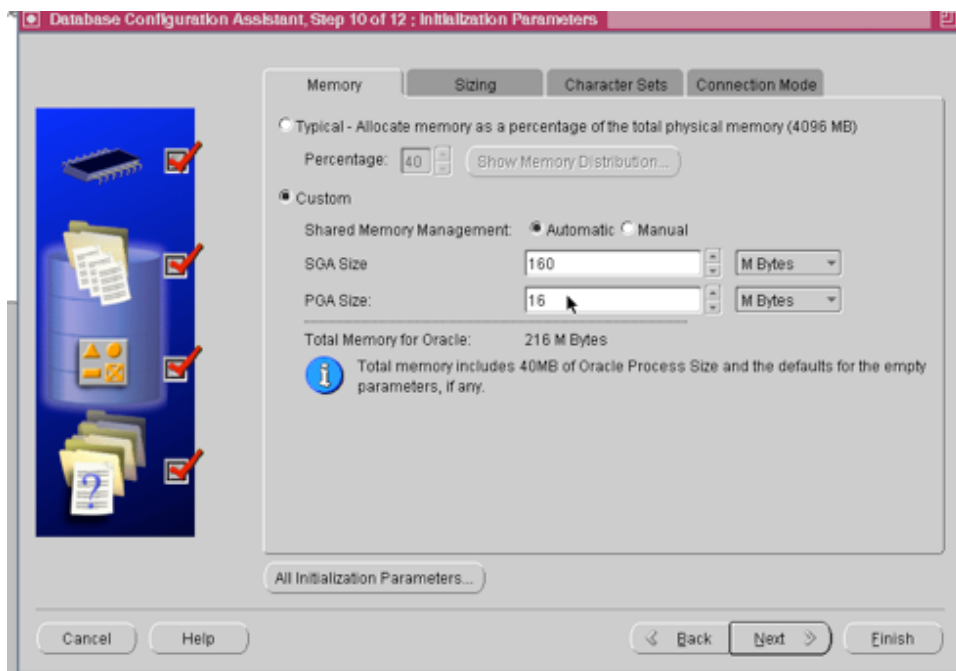
The datafile location was set to /oradata/db_files



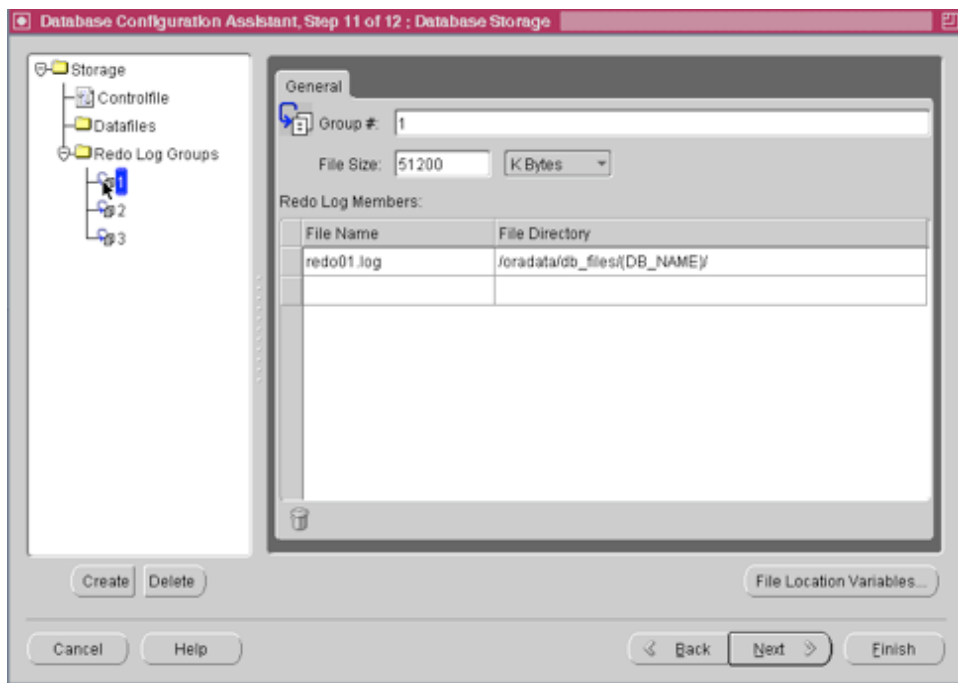
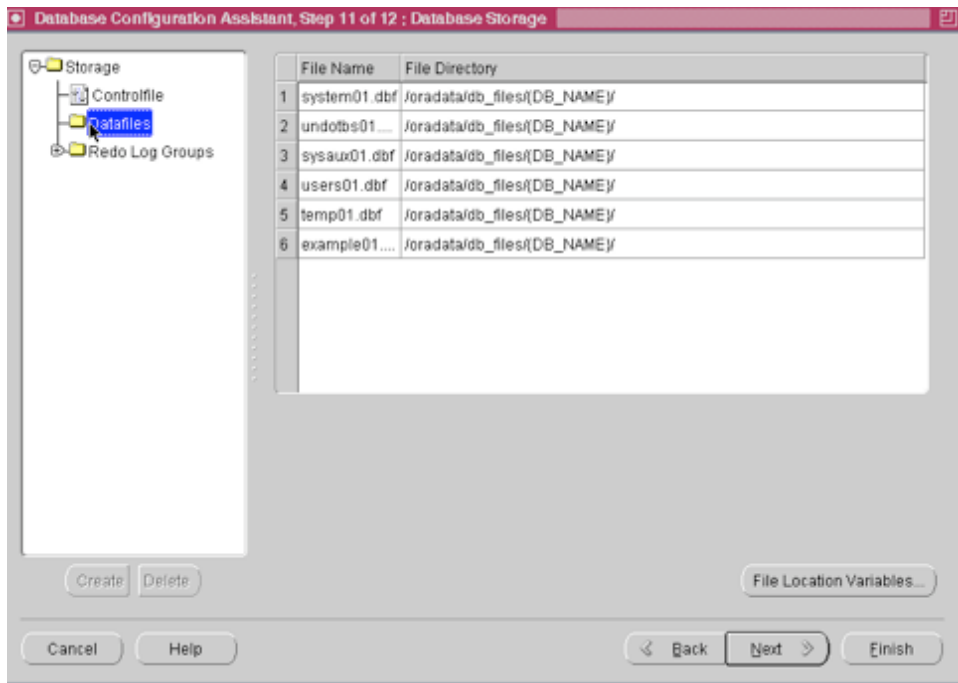
Archiving was enabled as this is required for Data Guard



