Oracle Maximum
Availability Architecture

An Oracle White Paper
October 2013

Siebel MAA

with Case Study on Exalogic and Exadata

ORACLE

1 Executive Overview

Oracle Maximum Availability Architecture (MAA) is Oracle's best practices blueprint based on proven Oracle high availability technologies and recommendations. The goal of MAA is to achieve the optimal high availability architecture at the lowest cost and complexity. Papers are published on the Oracle Technology Network (OTN) - http://www.oracle.com/goto/maa.

Siebel Maximum Availability Architecture is a best practice blueprint for achieving the optimal Siebel high availability deployment using Oracle high availability technologies and recommendations.

In this paper we describe:

- The Siebel MAA architecture along with installation, configuration and operational best practices.
- How Siebel MAA was implemented on Oracle Exalogic, Exadata, and F5 Networks BIG-IP LTM machines.
- Our tests to validate our best practices and measure downtime in various outage scenarios.

When Siebel was configured with our MAA best practices on Exalogic and Exadata, we demonstrated that there was minimal user impact during typical failure scenarios. In the event of a total site failure the disaster recovery site could be brought on line in as little as 77 seconds.

2 Introduction

This paper is organized into the following sections:

- A high-level introduction to Oracle Exalogic and Oracle Exadata Database Machines
- Siebel Maximum Availability Architecture – a high level description of the architecture and key technology components
- Siebel MAA Case Study on Exalogic and Exadata – how the MAA architecture was established on our target machines
- Outage Testing and Results – how the system behaved during various outages, and the impact on Siebel users
- Summary of Best Practices - a checklist of the best practices recommended by this paper
- References – a summary of the external documents referenced by this paper
- Appendix – Case Study State Transitions - detailed steps performed during the case study setup and outage testing

2.1 Introduction to Engineered Systems

Oracle's Engineered Systems combine best-of-breed hardware and software components with game-changing technical innovations. Designed, engineered, and tested to work best together, Oracle's Engineered Systems can power the cloud or streamline data center operations to make traditional deployments even more efficient. The components of Oracle's Engineered Systems are preassembled for targeted functionality and then—as a complete system—optimized for extreme performance. By taking the guesswork out of these highly available, purpose-built solutions, Oracle delivers a solution that is integrated across every layer of the technology stack—a simplicity that translates into less risk and lower costs for your business. Only Oracle can innovate and optimize at every layer of the stack to simplify data center operations, drive down costs, and accelerate business innovation.

2.1.1 Oracle Exalogic

Oracle Exalogic is an engineered system on which enterprises deploy Oracle business applications, Oracle Fusion Middleware or third-party software products. Exalogic comes pre-built with compute nodes, memory, flash storage and centralized storage, all connected using InfiniBand in a highly available architecture with fault tolerance and zero-down-time maintenance.

2.1.2 Oracle Exadata Database Machine

Oracle's Exadata Database Machine is Oracle's database platform delivering extreme performance for database applications including Online Transaction Processing, Data Warehousing, Reporting, Batch Processing, or Consolidation of mixed database workloads. Exadata is a pre-configured, pre-tuned, and pre-tested integrated system of servers, networking and storage all optimized around the Oracle database.

3 Siebel Maximum Availability Architecture

Siebel Maximum Availability Architecture (MAA) is a Siebel high availability architecture layered on top of the Oracle Database Maximum Availability Architecture, including a secondary site to provide business continuity in the event of a primary site failure.

In this section we will first present the Oracle Database Maximum Availability Architecture, and then we will describe how to provide high availability for the Siebel application on top of that foundation, resulting in a full Siebel MAA implementation.

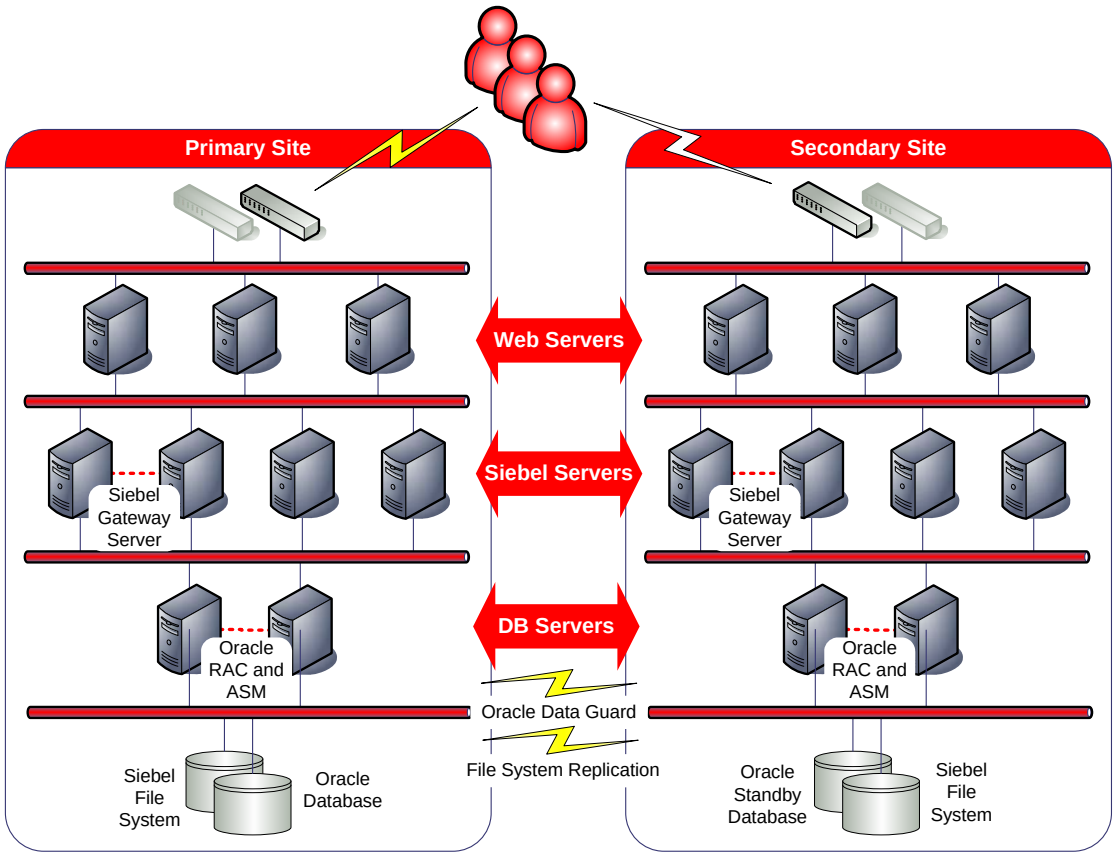

Figure 1 Siebel Maximum Availability Architecture

3.1 Oracle Database Maximum Availability Architecture

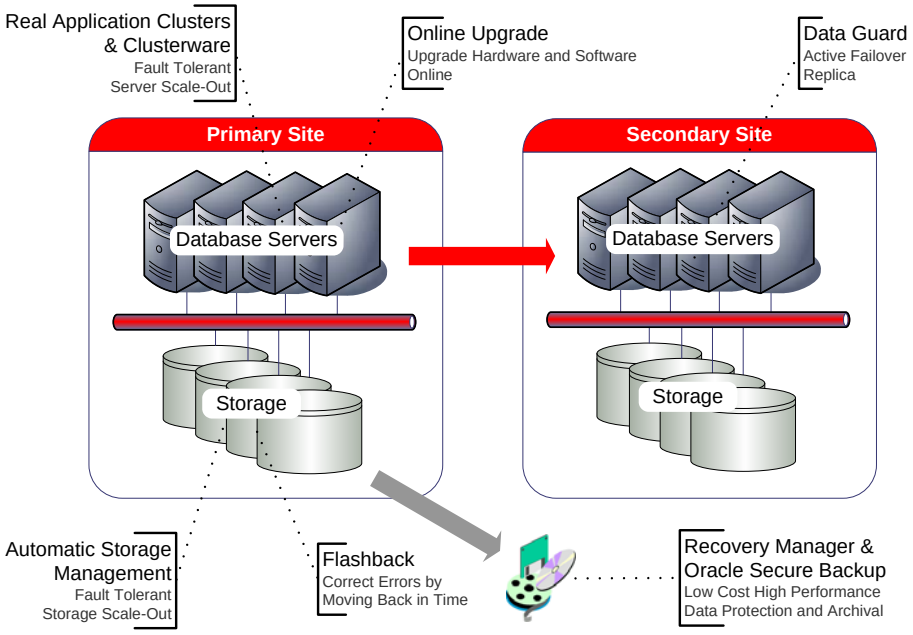

Figure 2 Oracle Database Maximum Availability Architecture

To achieve maximum Siebel application availability, Oracle recommends deploying Siebel on an Oracle Database MAA foundation that includes the following technologies:

- Oracle Real Application Clusters
- Oracle Clusterware
- Oracle Data Guard and Online Upgrade
- Oracle Flashback
- Oracle Automatic Storage Management
- Oracle Recovery Manager and Oracle Secure Backup

We briefly describe each of these technologies in this section. See also: "Oracle Database High Availability Overview" for a thorough introduction to Oracle Database high availability products, features and best practices.