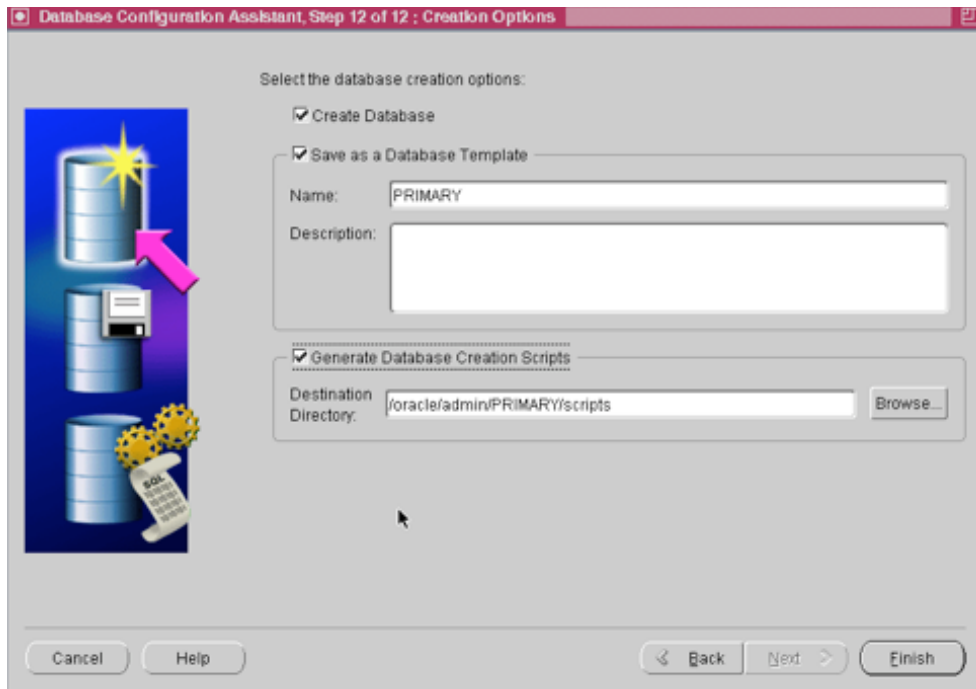
Sample schemas were included.



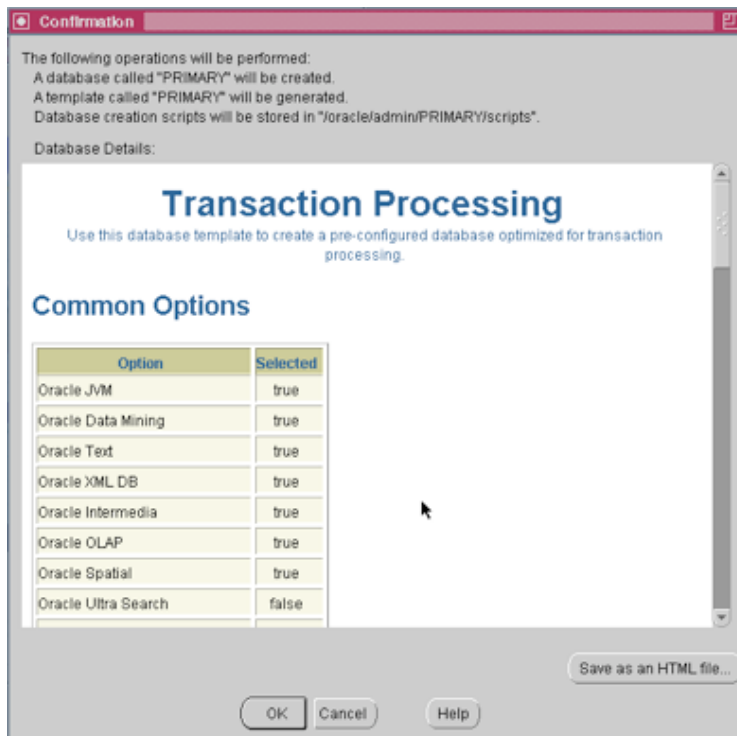
The Oracle memory parameters were reduced to the minimum values – in this case a total of 216MB was allocated. This was done due to the low hardware specification of the servers that were used for the test.

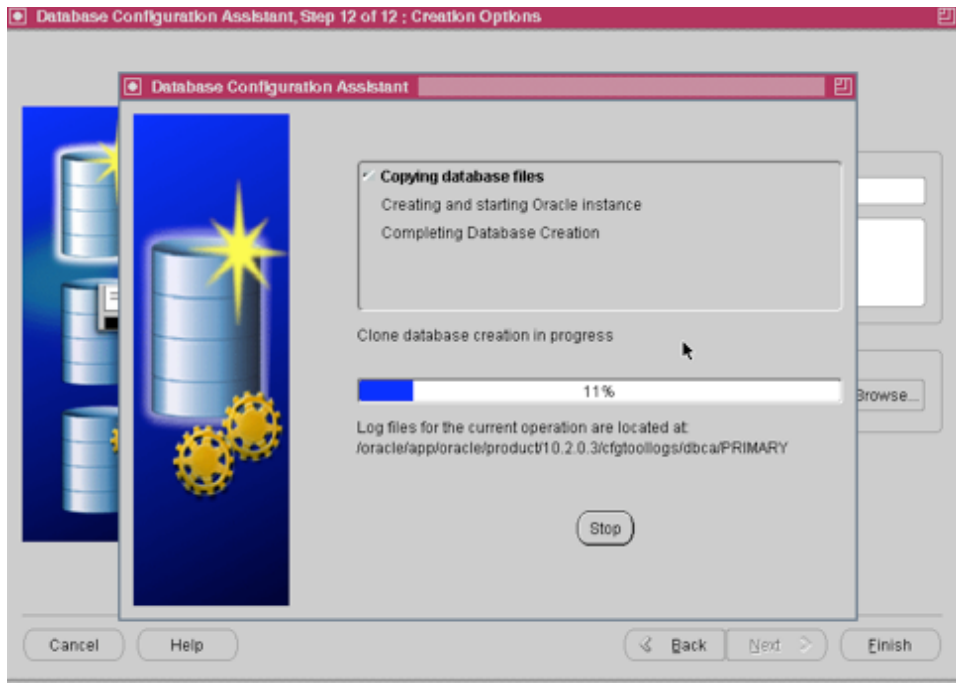


Review the datafile and redo log sizes and locations.

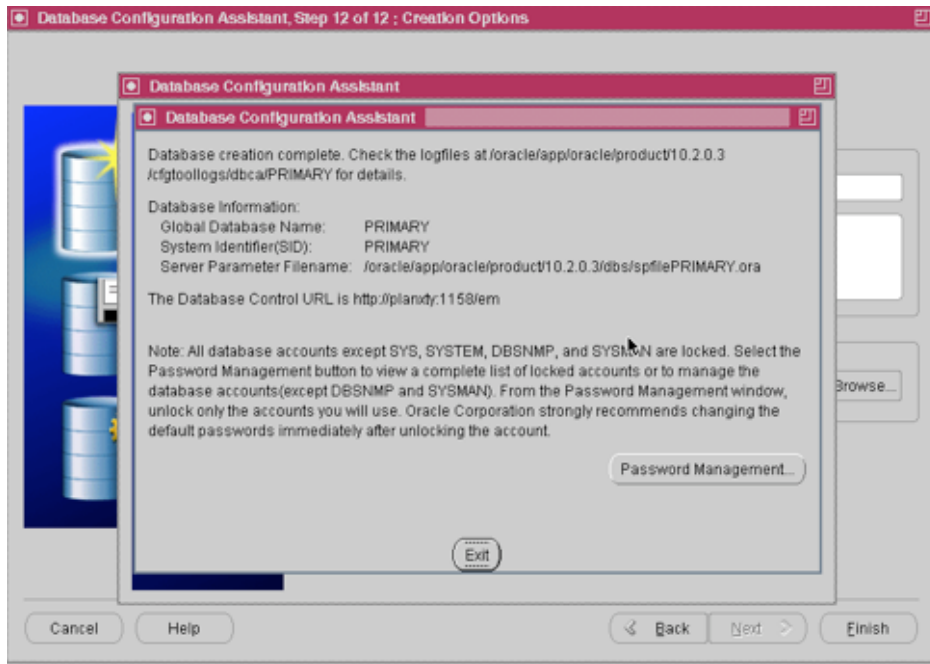


Create the database.





Database creation in progress



Database creation complete.

B.2 STANDBY DATABASE CREATION

Once the primary database was created on node1, a physical standby database was implemented on the standby server using DataGuard by following the instructions contained in section 3 of the Oracle 10g Data Guard Concepts and Administration manual (Oracle 2006a).

B.2.1 Prepare primary database for standby creation

Forced logging was enabled on the primary database using the following SQL command:

```
SQL> ALTER DATABASE FORCE LOGGING;
```

An Oracle password file was created for the primary database using the following Unix command:

```
orapwd file=$ORACLE_HOME/dbs/orapasswd password=<password> entries=64
```

<password> is a place holder for the password

Standby redo logs of the same size as the primary redo logs were created on the primary database. First the size of the primary redo logs was verified :

```
SQL> select a.member,a.group#,b.bytes/1024/1024 as MB
  2   from v$logfile a, v$log b
  3   where a.group#=b.group#
  4   ;
```

MEMBER	GROUP#	MB
/oradata/db_files/PRIMARY/redo03.log	3	50
/oradata/db_files/PRIMARY/redo02.log	2	50
/oradata/db_files/PRIMARY/redo01.log	1	50

Then 3 standby logs of 50MB in size were created:

```
alter database add standby logfile group 4 '/oradata/db_files/PRIMARY/redo01_s.log' size 50m
alter database add standby logfile group 5 '/oradata/db_files/PRIMARY/redo02_s.log' size 50m ;
alter database add standby logfile group 6 '/oradata/db_files/PRIMARY/redo03_s.log' size 50m ;
```

Check that the standby logs are created correctly:

```
SQL> select group# thread#,sequence#,archived,status from v$standby_log
```

```

      THREAD#  SEQUENCE# ARC STATUS
-----
          4             0 YES UNASSIGNED
          5             0 YES UNASSIGNED
          6             0 YES UNASSIGNED

```

The parameter file on the primary database was modified to add new parameters required for data guard. First a test pfile was created, then the database was shut down and the changes were made by editing that text file.

```
SQL> create pfile='/tmp/initPRIMARY.ora' from spfile ;
```

File created.

```
SQL> shutdown immediate
```

Database closed.

Database dismounted.

ORACLE instance shut down.

```
SQL>
```

```
SQL> exit
```

```
Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bit Production
With the Partitioning, OLAP and Data Mining options
root.oracle.nodel> (/oracle)
```

```
$ vi /tmp/initPRIMARY.ora
```

The modifications to the original spfile are highlighted in bold.

```
PRIMARY.__db_cache_size=50331648
PRIMARY.__java_pool_size=4194304
PRIMARY.__large_pool_size=4194304
PRIMARY.__shared_pool_size=100663296
PRIMARY.__streams_pool_size=0
*.audit_file_dest='/oracle/admin/PRIMARY/adump'
*.background_dump_dest='/oracle/admin/PRIMARY/bdump'
*.compatible='10.2.0.3.0'
*.control_files='/oradata/db_files/PRIMARY/control01.ctl','/oradata/db_files/PRIMARY/control02.ctl',
'/oradata/db_files/PRIMARY/control03.ctl'
*.core_dump_dest='/oracle/admin/PRIMARY/cdump'
*.db_block_size=8192
*.db_domain=''
```

```

*.db_file_multiblock_read_count=8
*.db_name='PRIMARY'
*.db_recovery_file_dest='/oracle/flash_recovery_area'
*.db_recovery_file_dest_size=2147483648
*.dispatchers='(PROTOCOL=TCP) (SERVICE=PRIMARYXDB)'
*.job_queue_processes=2
*.local_listener=''
*.log_archive_format='%t_%s_%r.dbf'
*.open_cursors=300
*.pga_aggregate_target=16777216
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=167772160
*.undo_management='AUTO'
*.undo_tablespace='UNDOTBS1'
*.user_dump_dest='/oracle/admin/PRIMARY/udump'
*.DB_UNIQUE_NAME=PRIMARY
*.LOG_ARCHIVE_CONFIG='DG_CONFIG=(PRIMARY,STANDBY)'
*.LOG_ARCHIVE_DEST_1=
'LOCATION=/archive1/PRIMARY/
VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
DB_UNIQUE_NAME=PRIMARY'
*.LOG_ARCHIVE_DEST_2=
'SERVICE=STANDBY LGWR ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=STANDBY'
*.LOG_ARCHIVE_DEST_STATE_1=ENABLE
*.LOG_ARCHIVE_DEST_STATE_2=ENABLE
*.REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE
*.LOG_ARCHIVE_FORMAT='%t_%s_%r.arc'
*.LOG_ARCHIVE_MAX_PROCESSES=30
*.FAL_SERVER=STANDBY
*.FAL_CLIENT=PRIMARY