3.1.1 Oracle Real Application Clusters

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

3.1.2 Oracle Clusterware

Oracle Clusterware is a general purpose clustering solution originally designed for the Oracle Real Application Clusters active-active multi-instance database and which has been extended to support clustering of all applications. Oracle Clusterware provides traditional HA failover support in addition to online management of protected resources such as online relocation of applications for planned maintenance. Oracle Clusterware is a policy engine providing a rich dependency model for start and stop dependencies, ordered startup and shutdown of applications and defined placement of resources for affinity, dispersion or exclusion. Oracle Clusterware provides a suite of integrated stand-alone or bundled agents for Oracle Application high availability and application resource management.
MAA best practices recommends Oracle Clusterware be used as the clustering solution for the Siebel mid-tier components and the Siebel Bundled Agent for integrated availability and management. See also: "Oracle Clusterware Administration and Deployment Guide".

3.1.3 Oracle Data Guard and Online Upgrade

Oracle Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable production Oracle databases to survive failures, disasters, user errors, and data corruption. Data Guard maintains these standby databases as transactionally consistent copies of the production database. If the production database becomes unavailable due to a planned or an unplanned outage, Data Guard can switch any standby database to the production role, thus greatly reducing the application downtime caused by the outage. Data Guard can be used with traditional backup, restore, and clustering solutions to provide a high level of data protection and data availability. See also: "Oracle Data Guard Concepts and Administration".

Siebel supports both physical and logical standby databases. A physical standby database provides a physically identical copy of the primary database, with on disk database structures that are identical to the primary database on a block-for-block basis. A physical standby database is kept synchronized with the primary database through Redo Apply, which recovers the redo data received from the primary database and applies the redo to the physical standby database.

With Active Data Guard, a physical standby database can receive and apply redo while it is open for read-only access and so may be used for other purposes as well as disaster recovery.

A logical standby database contains the same logical information as the production database, although the physical organization and structure of the data can be different. The logical standby database is kept synchronized with the primary database though SQL Apply, which transforms the data in the redo received from the primary database into SQL statements and then executes the SQL statements

on the standby database. A logical standby database can be used for disaster recovery and reporting requirements, and can also be used to upgrade the database software and apply patch sets while the application is online and with almost no downtime.

With a single command, a physical standby database can be converted into a Snapshot Standby and become an independent database open read-write, ideal for QA and other testing. The Snapshot Standby continues to receive and archive redo data from the primary database while it is open read-write, thus protecting primary data at all times. When testing is complete, a single command will convert the snapshot back into a standby database, and automatically resynchronize it with the primary.

A physical standby database can be used for rolling database upgrades using the SQL Apply (logical standby) process – and return to its function as a physical standby database once the upgrade is complete.

It is possible to deploy a local standby database at the primary site as well as a remote standby at a secondary site. This offers the advantage that a failover to the local standby can be performed while the Siebel Servers continue running - and can be done almost transparently to the end users. It also offers the ability to perform an online database upgrade without the need to switch to another site. We would recommend that both a local and remote standby be deployed for maximum availability.

3.1.4 Oracle Flashback

Oracle Flashback quickly rewinds an Oracle database, table or transaction to a previous time, to correct any problems caused by logical data corruption or user error. It is like a 'rewind button' for your database. Oracle Flashback is also used to quickly return a previously primary database to standby operation after a Data Guard failover, thus eliminating the need to recopy or re-instantiate the entire database from a backup. See MOS ID 565535.1 "Flashback Database Best Practices & Performance", for flashback database best practices.

3.1.5 Oracle Automatic Storage Management

Oracle Automatic Storage Management (ASM) provides a vertically integrated file system and volume manager directly in the Oracle kernel, resulting in:

- Significantly less work to provision database storage
- Higher levels of availability
- Elimination of the expense, installation, and maintenance of specialized storage products
- Unique capabilities for database applications

For optimal performance, ASM spreads files across all available storage. To protect against data loss, ASM extends the concept of SAME (stripe and mirror everything) and adds more flexibility in that it can mirror at the database file level rather than the entire disk level. See also: "Automatic Storage Management".

3.1.6 Oracle Recovery Manager and Oracle Secure Backup

Recovery Manager (RMAN) is an Oracle database utility that can back up, restore, and recover database files. It is a feature of the Oracle database and does not require separate installation. RMAN integrates with sessions running on an Oracle database to perform a range of backup and recovery activities, including maintaining a repository of historical data about backups. See also: "Oracle Recovery Manager"

Oracle Secure Backup (OSB) is a centralized tape backup management solution providing performant, heterogeneous data protection in distributed UNIX, Linux, Windows, and Network Attached Storage (NAS) environments. By protecting file system and Oracle database data, OSB provides a complete tape backup solution for your IT environment. OSB is tightly integrated with Recovery Manager (RMAN) to provide the media management layer for RMAN. See also: "Oracle Secure Backup".

3.1.7 Siebel Database Configuration Best Practices

We recommend that Siebel database is configured with the following best practices:

3.1.7.1 Create Role Based Database Services and Configure for Transparent Application Failover (TAF)

A database service provides a simple named access point to the database and more convenient TAF configuration. A service can physically span multiple instances in an Oracle RAC cluster and can be simply moved from one instance to another. By requiring Siebel to connect only through a service we are able to relocate or reconfigure the service without reconfiguring Siebel.

Role-based database services should be created for each database role and purpose as summarized in the following table:

DATABASE ROLE	PURPOSE
Primary	Production access to primary
Standby	Production offload of queries on standby
Snapshot Standby	Standby site testing

A database service can be created and configured through Enterprise Manager or using the srvctl command line tool. The "SELECT" failover type and "BASIC" failover method are supported by Siebel and should be adequate for most configurations. For example, this is how a service can be created with the srvctl command:

```
srvctl add service -d siebxd -s SIEB_PRIM -r "siebxd1,siebxd2" -e SELECT -m BASIC -P
 BASIC -w 5 -z 24 -j LONG -l PRIMARY -y AUTOMATIC
```

PARAMETER	VALUE	DESCRIPTION
-d	siebxd	The database unique name
-s	SIEB_PRIM	The database service name
-r	"siebxd1,siebxd2"	Preferred database instances that will start the service
-e	SELECT	The TAF failover type
-m	BASIC	The TAF failover method
-P	BASIC	TAF policy
-w	5	The time interval between connection retries [1]
-z	24	The number of connection retries before the connection will fail
-j	LONG	Connection load balancing method (Siebel connections normally last a long time)
-l	PRIMARY	The database role under which this service will be started
-y	AUTOMATIC	Indicates that the service should be started automatically

[1] Changing the –z (failover retries) and -w (failover delay) parameters will have no effect for the Siebel Object Manager components because they will always retry 24 times with a delay of 5 seconds.

3.1.7.2 Configure Hugepages (Linux Database Server Only)

Siebel will typically run with many database connections and a large SGA and so configuring hugepages for the Siebel database instances is essential. It is necessary to manually configure sufficient hugepages for the ASM instance and all database instances on each Linux database server node. This will result in more efficient page table memory usage, which is critically important with a large SGA or when there are high numbers of concurrent database connections. Hugepages can only be used for SGA memory space and so do not configure more than is required.

MOS ID 361468.1, "HugePages on Oracle Linux 64-bit" describes how to configure hugepages. Automatic Shared Memory Management (ASMM) can be used with hugepages and so use the SGA_MAX_SIZE parameter to set the SGA size for each instance.

Automatic Memory Management (AMM) cannot be used in conjunction with hugepages and so the MEMORY_TARGET and MEMORY_MAX_TARGET parameters should be unset for each database instance. See MOS ID 749851.1 "HugePages and Oracle Database 11g Automatic Memory Management (AMM) on Linux" for details.

Set the parameter USE_LARGE_PAGES='only' for each instance so that the instance will only start if sufficient hugepages are available. See MOS ID 1392497.1 "USE_LARGE_PAGES To Enable HugePages" for details.