```

After modifying the parameter file, the changes were applied to the spfile and the database restarted.

```
$ sqlplus / as sysdba
```

```

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Jul 25 20:07:55 2008
Copyright (c) 1982, 2006, Oracle. All Rights Reserved.
Connected to an idle instance.

```

```
SQL> create spfile from pfile='/tmp/initPRIMARY.ora' ;
```

```
File created.
```

```

SQL> startup
ORACLE instance started.
Total System Global Area 167772160 bytes
Fixed Size 2028624 bytes
Variable Size 113249200 bytes
Database Buffers 46137344 bytes
Redo Buffers 6356992 bytes
Database mounted.
Database opened.
SQL>

```


B.2.2 Configure Oracle Net listener and service names.

Oracle Network manager was used to create a listener called LISTENER_PRIMARY on the primary server and a listener called LISTENER_STANDBY on the standby server. This resulted in the following entries being added to \$ORACLE_HOME/network/admin/listener.ora on the respective nodes:

Node1

```
LISTENER_PRIMARY =  
  (DESCRIPTION_LIST =  
    (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = TCP) (HOST = node1) (PORT = 1521))  
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC0))  
    )  
  )
```

Node2

```
LISTENER_STANDBY =  
  (DESCRIPTION_LIST =  
    (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = TCP) (HOST = node2) (PORT = 1521))  
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC0))  
    )  
  )
```

Oracle Network manager was then used to create PRIMARY and STANDBY services on their respective servers. This resulted in the following entries being added to \$ORACLE_HOME/network/admin/tnsnames.ora

```
PRIMARY.WORLD =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = node1) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = PRIMARY)
    )
  )

STANDBY.WORLD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = node2) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = STANDBY)
    )
  )
```

The listeners on each node were started using the lsnrctl command follows:

```
lsnrctl start LISTENER_PRIMARY
```

Then connectivity was verified using the tnsping command:

```
$ tnsping PRIMARY

TNS Ping Utility for Solaris: Version 10.2.0.3.0 - Production on 27-JUL-2008 19:29:40

Copyright (c) 1997, 2006, Oracle. All rights reserved.

Used parameter files:
/oracle/app/oracle/product/10.2.0.3/network/admin/sqlnet.ora

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = node1) (PORT =
1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = PRIMARY)))
OK (10 msec)
```

This was carried out on both node1 and node2 to verify that the PRIMARY and STANDBY services were accessible from either node.

B.2.3 Create standby database

A ZFS snapshot of the primary database datafiles was taken while the primary database was shut down. This snapshot was replicated to the standby server using the following commands:

On node 1 create a snapshot of zfs1/oradata called snap0:

```
zfs snapshot zfs1/oradata@snap0
```

On node1 send that snapshot to node2 by piping it through ssh:

```
zfs send zfs1/oradata@snap0 | ssh -C node2 zfs receive zfs1/oradata
```

On node2, set the mountpoint for zfs1/oradata to be /oradata:

```
zfs set mountpoint=/oradata zfs1/oradata
```

On node2 verify that the snapshot sent has been received and mounted:

```
bash-3.00# zfs list
```

```
$ ssh node2 /sbin/zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
zfs1	9.46G	7.15G	24.5K	/zfs1
zfs1/archive1	716M	7.15G	716M	/archive1
zfs1/oracle	4.65G	7.15G	4.65G	/oracle
zfs1/oracle@snap0	913K	-	4.65G	-
zfs1/oradata	4.11G	7.15G	1.14G	/oradata
zfs1/oradata@snap0	2.97G	-	3.83G	-

```
df -k /oradata
```

Filesystem	kbytes	used	avail	capacity	Mounted on
zfs1/oradata	17418240	1199654	7495759	14%	/oradata

The primary database was then re-started and a standby control file was created as follows:

```
SQL> STARTUP MOUNT;
```

```
SQL> ALTER DATABASE CREATE STANDBY CONTROLFILE AS '/tmp/standby.ctl';
```

```
SQL> ALTER DATABASE OPEN;
```

An initialization parameter file for the standby database was created from the primary database as follows:

```
SQL> CREATE PFILE='/tmp/initSTANDBY.ora' FROM SPFILE;
```

The initSTANDBY.ora file was edited to make the following changes:
(changes from primary in **bold**)

```
PRIMARY.__db_cache_size=41943040
PRIMARY.__java_pool_size=4194304
PRIMARY.__large_pool_size=4194304
PRIMARY.__shared_pool_size=109051904
PRIMARY.__streams_pool_size=0
*.audit_file_dest='/oracle/admin/PRIMARY/adump'
*.background_dump_dest='/oracle/admin/PRIMARY/bdump'
*.compatible='10.2.0.3.0'
*.control_files='/oradata/db_files/PRIMARY/control01.ctl','/oradata/db_files/PRIMARY/control02.ctl','/oradata/db_files/PRIMARY/control03.ctl'
*.core_dump_dest='/oracle/admin/PRIMARY/cdump'
*.db_block_size=8192
*.db_domain=''
*.db_file_multiblock_read_count=8
*.db_name='PRIMARY'
*.db_recovery_file_dest='/oracle/flash_recovery_area'
*.db_recovery_file_dest_size=2147483648
*.DB_UNIQUE_NAME='STANDBY'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=PRIMARYXDB)'
*.FAL_CLIENT='STANDBY'
*.FAL_SERVER='PRIMARY'
*.job_queue_processes=2
*.local_listener=''
*.LOG_ARCHIVE_CONFIG='DG_CONFIG=(PRIMARY,STANDBY)'
*.LOG_ARCHIVE_DEST_1='LOCATION=/archive1/STANDBY/VALID_FOR=(ALL_LOGFILES,ALL_ROLES)DB_UNIQUE_NAME=STANDBY'
*.LOG_ARCHIVE_DEST_2='SERVICE=PRIMARY LGWR ASYNCVALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)DB_UNIQUE_NAME=PRIMARY'
*.LOG_ARCHIVE_DEST_STATE_1='ENABLE'
*.LOG_ARCHIVE_DEST_STATE_2='ENABLE'
*.LOG_ARCHIVE_FORMAT='%t_%s_%r.arc'
*.LOG_ARCHIVE_MAX_PROCESSES=30
*.LOG_FILE_NAME_CONVERT='/archive1/STANDBY','/archive1/PRIMARY/'
*.open_cursors=300
*.pga_aggregate_target=16777216
*.processes=150
*.REMOTE_LOGIN_PASSWORDFILE='EXCLUSIVE'
*.sga_target=167772160
*.STANDBY_FILE_MANAGEMENT='AUTO'
*.undo_management='AUTO'
*.undo_tablespace='UNDOTBS1'
*.user_dump_dest='/oracle/admin/PRIMARY/udump'
```

The standby control file, Oracle password file and initSTANDBY.ora files created above were copied over to the standby server using the ssh scp command:

```
scp /tmp/initSTANDBY.ora node2:/tmp
scp /tmp/standby.ctl node2:/tmp

scp /oracle/app/oracle/product/10.2.0.3/dbs/orapwPRIMARY \
node2:/oracle/app/oracle/product/10.2.0.3/dbs/orapwPRIMARY
```

The standby control file was then copied over the original control files on node2:

```
oracle.node2>cp /tmp/standby.ctl /oradata/db_files/PRIMARY/control01.ctl
oracle.node2>cp /tmp/standby.ctl /oradata/db_files/PRIMARY/control02.ctl
oracle.node2>cp /tmp/standby.ctl /oradata/db_files/PRIMARY/control03.ctl
```

The standby database was brought up using the following commands:

```
root.oracle.node2>sqlplus / as sysdba

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Jul 25 20:55:12 2008
Copyright (c) 1982, 2006, Oracle. All Rights Reserved.
Connected to an idle instance.

SQL> create spfile from pfile='/tmp/initSTANDBY.ora' ;

File created.

SQL> startup mount

ORACLE instance started.

Total System Global Area 163577856 bytes
Fixed Size 2028624 bytes
Variable Size 79694768 bytes
Database Buffers 79691776 bytes
Redo Buffers 2162688 bytes
Database mounted.
SQL>
SQL> alter database recover managed standby database disconnect from session ;

Database altered.

SQL>
```

At this point, the standby database was operational and receiving redo log updates from the primary database.

B.2.4 Verify standby database

To verify that the standby database was correctly receiving database modifications from the primary, the V\$ARCHIVED_LOG view was queried on the standby database:

```
SQL> SELECT SEQUENCE#, FIRST_TIME, NEXT_TIME FROM V$ARCHIVED_LOG ORDER BY SEQUENCE#;
```

SEQUENCE#	FIRST_TIM	NEXT_TIME
2	18-JUL-08	18-JUL-08
3	18-JUL-08	19-JUL-08
4	19-JUL-08	19-JUL-08
5	19-JUL-08	19-JUL-08
6	19-JUL-08	20-JUL-08
7	20-JUL-08	21-JUL-08
8	21-JUL-08	21-JUL-08
9	21-JUL-08	22-JUL-08
10	22-JUL-08	22-JUL-08
11	22-JUL-08	22-JUL-08
12	22-JUL-08	23-JUL-08
13	23-JUL-08	25-JUL-08
14	25-JUL-08	25-JUL-08
15	25-JUL-08	25-JUL-08
16	25-JUL-08	25-JUL-08
17	25-JUL-08	25-JUL-08
18	25-JUL-08	25-JUL-08
19	25-JUL-08	25-JUL-08
20	25-JUL-08	25-JUL-08
21	25-JUL-08	25-JUL-08
22	25-JUL-08	25-JUL-08
23	25-JUL-08	25-JUL-08
24	25-JUL-08	26-JUL-08

Then a logfile switch was carried out on the primary:

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

Then the V\$ARCHIVED_LOG view was again queried on the standby database:

```
SQL> /
```

SEQUENCE#	FIRST_TIM	NEXT_TIME
2	18-JUL-08	18-JUL-08
3	18-JUL-08	19-JUL-08
4	19-JUL-08	19-JUL-08
5	19-JUL-08	19-JUL-08
6	19-JUL-08	20-JUL-08
7	20-JUL-08	21-JUL-08
8	21-JUL-08	21-JUL-08
9	21-JUL-08	22-JUL-08
10	22-JUL-08	22-JUL-08
11	22-JUL-08	22-JUL-08
12	22-JUL-08	23-JUL-08
13	23-JUL-08	25-JUL-08
14	25-JUL-08	25-JUL-08

```

15 25-JUL-08 25-JUL-08
16 25-JUL-08 25-JUL-08
17 25-JUL-08 25-JUL-08
18 25-JUL-08 25-JUL-08
19 25-JUL-08 25-JUL-08
20 25-JUL-08 25-JUL-08
21 25-JUL-08 25-JUL-08
22 25-JUL-08 25-JUL-08
23 25-JUL-08 25-JUL-08
24 25-JUL-08 26-JUL-08
25 26-JUL-08 26-JUL-08

```

The presence of the archived log with sequence 25 indicated that the log switch on the primary node had been received successfully on the standby. The following query confirmed that the logs were being successfully applied.