It may be necessary to reboot the database server to bring the new hugepages system configuration into effect. Check to make sure that you have sufficient hugepages by starting all the database instances at the same time.

Starting with Oracle Database 11.2.0.2, a message is logged to the database alert log when hugepages are being used, for example:

```
****************** Huge Pages Information *****************
Huge Pages memory pool detected (total: 18482 free: 17994)
DFLT Huge Pages allocation successful (allocated: 4609)
***********************************************************
```

In this example, 4609 hugepages were used.

3.1.7.3 Handle Database Password Expiration

The default behavior of Oracle Database has changed in release 11*g* such that database user passwords will expire after 180 days. Processes should be put in place to refresh passwords regularly or expiration should be extended or disabled. Siebel application availability will be impacted if passwords are allowed to expire. Password expiration for the default user profile can be disabled with the following command:

```
alter profile default limit password_life_time unlimited;
```

A non-fatal warning, ORA-28002, will begin to be reported by the database when the password is about to expire (by default 7 days before expiration). The Siebel application will interpret this warning as an error, and sessions will begin to fail, even before the password has expired.

Please refer to the section Configuring Password Protection in chapter 3 of the Oracle Database Security Guide 11g Release 2 (11.2) for more details.

3.1.7.4 Configure Dead Connection Detection

When a Siebel Server node fails suddenly there may not be time for the operating system to reset the TCP connections and as a result the connections on the database server will remain open. To clean up the "dead" connections it is recommended that Dead Connection Detection is configured. See MOS ID 151972.1 "(Dead Connection Detection (DCD) Explained" for details.

Making these configuration changes may have adverse effect on network utilization and so all changes should be tested and monitored carefully.

3.1.7.5 Consider Reducing Timeout on RAC Node Failure (Exadata Only)

On Exadata, it is possible to fail over more quickly in the event of a RAC node failure by reducing the "misscount" parameter. The parameter defines how long to wait after a node becomes unresponsive before evicting the node from the cluster. The parameter has a default of 60 (seconds) on Exadata, and can be reduce to 30 if necessary, but should not be configure less than 30. Reducing the "misscount" parameter will increase the risk that a node is evicted unnecessarily, and so it should only be changed if absolutely necessary. To update the CSS misscount setting, log in as the root user on one of the database servers and run the command:

```
$GRID_HOME/bin/crsctl set css misscount 30
```

3.2 Siebel High Availability Architecture

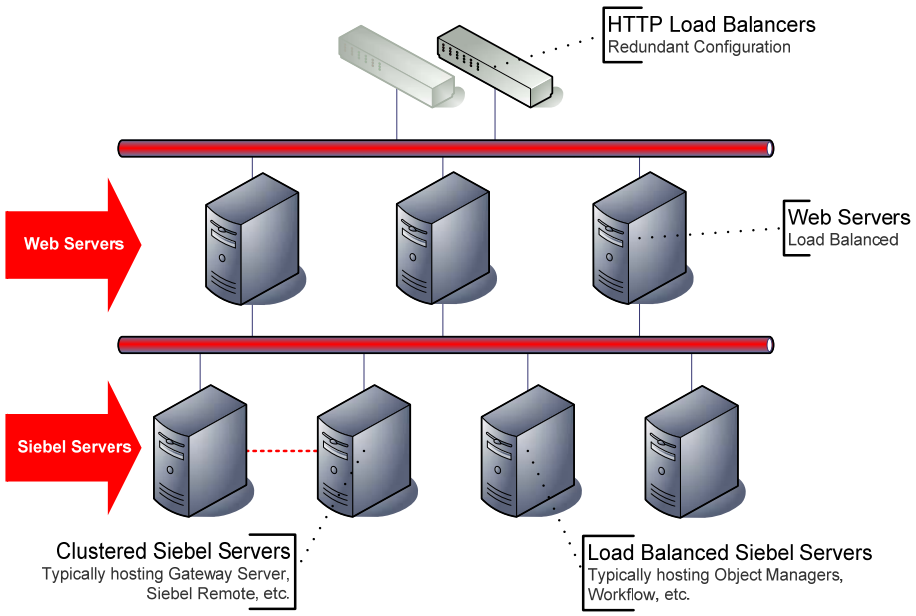

Figure 3 Siebel High Availability Architecture

In this section we discuss the high availability deployment of the Siebel application software that is layered on top of the Database MAA foundation, as well as the HA requirements for Siebel file system.

3.2.1 Siebel Application Software HA Deployment Options

Siebel components can each be deployed in a highly available manner, using one of three options depending on the requirements and constraints of the component being deployed – active/active load balanced, distributed services deployed across multiple servers, and active/passive for singletons. In all cases we recommend more than one instance of each component be deployed at each site, on separate physical servers so a server outage does not affect availability. Where more than one instance of a component can be serving users at a time, we recommend the servers have adequate capacity to run peak load even when one server is down.

The three high availability deployment options for Siebel components are:

Load Balancing

Core Siebel components (e.g., Siebel Server / Object Managers, Web Server) are installed and deployed on multiple servers, and run in an "active/active" configuration for high availability and scalability purposes. Client-initiated workload is distributed across multiple component instances running on multiple servers through load balancing. Web Server load is distributed by an HTTP load balancer. Siebel Server load may be delivered by an HTTP load balancer or by native Siebel load balancing.

Distributed Services

Many Siebel components are implemented as Business Services. Business Services are invoked by other components to complete their business function. In some cases Business Services can be deployed redundantly across multiple Siebel Servers in a configuration known as Distributed Services.

The Siebel Server Request Broker (SRB) balances Service requests across the component instances. In the event that a component instance is lost, the request is re-routed to the surviving instances. An SRB instance will typically be running on all Siebel Servers.

Clustering

Some Siebel services are singletons, meaning only one instance of the service can be running at a time. These are deployed in Siebel Server clusters.

Siebel Server clusters consist of two or more physical servers linked together so that if one server fails, resources such as disks, network addresses, and Siebel components can be switched over to another server. Clustered Siebel components run in an active/passive configuration where a specific Siebel component instance is running on only one physical host at a time. We use Oracle Clusterware (or other 3rd party cluster manager) to monitor and manage the configuration to ensure the components are enabled on only one node of a hardware cluster at a time.

Not all deployment options are supported by all components. The following table gives an example of the supported and preferred options for some of the most commonly deployed components. The "Siebel Deployment Planning Guide" has a comprehensive list.

Table 1 High Availability Options for Various Siebel Components

COMPONENT	CLUSTERING	LOAD BALANCING	DISTRIBUTED SERVICES
Object Manager	Supported	Preferred	
EAI Object Manager	Supported	Preferred	
Siebel Remote	Preferred		
Workflow Process Manager	Supported		Preferred
Siebel Web Server	Supported	Preferred	
Siebel Gateway Server	Preferred		

3.2.2 Load Balancing Deployment

Web Servers and many Siebel Server Components can be load balanced.

- A third-party load balancer is required to balance Web Server load.
- Siebel native (software) or third party load balancing may be used to balance Siebel Server load.

The load balancer monitors the servers and improves availability by routing traffic appropriately when outages occur. Third party load balancers should be deployed in a redundant configuration. Please refer to your load balancer documentation for specific details on how to configure Siebel load balancing.

3.2.2.1 Optimal Siebel Web Server Load Balancing

It has been found that optimal web server load balancing is achieved when the following logic is implemented in the load balancer:

- Load balancing is performed on every request in a round robin fashion with no persistence. This ensures a balanced load across all web servers, smooth recovery when a web server goes down, and almost instant rebalancing when a web server comes up. It is possible to do this with Siebel because session state is not maintained in the Web servers.
- A web server is marked down if attempts to get a static page from the server failed for a period of 16 seconds, checking every 5 seconds. This has been found to be a reliable test of web server availability because it only requires the web server itself to be available to respond to the request and it is unlikely for the web server to be available but not be able to respond within 16 seconds. The load balancer continues to monitor the downed web server and when it begins to respond it is marked up and requests are once more routed to it.
- If the web server is down then traffic is not routed to that web server, and all existing connections to that server are dropped. If all web servers are marked down then all connections are rejected.
- If, after load balancing, the connection to a web server fails, the other web servers were tried, and only if connections to all web servers fail should an error be returned to the client. A connection failure is most likely due to a web server failure and a precursor to marking the web server down, but this logic prevents many connection errors in the meantime.