```
SQL> SELECT SEQUENCE#,APPLIED FROM V$ARCHIVED_LOG ORDER BY SEQUENCE#;
```

```

SEQUENCE#  APP
-----  ---
2         NO
3         NO
4         NO
5         NO
6         NO
7         NO
8         NO
9         NO
10        NO
11        NO
12        YES
13        YES
14        YES
15        YES
16        YES
17        YES
18        YES
19        YES
20        YES
21        YES
22        YES
23        YES
24        YES
25        YES

```

At this point the standby database was operational and receiving replicated redo logs from the primary database.

BIBLIOGRAPHY

Babineau B. (2006) Double up with database data protection and disaster recovery with Oracle Data Guard

Retrieved March 12th 2008 from <http://www.oracle.com/database/docs/ESG-WP-Oracle-Data-Guard.pdf>

This white paper from the Enterprise Strategy Group gives an overview of the advantages of Oracle's Data Guard database replication solution and provides case studies for a number of enterprises that have implemented it. Notable points are Barnes and Noble case study where they used Data Guard to provide a 5 minute RPO and 45 minute RTO for their critical 400GB database, also the fact that they chose Data Guard over a remote mirroring solution due to Data Guard's consistency and data integrity features, despite the fact that they already had the bandwidth and facility to use remote mirroring solutions such as SRDF. Although written by the Enterprise Strategy Group, this white paper is available from the Oracle web page and may have been sponsored by Oracle.

Cisco (1998) Call Detail Records

Retrieved March 12th 2008 from
http://www.cisco.com/univercd/cc/td/doc/product/wanbu/das/das_1_4/das14/das14apd.htm

This document from Cisco describes the Call detail record (CDR) format used in one of their network switching products. The CDR format was used to produce sample data for the proof of concept billing system DR test.

Citrix Data Security, Regulatory Compliance, Disaster Recovery

Retrieved March 12th 2008 from
http://www.silicon.com/i/s/wp/spnsr/US_DataSecurityWhitepaper_0106%5b1%5d.pdf

This white paper from Citrix looks at the importance of data security and disaster recovery in the context of new regulatory requirements such as Sarbanes-Oxley, HIPAA and Basel II.

Eagle Rock Alliance LTD. (2001) 2001 Cost of downtime survey

Retrieved March 10th 2008 from

<http://www.contingencyplanningresearch.com/2001%20Survey.pdf>

This document, a joint report from Eagle Rock Alliance and Contingency planning and management magazine contains the results of a survey carried out in 2001 to assess the importance companies placed on contingency planning and how they would be affected both in monetary and non-monetary terms by an IT systems outage.

EMC. (2008a) EMC CLARiiON Systems: How They Compare

Retrieved 27 March 2008 from <http://uk.emc.com/collateral/hardware/comparison/emc-clarion.htm>

This web page from EMC provides an overview of their mid-range Clariion storage array systems, describing their capacities and the host platforms and mirroring technologies they are compatible with.

EMC. (2008b) EMC Symmetrix DMX Series: How They Compare

Retrieved 27 March 2008 from <http://uk.emc.com/collateral/hardware/comparison/emc-symmetrix-dmx.htm>

This web page from EMC provides an overview of their high-end Symmetrix storage array systems, describing their capacities and the host platforms and mirroring technologies they are compatible with.

EMC. (2008c) EMC Support Matrix

Retrieved 27 March 2008 from <http://uk.emc.com/collateral/elab/emc-support-matrices.pdf>

This document provides compatibility information for all of EMCs products, detailing what host platforms are supported by each product.

EMC. (2008d) MirrorView Local and remote data mirroring

Retrieved 27 March 2008 from <http://uk.emc.com/products/detail/software/mirrorview.htm>

This web page from EMC provides an overview of their MirrorView remote mirroring product, which is used for remote mirroring on the mid-range Clariion disk array systems.

EMC. (2008e) EMC SRDF family

Retrieved 27 March 2008 from <http://uk.emc.com/collateral/software/data-sheet/1523-emc-srdf.pdf>

This white paper from EMC provides an overview of their Symmetrix Remote Data Facility (SRDF) product, which is used for remote mirroring on the high-end Symmetrix disk array systems.

Hewlett Packard (2008) HP Storage Works Arrays family guide.

Retrieved March 16th 2008 from <http://h71028.www7.hp.com/ERC/downloads/4AA0-7118ENW.pdf>

This data sheet from HP provides an overview of the entry, mid-range and high-end disk storage arrays available from HP. It lists their maximum storage capacities, hardware support and additional software options such as the 'Continuous Access' SAN based remote mirroring software.

Hunter J, Thiebaud M (2003) Telecommunications billing systems

New York NY: McGraw Hill

This book provides a comprehensive reference to the operations of a telecommunications billing system.

IBM (2008a) System storage DS3000 interoperability matrix.

Retrieved March 17th 2008 from <http://www-03.ibm.com/systems/storage/disk/ds3000/pdf/interop.pdf>

This document from IBM contains details of the operating systems and hardware platforms supported by the entry-level DS3000 class disk arrays. Of note, these disk arrays do not support the Solaris operating system

IBM (2008b) System storage DS4000 interoperability matrix.

Retrieved March 17th 2008 from <http://www-03.ibm.com/systems/storage/disk/ds4000/pdf/interop-matrix.pdf>

This document from IBM contains details of the operating systems and hardware platforms supported by the mid-range DS4000 class disk arrays. These disk arrays do support the Solaris operating system

IBM (2008c) System storage DS6000 interoperability matrix.

Retrieved March 17th 2008 from <http://www-03.ibm.com/systems/storage/disk/ds6000/pdf/interop.pdf>

This document from IBM contains details of the operating systems and hardware platforms supported by the high-end DS6000 class disk arrays. These disk arrays do support the Solaris operating system, however they do not yet appear to support the recently introduced Sun T5110/T5120 servers.

IBM (2008d) System storage DS8000 interoperability matrix.

Retrieved March 17th 2008 from <http://www-03.ibm.com/systems/storage/disk/ds8000/interop.pdf>

This document from IBM contains details of the operating systems and hardware platforms supported by the high-end DS8000 class disk arrays. These disk arrays do support the Solaris operating system, however they do not yet appear to support the recently introduced Sun T5110/T5120 servers.

IBM (2008e) System storage DS8000 and DS6000 Advanced Copy and Mirroring Functions

Retrieved March 17th 2008 from http://www.ibm.com/common/ssi/fcgi-bin/ssialias?infotype=PM&subtype=SP&appname=STG_TS_USEN&htmlfid=TSS00241USEN&attachment=TSS00241USEN.PDF

This document from IBM describes the Global/Metro mirror remote mirroring software, which is available for IBM's mid-range and high-end disk arrays. Metro Mirror is a synchronous mirroring solution for distances up to 300km, Global Mirror is an asynchronous mirroring solution for longer distances.

IBM (2008f) IBM System storage product guide

Retrieved March 17th 2008 from <http://www-03.ibm.com/systems/storage/resource/pguide/prodguidedisk.pdf>

This document from IBM provides an overview of the capabilities and capacities of IBM's disk array products.

IBM (2004) IBM System Storage DS4000 Storage Manager v10.10 Copy Services Users Guide

Retrieved March 17th 2008 from ftp://ftp.software.ibm.com/systems/support/system_x_pdf/gc27217200.pdf

This manual from IBM provides details on the configuration and use of the enhanced remote mirroring facility on the mid-range DS4000 class disk arrays. Of note, synchronous replication is only supported up to distances of 10km

Intec (2008) Intec Convergent Billing achieves outstanding real-time performance on distributed grid of Sun CoolThreads Server

Retrieved August 9th 2008 from

<http://www.intecbilling.com/Intec/Media/Press+Releases/2008/Intec+Convergent+Billing+achieves+outstanding+real+time+performance+on+distributed+grid+of+Sun+CoolThreads>

This press release from Intec provides information regarding the use of a cluster of Sun T2000 servers to support billing and rating at a rate of up to 6 Million busy hour call attempts (BHCA) , which indicates that a single T2000 server is capable of supporting at least 1 Million subscribers.

Ixion (2005) ixPropogator

Retrieved March 16th 2008 from

http://www.ixionsoftware.com/products/ixprop_comparison.php

This data sheet provides an overview of Ixion's ixPropogator – a log based database replication solution that operates in a manner similar to Oracle's Data Guard.

Nadgir N (2006) Databases and ZFS

Retrieved August 1st 2008 from http://blogs.sun.com/realneel/entry/zfs_and_databases

This blog entry by a member of Sun's Performance Engineering team looks at the relative performance of the ZFS and UFS filesystems when used with an OLTP database workload. It indicates that while ZFS is faster than untuned UFS, it lags UFS performance once UFS is tuned for databases by utilizing the direct I/O (unbuffered I/O) features. It also points out that ZFS tuning is also possible and that as ZFS is a relatively new filesystem, future enhancements should improve its performance for OLTP databases.

OpenSolaris Community (2007) What is ZFS

Retrieved April 12th 2008 from <http://www.opensolaris.org/os/community/zfs/whatis>

This web page from the OpenSolaris community web site provides an overview of the main features of Sun's new ZFS filesystem.

Ofrane A, Harte L (2004) An introduction to Telecom Billing

Fuquay-Varina NC: Althos Publishing

This book provides an overview of how telecommunications billing systems operate, covering the major components such as rating, billing, invoicing and interfacing to external systems.

Oracle Corporation (n.d.a) Overview of Oracle's Data Protection and Disaster Recovery Solutions

Retrieved March 12th 2008 from

<http://www.oracle.com/technology/deploy/availability/htdocs/OracleDRSolutions.html>

This document from Oracle provides a brief overview of all the various Oracle technologies that can be used to provide data protection and disaster recovery for an Oracle database. They strongly recommend that Oracle Data Guard should be the preferred solution.

Oracle Corporation (n.d.b) Oracle Data Guard and Remote Mirroring Solutions

Retrieved March 14th 2008 from

<http://www.oracle.com/technology/deploy/availability/htdocs/DataGuardRemoteMirroring.html>

This document from Oracle compares Oracle Data Guard replication with host or hardware based remote mirroring solutions. It points out that as data guard only needs to replicate redo log data between sites, it has much better performance and requires less network bandwidth than remote mirroring solutions which must replicate data files and control files as well as redo logs. It provides an example of an internal Oracle system where it was found that remote mirroring requires 7 times more bandwidth and 27 times more I/O operations than a data guard based solution.

Oracle Corporation (n.d.c) Oracle Data Guard and Oracle Streams

Retrieved March 14th 2008 from

<http://www.oracle.com/technology/deploy/availability/htdocs/DataGuardStreams.html>

This Oracle web page discusses the similarities and differences between Oracle data guard and Oracle streams. While both can be used to implement high availability/disaster recovery features, Streams is primarily intended for information sharing and Data Guard is designed primarily for Disaster Recovery.

Oracle Corporation (2008) Oracle Technology global price list

Retrieved March 14th 2008 from <http://www.oracle.com/corporate/pricing/technology-price-list.pdf>

This document contains details of pricing for Oracle technology. Of note is the pricing for systems based on the Sun UltraSparc T1 Processor, where the total number of pre-processor licenses required is determined by multiplying the number of cores by 0.25, which means that an 8 core server only requires 2 (8 cores x 0.25) Oracle pre-processor licenses.

Oracle Corporation (2007a) Oracle Data Guard 11g. The next era in Data protection and availability

Retrieved March 14th 2008 from

http://www.oracle.com/technology/deploy/availability/pdf/twp_dataguard_11gr1.pdf

This white paper from Oracle provides a good overview of the capabilities and features of the latest version of the Oracle Data Guard Software.

Oracle Corporation (2007b) Oracle Clusterware Installation Guide 11g Release 1 (11.1) for Solaris Operating System

Retrieved July 8th 2008 from

http://download.oracle.com/docs/cd/B28359_01/install.111/b28262/presolar.htm#BABJB AEB

This manual from Oracle describes the procedures required to install Oracle clusterware. Section 2.10 covers configuring Secure Shell (SSH) for public key authentication. This procedure

was followed to configure the primary and standby servers to be able to execute remote commands on each other for the purposes of ZFS replication.

Oracle Corporation (2006a) Oracle® Data Guard Concepts and Administration 10g Release 2 (10.2)

Retrieved July 14th 2008 from

http://download.oracle.com/docs/cd/B19306_01/server.102/b14239/toc.htm

This manual from Oracle provides information on configuring and using the Oracle DataGuard product.

Oracle Corporation (2006b) Oracle® Database 2 Day DBA 10g Release 2 (10.2)

Retrieved July 14th 2008 from

http://download.oracle.com/docs/cd/B19306_01/server.102/b14196/toc.htm

This manual from Oracle provides high-level information on administering an Oracle 10g database. It covers the most commonly performed tasks and is designed for new Oracle DBAs.

Oracle Corporation (2005a) The right choice for disaster recovery: Data Guard, Stretch clusters or remote mirroring

Retrieved March 10th 2008 from