3.2.3 Siebel Cluster Deployment

To create a Siebel cluster you need a cluster manager and a shared Siebel software home.

Cluster Manager:

- Supports service virtual IP management with failover. The virtual IP address is used as a single network address for the Siebel Server or Gateway Server independent of the physical service location
- Performs service monitoring so it will know when services fail.
- Will restart and relocate Siebel services in the event of failure.

Shared Siebel Software Home:

- Shared by all cluster nodes for failover.
- Contains Siebel software, name server backing file, remote docking folders, etc.
- Must be deployed in a HA configuration to avoid a single point of failure. Typically, a cluster file system or NFS solution would be used.

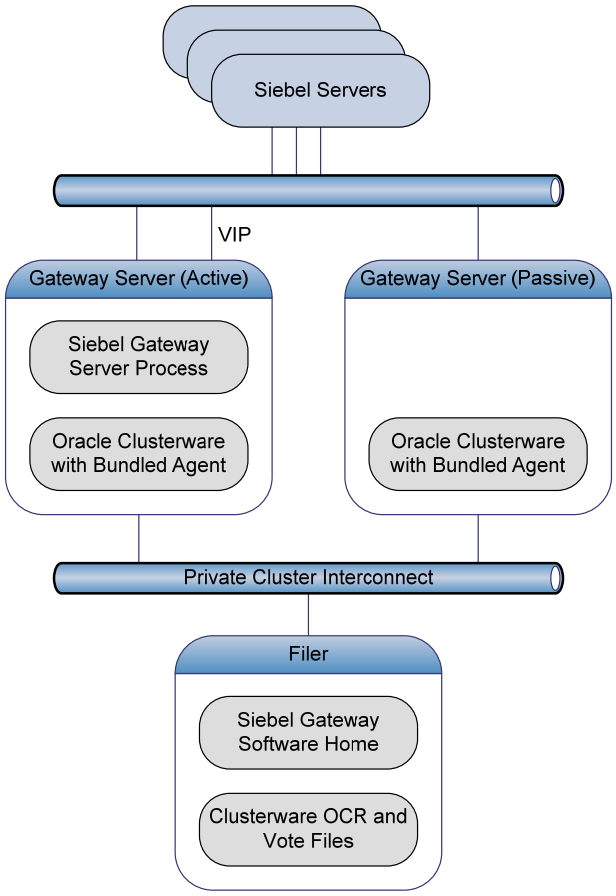

Figure 4 Siebel Gateway Server Deployed with Oracle Clusterware and Bundled Agents

Oracle Clusterware may be used as the Cluster Manager for protecting Siebel components and bundled agents are available that are specifically design to manage and monitor the Siebel Gateway Server and Siebel Servers. For more details see http://www.oracle.com/technetwork/products/clusterware/overview/index.html.

3.2.4 Siebel File System Deployment

The Siebel file system is used to store file attachments and other documents in the Siebel application, and is accessed in parallel by all Siebel Servers. It is a critical part of the Siebel application and so must be deployed in a highly available configuration. Typically, this would be achieved through a cluster file system or network file system (NFS).

The contents of the Siebel File System must be continuously replicated to the secondary site so that the data is available in the event of a primary site failure.

3.2.5 Siebel Tier Configuration Best Practices

We recommend the following when configuring the Siebel application:

3.2.5.1 Database Connection Configuration

When configuring the Siebel application connection to the database it is recommended that the following requirements are met:

- In order for Siebel to take advantage of TAF it must be configured to connect to the database service as recommended in section 3.1.7.1 Create Role Based Database Services and Configure for Transparent Application Failover (TAF). This is achieved through the SERVICE_NAME parameter in the TNS alias configuration.
- The client must be configured to try all database listeners in the RAC cluster so that a new connection can be established even when nodes are down.
  - If Siebel is configured with an Oracle Database Client at 11g Release 2 or later, and the SCAN feature is configured on the database, then a single SCAN address can be configured, for example:
    `(ADDRESS=(PROTOCOL=TCP)(HOST=test-scan)(PORT=1521))`
  - Otherwise, each database VIP address should be configured, for example:
    `(ADDRESS=(PROTOCOL=TCP)(HOST=test_vip1)(PORT=1521))`
    `(ADDRESS=(PROTOCOL=TCP)(HOST=test_vip2)(PORT=1521)))`
- The client must be configured to timeout if the connection to a listener is taking too long.
  - If Siebel is configured with an Oracle Database Client at 11g Release 2 or later, the CONNECT_TIMEOUT parameter can be used in the TNS alias configuration, for example:
    `(CONNECT_TIMEOUT=3)`
  - Otherwise, the OUTBOUND_CONNECT_TIMEOUT parameter may be configured in the client side SQLNET.ORA file, for example:
    `SQLNET.OUTBOUND_CONNECT_TIMEOUT=3`
- If Siebel is configured with an Oracle Database Client at 11g Release 2 or later, the RETRY_COUNT parameter may be used to keep retrying the connection, for example:
  `(RETRY_COUNT=3)`

Here is a complete example with an Oracle Database Client 11g Release 2 and SCAN:

```
SIEBEL = (DESCRIPTION =
    (CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
    (ADDRESS=(PROTOCOL=TCP)(HOST=test-scan)(PORT=1521))
    (CONNECT_DATA= (SERVICE_NAME=SIEB_PRIM))
  )
```

The configuration changes must be the same on all Siebel servers.

3.2.5.2 Reduce TCP Keepalive Timeout

It is possible for some database connections to hang on the Siebel Server if a database node fails, and for TCP connections to hang on the Web Server if the Siebel Server node fails. This is only in the rare case where the node crashes or network fails before the TCP connections can be cleaned up by the operating system. To clean up the "dead" connections it is recommended to reduce the TCP Keepalive Timeout parameters for Siebel Servers, Gateway Servers and Web Servers. Please refer to MOS ID 249213.1 – "Performance problems with Failover when TCP Network goes down (no IP address)" for details on how to configure the TCP Keepalive timeout.

Making these configuration changes may have adverse effect on network utilization and so all changes should be tested and monitored carefully.

3.3 Siebel MAA Site State Model and State Transitions

In Figure 5 Siebel MAA Site State Model and State Transitions, we picture the states that a deployment goes through as it progresses from the initial single site implementation through the setup, testing and an eventual dual site MAA deployment. The systems will have a specific configuration in each state and there is a set of steps that must be performed to move from one state to the next.

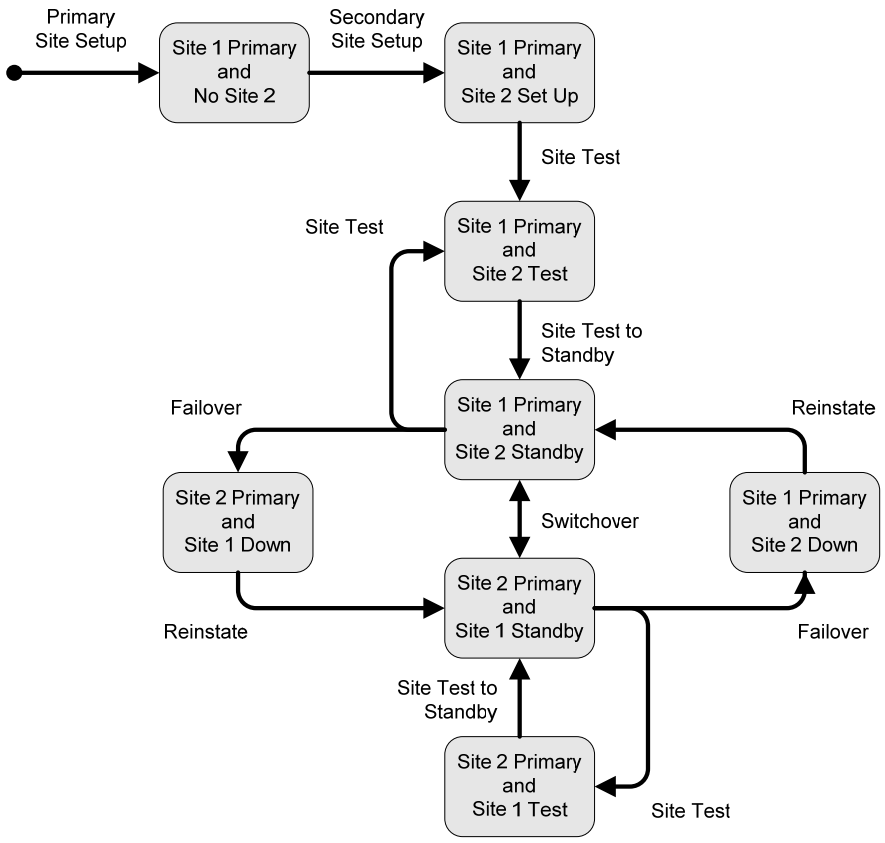

Figure 5 Siebel MAA Site State Model and State Transitions

A summary description of the state transitions is provided in the following table:

Table 2 Description of State Transitions

TRANSITION	DESCRIPTION
Primary Site Setup	Install and configure the primary site.
Secondary Site Setup	Establish the secondary site.
Site Test	Prepare the standby site for a site test.
Site Test to Standby	Convert the site performing a site test back to standby mode.
Switchover	Switch the roles so that the current standby becomes the primary and the current primary becomes the standby.
Failover	Switch the current standby to primary mode. The current primary is assumed to be down or unavailable.
Reinstate Standby	Reinstate the old primary as a standby after failover.