www.oracle.com/technology/deploy/availability/pdf/1126_Ray_WP.pdf

This white paper from Oracle describes and compares the various options for replicating a database to a remote location for the purposes of disaster recovery.

Oracle Corporation (2005b) Oracle® Database Installation Guide 10g Release 2 (10.2) for Solaris Operating System (SPARC 64-Bit)

Retrieved July 16th 2008 from

http://download.oracle.com/docs/cd/B19306_01/install.102/b15690/toc.htm

This manual from Oracle describes the procedures required to install the Oracle 10g database software onto a Solaris Sparc 64-bit platform.

Quest Software (2007) SharePlex for Oracle

Retrieved March 16th 2008 from

http://www.quest.com/Quest_Site_Assets/PDF/DSDshareplex12528_1.pdf

This data sheet provides an overview of Quest software's SharePlex for Oracle – a log based database replication solution that operates in a manner similar to Oracle's Data Guard.

Solaris Internals (n.d.) ZFS for databases.

Retrieved August 1st 2008 http://www.solarisinternals.com/wiki/index.php/ZFS_for_Databases

This web page discusses the use of ZFS for hosting database datafiles. It indicates that while ZFS currently has slightly less performance (12%) than UFS with direct I/O, future enhancements should allow it to outperform UFS.

Sun Microsystems (2007a) Maximize IT service Uptime by utilizing dependable Sun SPARC Enterprise T5120 and T5220 Servers.

Retrieved March 12th 2008 from http://www.sun.com/servers/coolthreads/t5220/ras_wp.pdf

This white paper provides an overview of the reliability, availability and serviceability features of their T5110 and T5120 servers. It points out the key features such as reduced component count, redundant and hot swap components that enable this server to approach 99.999% availability levels.

Sun Microsystems (2007b) Solaris ZFS and Veritas Storage Foundation Performance

Retrieved January 10th 2008 from

http://www.sun.com/software/whitepapers/solaris10/zfs_veritas.pdf

This white paper from Sun Microsystems compares the performance of the ZFS filesystem with that of the Veritas filesystem (VxFS) – another popular filesystem used in enterprise class Unix deployments. The paper compares ZFS to VxFS both on ease of use and on raw performance. ZFS is found to be significantly easier to administer and also outperforms VxFS on nearly all benchmarks, having higher throughput and lower CPU utilization. Important factors to consider when implementing a low-cost, easy to administer enterprise system. One area where ZFS underperforms is in the 8k uncached OLTP database test where VxFS outperforms it by a factor of 2.5. This is an important factor to take into account when considering ZFS for use with an Oracle database. Further research is required to determine if ZFS is suitable for OLTP Oracle databases.

Sun Microsystems (2007c) Solaris ZFS and Red Hat Enterprise Linux Ext3 Performance
Retrieved January 10th 2008 from
http://www.sun.com/software/whitepapers/solaris10/zfs_linux.pdf

This white paper from Sun Microsystems compares the performance of the ZFS filesystem with that of the Linux ext3 filesystem – another popular filesystem used in enterprise class Unix deployments. The paper compares ZFS to ext3 both on ease of use and on raw performance. ZFS is found to be significantly easier to administer than ext3 and of comparable or better performance on most tests. Unlike in the ZVS vs. VxFS benchmark document, OLTP performance is not considered.

Sun Microsystems (2007d) Storage quick reference product guide
Retrieved March 17th 2008 from
http://www.sun.com/storagetek/docs/Storage_QRPG.pdf

This document provides a quick reference to the capabilities of all of Sun Microsystems' current storage products, from tape arrays through to disk arrays and storage software.

Sun Microsystems (2007e) Sun Fire enterprise T1000/T2000 server architecture.

Retrieved March 31st 2008 from <http://www.sun.com/servers/coolthreads/t1000-2000-architecture-wp.pdf>

This white paper from Sun Microsystems describes the architecture and features of tier UltraSparc T1 based T1000 and T2000 servers which offer high performance, low cost and low power consumption.

Sun Microsystems (2007f) Sun StorageTek 2540 Array data sheet

Retrieved March 31st 2008 from
http://www.sun.com/storagetek/disk_systems/workgroup/2540/datasheet.pdf

This data sheet provides details on the entry-level Sun StorageTek 2540 disk array.

Sun Microsystems (2007g) Sun Microsystems Expands Leading Storage Portfolio with New World-Record Setting Modular Disk Low-Cost Array (LCA)

Retrieved March 31st 2008 from <http://www.sun.com/aboutsun/pr/2007-04/sunflash.20070417.2.xml>

This white paper from Sun presents performance and price performance benchmark tests for the various disk arrays in their StorageTek disk array portfolio. The tests are based on independent benchmarks designed by the storage processing council (SPC) to provide comparative test data for storage systems. The benchmark results indicate that Sun's entry-level 2540 disk array is capable of similar performance to the mid-range 6140 disk array - 735MB/s vs. 790MB/s in the SPC-2 benchmark, for significantly less cost - \$45.91 vs. \$67.82 \$/SPC2-MB/s.

Sun Microsystems (2007h) Sun StorageTek 6140 Array data sheet

Retrieved March 31st 2008 from http://www.sun.com/storagetek/disk_systems/midrange/6140/datasheet.pdf

This data sheet provides details on the mid-range Sun StorageTek 6140 disk array.

Sun Microsystems (2008a) Sun StorageTek Availability Suite Software

Retrieved April 13th 2008 from http://www.sun.com/storagetek/management_software/data_protection/availability/features.xml

This web page from Sun provides an overview of the features and functionality of the Sun StorageTek availability suite, which provides host-based remote mirroring and point in time copy features for any disk storage attached to a server running Solaris on the SPARC or x86 platforms.

Sun Microsystems (2008b) Solaris ZFS administration guide

Retrieved March 31st 2008 from <http://dlc.sun.com/pdf/819-5461/819-5461.pdf>

This manual from Sun Microsystems provides background information and technical details on the new ZFS filesystem included with Solaris 10. This new filesystem offers impressive scalability, reliability and manageability features along with the ability to take and replicate snapshots of filesystem data at a particular point in time.

Symantec Corporation (2006) Veritas Volume replicator option

Retrieved April 13th 2008 from
http://eval.symantec.com/mktginfo/products/White_Papers/Storage_Server_Management/sf_vvr_wp.pdf

This document from Symantec provides an introduction to the features and capabilities provided by their Veritas Volume Replicator software – an add on option the Veritas Storage Foundation suite which contains the Veritas Filesystem (VxFS) and volume manager (VxVM). It provides host based synchronous and asynchronous replication of VxVM volumes over an IP network.