The following table summarizes how the database and Siebel File System are configured in each state:

Table 3 Siebel Database and Siebel File System Configuration in Each Site State

SITE STATE	SIEBEL DATABASE - DATA GUARD	SIEBEL FILE SYSTEM - REPLICATION
Site 1 Primary and No Site 2	Not configured	Not configured
Site 1 Primary and Site 2 Set Up	Site 1 primary and site 2 physical standby. Snapshot standby during setup.	Site 1 primary with continuous replication to site 2. Site 2 clone during setup.
Site 1 Primary and Site 2 Test	Site 1 primary and site 2 snapshot standby.	Site 1 primary with continuous replication to site 2. Site 2 clone created for test.
Site 1 Primary and Site 2 Standby	Site 1 primary and site 2 physical standby.	Site 1 primary with continuous replication to site 2.
Site 2 Primary and Site 1 Down	Site 2 primary through failover, and site 1 down.	Site 2 primary established from replica, and site 1 down.
Site 2 Primary and Site 1 Standby	Site 2 primary and site 1 physical standby.	Site 2 primary and continuous replication to site 1.
Site 1Primary and Site 2 Down	Site 1 primary through failover and site 2 down.	Site 1 primary established from replica and site 2 down.
Site 2 Primary and Site 1 Test	Site 2 primary and site 1 snapshot standby.	Site 2 primary with continuous replication to site 1. Site 1 clone created for test.

3.4 Planned and Unplanned Outage Solutions

In the following sections we summarize the outages that may occur in a Siebel environment and the Oracle solution that would be used to minimize application downtime. In all cases, we are focused on Siebel Application downtime as perceived by the end user, not the downtime of the individual component.

3.4.1 Unplanned Outage Solutions

In the following table we describe the unplanned outages that may be caused by system or human failures in a Siebel environment and the technology solutions that would be used to recover and keep downtime to a minimum.

Table 4 Unplanned Outage Solutions

OUTAGE TYPE	ORACLE SOLUTION	BENEFITS	RECOVERY TIME
Siebel Server Node or Component Failure	Load Balancing	Surviving nodes pick up the slack	Affected users reconnect
	Distributed Services	Surviving nodes continue processing	No downtime
	Clustering	Automatic failover to surviving node	Seconds to < 2 minutes
Database Node or Instance Failure	RAC	Automatic recovery of failed nodes and instances, transparent application and service failover	Users transparently fail over Updates may need to be re-submitted
Site Failure	Data Guard	Fast Start Failover	< 2 minutes [2]
Storage Failure	ASM	Mirroring and automatic rebalance	No downtime
	RMAN with flash recovery area	Fully managed database recovery and disk based backups	Minutes to hours
	Data Guard	Fast Start Failover	< 2 minutes
Human Error	Oracle Flashback	Database and fine grained rewind capability	Minutes
	Log Miner	Log analysis	Minutes to hours
Data Corruption	RMAN with flash recovery area	Online block media recovery and managed disk-based backups	Minutes to hours
	Data Guard	Automatic validation of redo blocks before they are applied, fast failover to an uncorrupted standby database	Seconds to 5 minutes

3.4.2

[2] Site failure will require Siebel Remote re-extract ion

Planned Maintenance Solutions

In the following table we summarize the planned maintenance activities that may typically occur in a Siebel environment and the technology solutions that we would recommend to keep downtime to a minimum.

Table 5 Planned Outage Solutions

MAINTENANCE ACTIVITY	SOLUTION	SIEBEL OUTAGE
Mid-Tier Operating System or Hardware Upgrade	Siebel Load balancing, distributed services and clustering	No downtime
Siebel Application Patching	Siebel rolling patch application	No downtime
Siebel Application Configuration Change	Siebel Application Restart	Minutes
Siebel Upgrades	Siebel Upgrade and Upgrade Tuner	Hours to days (depending on DB size) [3]
Database Tier Operating System or Hardware Upgrade	Oracle RAC	No downtime
Oracle Database interim patching	Oracle RAC rolling apply	No downtime
Oracle Database online patching	Online Patching	No downtime
Grid Infrastructure upgrade and patches	Rolling apply/upgrade	No downtime
Database storage migration	Oracle ASM	No downtime
Migrating to ASM or migrating a single-instance database to Oracle RAC	Oracle Data Guard	Seconds to minutes
Database patch set and upgrades	Oracle Data Guard logical standby	Seconds to minutes

[3] In reality there are a number of ways to mitigate the impact of extended upgrade downtime, for example, by providing a read-only replica. Oracle Consulting Services have significant experience in this area and can help to plan and execute the upgrade.

4 Siebel MAA Case Study on Exalogic and Exadata

In this section we describe how the Siebel Maximum Availability Architecture described in Chapter 3 was deployed on a system consisting of two X3-2 Exalogic machines, two X3-2 Exadata machines, and two F5 4200v Local Traffic Managers. The high level view of the configured system is pictured in Figure 6 Case Study System Configuration:

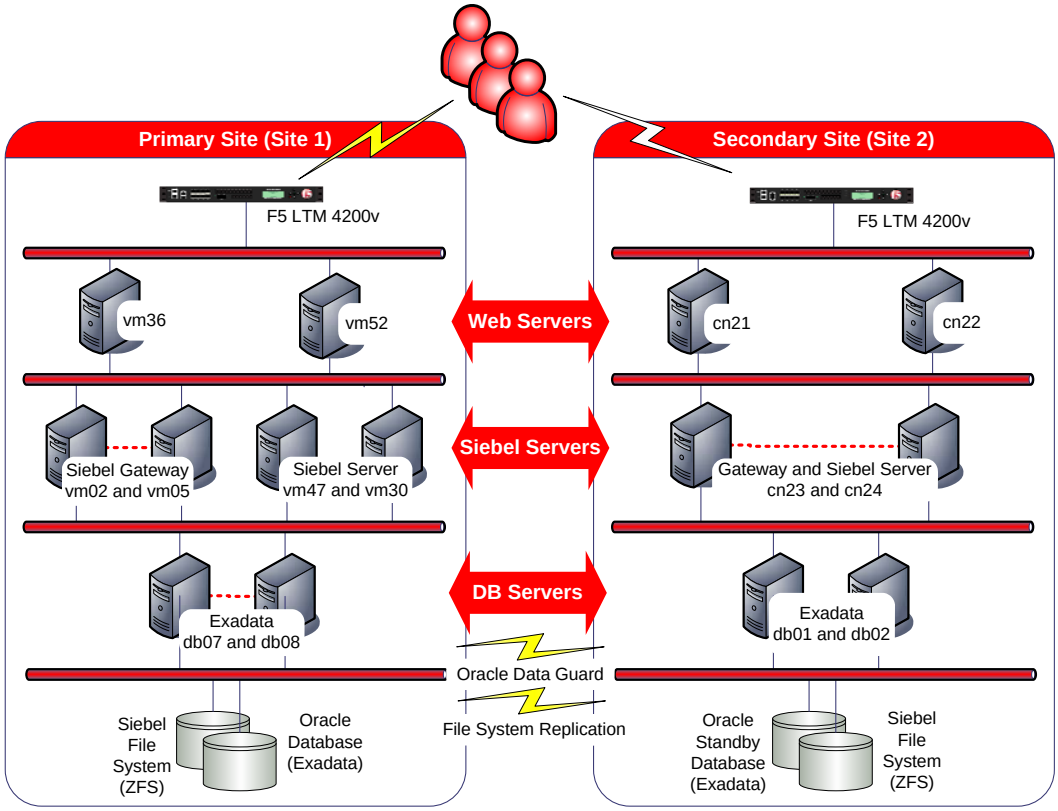

Figure 6 Case Study System Configuration

4.1 Systems

The following systems were used for the case study:

SYSTEM	SITE 1 MODEL	SITE 2 MODEL
F5 LTM	F5 LTM 4200v	F5 LTM 4200v
Exalogic	X3-2 Quarter Rack (8 virtualized nodes)	X3-2 Quarter Rack (4 bare metal nodes)
Exadata	X3-2 Quarter Rack	X3-2 Quarter Rack

4.2 Software

The following software was used for the case study:

SOFTWARE	VERSION
Exalogic	2.0.4.0.0 (Site 1) 2.0.3.0.0 (Site 2)
Siebel	8.1.1.9
ZFSSA	2011.04.24.5.0,1-1.33 (Site 1) 2011.04.24.6.0,1-1.36 (Site 2)
Exadata Database	11.2.0.3 Exadata Bundle Patch 20
Exadata Grid Infrastructure	11.2.0.3 Exadata Bundle Patch 20
Database Client	11.2.0.3
Oracle Clusterware	11.2.0.3 [4]
Oracle Clusterware Bundled Agents	2.1
F5 BIG-IP Local Traffic Manager	11.3

4.3 State Transitions

The detailed steps that were followed to setup and test the system are documented in Section 8 Appendix – Case Study State Transitions:

- 8.1 Primary Site Setup
- 8.2 Secondary Site Setup
- 8.3 Site Test
- 8.4 Site Test to Standby
- 8.5 Switchover
- 8.6 Failover
- 8.7 Reinstate Standby

[4] Oracle Clusterware was used to create a Siebel Gateway Server cluster in this case study. It may also be used for certain Siebel Server components, such as Siebel Remote.

4.4 Administrative Roles

The three administrative roles that were established to manage the system are summarized in the following table:

OS USER	OS GROUPS	ROLE
siebel	oinstall	Siebel application administrator
oracle_siebel	oinstall, siebel_dba	Siebel database administrator
grid	oinstall, dba	Clusterware and ASM administrator

The users and groups were registered in NIS to facilitate the NFS v4 security model.

4.5 Exalogic

The Siebel application servers were all deployed on Exalogic. Site 1 was configured with virtualized Exalogic, and site 2 had a bare metal configuration.

The Siebel File System and all other shared file systems were hosted on the highly available ZFS Storage Appliances located in the Exalogic machines. All Exalogic specific NFS best practices were respected, for example, NFS delegation was disabled.

The configuration of the ZFS shares for appropriate data security was out of scope for this paper. This may add additional steps to configuration and state transations.

4.5.1 Application Deployment

The following application components were deployed for the case study:

SYSTEM	SITE 1	SITE 2
Siebel Web Server	vm36 and vm52 (dedicated Exalogic vServers)	cn21 and cn22 (bare metal Exalogic Nodes)
Siebel Server	vm30 and vm47 (dedicated Exalogic vServers)	cn23 and cn24 (bare metal Exalogic Nodes)
Siebel Gateway Server	vm02 and vm05 (dedicated Exalogic vServers)	cn23 and cn24 (bare metal Exalogic Nodes)

4.6 Exadata

The Siebel database was hosted on Exadata on site 1 and site 2. The Exadata machines had a standard configuration.

4.6.1 Database Services

Database services were created on the primary and standby to provide access in the primary, standby and snapshot standby modes:

SERVICE NAME	DATABASE ROLE	PURPOSE
SIEB_PRIM	Primary	Production
SIEB_STBY	Standby	Production offload of read-only workloads
SIEB_STBY_TEST	Snapshot Standby	Standby site testing

4.6.2 Exadata Exachk Best Practices

Exachk was run after the database was created to validate the configuration and the following exceptions to the generic Exadata best practices were identified:

- The database PROCESSES parameter was set higher than the recommendation. This was because Siebel will typically have many concurrent users and, even with Siebel connection pooling, this can result in many database connections. In general, only a small percentage of Siebel sessions are active at any point in time and so a larger number of connections can be maintained concurrently, however, there must be enough memory available to support the peak number of connections. Each connection results in an additional Oracle shadow process to be spawned on the Database Server. In our tests we found that each shadow process consumed approximately 7MB of memory. It is also important to allow for the additional connections and memory requirements in the event of a database server outage.

- The MOS ID 781927.1 "Performance Tuning Guidelines for Siebel CRM Application on Oracle Database" recommends several hidden database initialization parameters that may be flagged by Exachk. The following parameter settings as recommended by this document were applied:

PARAMETER	VALUE
optimizer_features_enable	<unset> [5]
optimizer_index_caching	0
optimizer_mode	all_rows
query_rewrite_integrity	enforced
star_transformation_enabled	false
optimizer_index_cost_adj	1
optimizer_dynamic_sampling	1
query_rewrite_enabled	false
pga_aggregate_target	12G [6]
statistics_level	typical
_always_semi_join	off
_b_tree_bimap_plans	false
_partition_view_enabled	false
_gc_defer_time	0
_no_or_expansion	false
_optmizer_max_permutations	100

Note, the memory_target and memory_max_target parameters recommended by the document are incompatible with hugepages and so were not set.

[5] The parameter optimizer_features_enable defaulted to the current version 11.2.0.3 when unset.

[6] The parameter pga_aggregate_target was originally estimated as 7MB per connections. In practice, the V$PGA_STAT view should be monitored to check true usage (maximum PGA allocated) and the parameter adjusted accordingly. Note this view can also be used to monitor the maximum number of processes (max processes count).

4.7 F5 Networks BIG-IP Local Traffic Manager

F5 Networks BIG-IP Local Traffic Manager (LTM) application delivery controllers were used for Siebel Web Server load balancing on the primary and secondary sites. The BIG-IP LTM provides application health monitoring, TCP connection management, load balancing, and high availability to the Siebel Web tier.

4.7.1 F5 Web Server Load Balancing Configuration

The base configuration of the F5 BIG-IP was done in the accordance with the existing "F5 Siebel 8.0 Deployment Guide", and then modified based on test results to produce a more optimal configuration. The changes to this base configuration, and the reselect iRule, are detailed in Section 8.1.10

Web Server Load Balancing Configuration. In particular:

- An advanced iRule was developed to implement the optimized logic stated in Section 3.2.2 Load Balancing Deployment. It allows the seamless reselection of another Web server if a Web server is not responding, even if it has not yet failed the periodic health checks and been marked as down. This iRule allows the LTM to determine in real-time if a selected Siebel Web server is not responding, and to quickly choose a Web server that is responding. In the absence of this iRule, existing users connecting to a failed Web Server would be presented with an error message, and be forced to retry.
- The OneConnect feature was also used to reduce server side TCP connections.

For more information on F5 Networks BIG-IP family of application delivery controllers, see "BIG-IP Product Suite". For more information on iRules, see "F5 DevCentral".

4.8 Test Workload

Oracle Application Testing Suite was used to drive a Siebel application workload during outage tests. The behavior of this workload was monitored during the outages and any anomalies were noted.

The workload was comprised of three complex user journeys executed in the Siebel Financial Services Call Center. 600 virtual users were executing each journey in a loop, making a total of 1800 concurrent virtual users. The details of each user journey were as follows:

4.8.1 Incoming Call Updates Service Request

SEQUENCE	TRANSACTION
1	Click on Service Screen Tab – Go to My Service Request
2	Drill down on Service Request and go to SR Activity
3	Navigate to SR – Related SR
4	Select, Add Solution and Save
5	Update Service Request, set it to “Pending” and Save
6	Navigate back to Service Request Activity

4.8.2 Incoming Call Creates Opportunity, Quote and Order

SEQUENCE	TRANSACTION
1	Create a new contact
2	Create a new Opportunity for that contact
3	Add two products to Opportunity
4	Navigate to Opportunities - Quotes View
5	Click "AutoQuote" button to generate quote
6	Enter Quote Name, and Price List
7	Drilldown on the quote name to go to Quote - Line Items View and specify discount
8	Click "Reprice All" button
9	Update opportunity
10	Navigate to Quotes - Orders View
11	Click on "AutoOrder" button to automatically generate order
12	Navigate back to Opportunity

4.8.3 Incoming Call Creates Service Request

SEQUENCE	TRANSACTION
1	Create a new Service Request
2	Associate Contact and Account for that Service Request
3	Click “Verify” button to bring up pick applet
4	Select “Entitlement”
5	Query and Select Policy
6	Select Product and add Product to Service Request
7	Save Service Request
8	Go to Service Request Activity Plan
9	Select Activity Plan and Save Service Request

5 Outage Testing and Results

5.1 Unplanned Outage Testing Procedure

5.1.1 Siebel Web Server Outages

The workload was ramped up and the following outages were simulated:

- **Instance** - A Siebel Web Server was stopped abruptly using the command "opmnctl stopall".
- **Node** - In the virtualized case, a vServer was stopped abruptly using the command "xm destroy" issued from the VM Server (DOM0). In the bare metal case, a server was stopped abruptly using a power reset from the ILOM.

5.1.2 Siebel Server Outages

The workload was ramped up and the following outages were simulated:

- **Node** - In the virtualized case, a vServer was stopped abruptly using the command "xm destroy" issued from the VM Server (DOM0). In the bare metal case, a server node was stopped abruptly using a power reset from the ILOM.

5.1.3 Gateway Server Outages

The workload was ramped up and the following outages were simulated:

- **Instance** - A Siebel Gateway Server was stopped abruptly using the command "kill -9" on the Gateway Server process.
- **Node** - In the virtualized case, a vServer was stopped abruptly using the command "xm destroy" issued from the VM Server (DOM0). In the bare metal case, a server node was stopped abruptly using a power reset from the ILOM.

5.1.4 Database Server Outages

The workload was ramped up and the following outages were simulated:

- **Instance** - A database instance was stopped abruptly using the command "shutdown abort" from the sqlplus tool.
- **Node –** A database server node was stopped abruptly using a power reset from the ILOM.

5.1.5 Site Outages

The workload was ramped up and the following outages were simulated:

- **Site –** All servers on the primary site were stopped abruptly and site failover was performed.

5.2 Unplanned Outage Test Results

Table 6 Summary of Unplanned Outage Results

COMPONENT	OUTAGE TYPE	OUTAGE (SECONDS)	USER ERRORS	USER FAILURES	TIME TO FAILOVER (SECONDS)
Web Server (0)	Instance	0	0.3%	0.0%	
	Node	25 (partial)	0.3%	0.0%	
Siebel Server (0)	Node	0	50.0%	50.0%	
Gateway Server (5.2.3)	Instance	0	0.0%	0.0%	30
	Node	0	0.0%	0.0%	100
Database Server (0)	Instance	10 (partial)	1.9%	0.0%	
	Node	45 (complete)	2.1%	0.0%	
Site (0)	Unplanned	77 (complete)	100.0%	100.0%	77

Key:

- **Outage** – Time for monitored service performance to return to normal, or 50% of normal in the case of Siebel Server Node outage.
- **User Errors** – Percentage of users who experienced an error message during the outage.
- **User Failures** – Percentage of users who were experienced a total application failure during the outage.
- **Time to Failover** – In the case of the Gateway Server the time to detect the failure and start the service on a different node. In the case of site failure, the time taken to perform complete site fail over and for the Siebel application service to be restored to normal operation.

The following sections provide observations on the tests and results.

5.2.1 Observations on Web Server Outages

There was no perceptible performance impact of Web Server instance failure.

There was a performance impact during node failure as can be seen in the following graph. This was primarily due to the time taken by the F5 monitor to confirm that the node was down (configured at 16 seconds) and to break existing connections. There is an option to reduce the F5 monitor period, but this would increase the risk of marking down a sluggish web server.

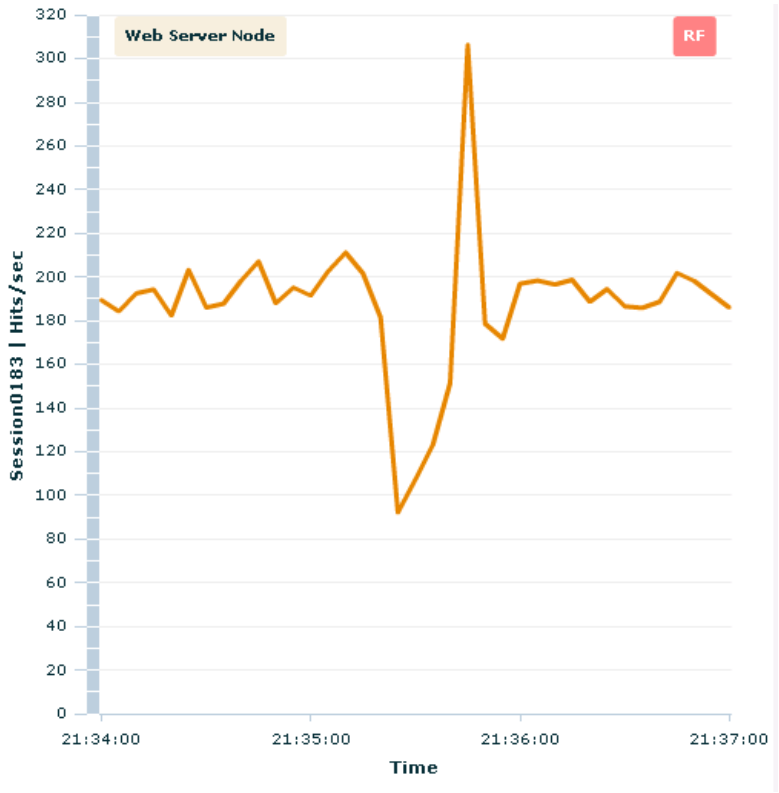

Figure 7 Performance Impact of Web Server Node Failure

Users that had requests in-flight at the time of the failure experienced an error message, but they were able to resubmit their work and continue.

5.2.2 Observations on Siebel Server Outage

The Siebel Server holds state for each Siebel session and so user failures are unavoidable when there is a Siebel Server failure. Users that fail are prompted with an error and may log back in and continue processing on a surviving Siebel Server. Our tests scripts were not coded to reconnect on failure and so throughput was expected to drop by half after the failure.

The following graph shows how performance remained normal for the surviving users and there was no interruption of service for them.

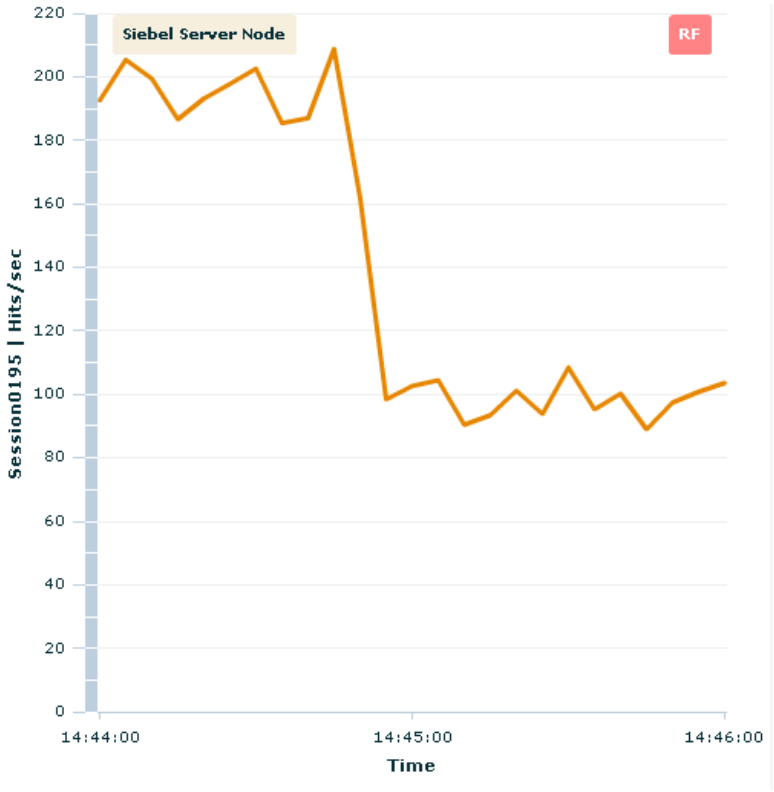

Figure 8 Performance Impact of Siebel Server Node Failure

5.2.3 Observations on Gateway Server Outages

There was no perceptible virtual user impact during all Gateway Server outages.

5.2.4 Observations on Database Server Outages

One step in the test workload involved the invocation of a transactional Siebel Workflow that was not recoverable after failover and this resulted in an error for these users. The users were able to retry the request successfully and continue processing.

When a database instance failed, connections on that instance failed over to the surviving node and Siebel was able to continue processing. The following graph shows that there was only a very slight and short reduction in performance after the outage.

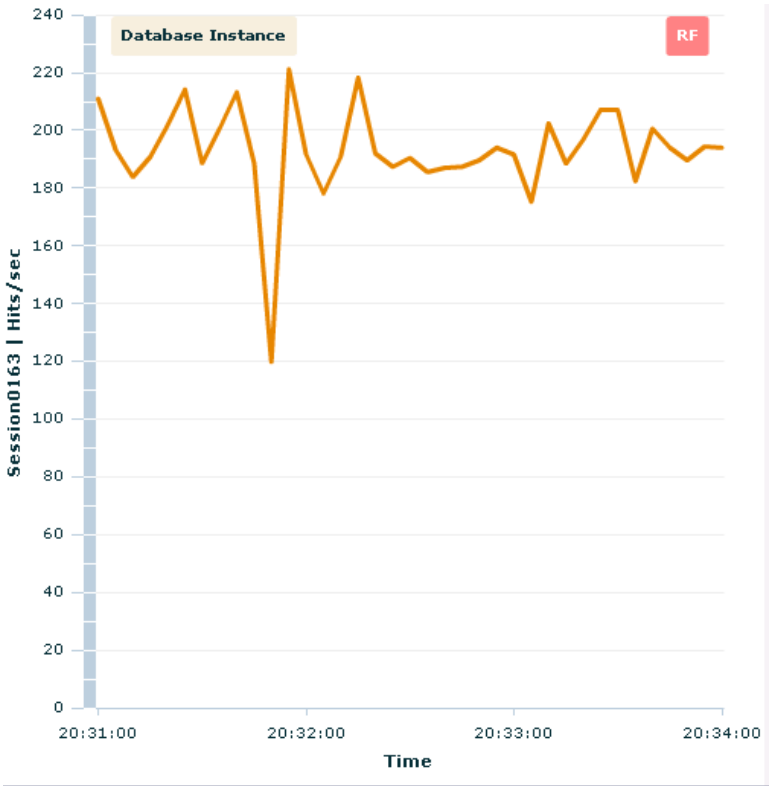

Figure 9 Performance Impact of Database Instance Failure

When the node failed, the cluster had to be reformed before the connections could failover and so there was a more prolonged impact on performance as can be seen on the following graph:

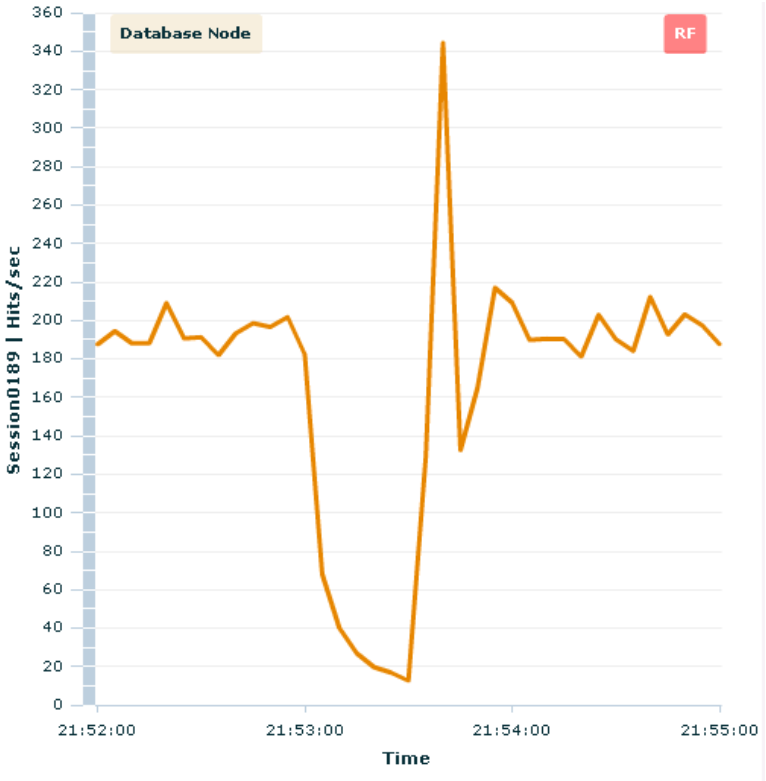

Figure 10 Performance Impact of Database Node Failure

5.2.5 Observations on Site Outage

All users were lost on site outage and the site failover procedure was followed to restore service on the standby site. The failover procedure is documented in the section entitled "8.6 Failover". The failover procedure was performed manually and the timing for each step is shown in the following table:

Table 7 Recovery Times for Unplanned Site Outage

RECOVERY STEP	ELAPSED TIME (SECONDS)	CUMULATIVE TIME (SECONDS)	NOTES
Database Data Guard Failover	33	33	These two steps performed in parallel
Siebel FS Replication Role Reversal & Mount	32		
Siebel Gateway Startup	6	39	
Siebel Server Startup	3	42	Performed in parallel across Servers
Siebel Web Server Startup	5	47	Performed in parallel across Servers
First Siebel Login	30	77	Login and select Contacts View

Each step in the process could have been executed by script with no delay between steps and in that case the total time to perform recovery, excluding the first user login, was estimated to be 47 seconds.

6 Summary of Best Practices

Here is a summary of the best practices that have been presented in this paper providing a checklist for a Siebel MAA implementation.

6.1 Best Practices Siebel Database High Availability

Here are the Siebel database best practices that should be applied to the primary and secondary site to achieve highest availability:

- Deploy Siebel on an Oracle RAC database for the highest availability and scalability.
- Use Automatic Storage Management to simplify the provisioning and management of database storage.
- Enable Oracle Flashback Database to provide the ability to "rewind" the database in the event of user errors.
- Use Oracle Recovery Manager to regularly backup the Siebel database.
- Configure database services with TAF for the connection to the Siebel database.
- Always use Hugepages for Siebel databases on Linux. Monitor memory usage and adjust the workload and parameters accordingly.
- Configure database Dead Connection Detection to actively remove dead connections in the event of Siebel Server node failure.
- For Exadata deployments, configure cluster misscount to 30 seconds to reduce downtime in the event of database node failure.
- Revalidate the configuration regularly and especially after changes are made. Exachk can be used to assist in the validation process when deployed on Exadata.

6.2 Best Practices for Siebel Application High Availability

Here are the Siebel application best practices that should be applied to the primary and secondary site to achieve highest availability:

- Use Oracle Clusterware to protect the Siebel Gateway server and deploy the Siebel Bundled Agents.
- Deploy multiple Siebel servers and deployed all critical Siebel components in a load balanced, distributed service, or clustered configuration, so that work can continue in the event of a Siebel Server node failure.
- Deploy multiple web servers so that work can continue normally if there is a web server outage.
- Deploy a load balancer in a redundant configuration and load balance web server load using our recommended logic.
- Deploy the Siebel File System on a fault tolerant filer.
- Take regular backups of the Siebel Servers, Web Servers, Gateway Servers, and Siebel File System.

- Take regular backups of the Siebel Gateway Server backing file using the Siebel server manager.
- Connect to the database through the role based services by connecting through all the possible database listeners and with connection timeouts and retries.
- Reduce TCP Keepalive Timeout on Siebel Web Servers, Siebel Servers, and Siebel Gateway Servers.

6.3 Best Practices for Disaster Readiness and Recovery

Here are the best practices for deploying a secondary site and recovery procedures in readiness for a site outage:

- Deploy a second geographically separated site that can run the Siebel workload in the event the primary site is down.
- Use Data Guard to replicate all database changes to a standby database located on the secondary site.
- Take advantage of Active Database Guard to offload read-only queries to the standby database.
- Enable Oracle Flashback Database so the old primary database can be quickly reinstated as a standby in the event of site failover.
- Continuously replicate the Siebel File System to the secondary site with minimal lag. Develop procedures for how to reverse the direction of replication in the event of failover or switchover, and procedures to clone the replica for site testing.
- Export the Siebel File System primary, standby replica, and clones, with different names to avoid mounting the incorrect one.
- Create different role based database services for the Siebel database in primary, standby and snapshot standby mode.
- Develop and document operational procedures in line with the Siebel MAA state model and state transitions.
- Use Data Guard Broker to simplify Data Guard administration.
- Use the snapshot standby to provide an updatable replica of the primary database for temporary site testing.

7 References

- MAA on Oracle Technology Network (OTN)
  http://www.oracle.com/goto/maa
- Oracle® Database High Availability Overview
  http://www.oracle.com/pls/db112/to_toc?pathname=server.112/e17157/toc.htm
- Oracle Real Application Clusters Administration and Deployment Guide
  http://www.oracle.com/pls/db112/to_toc?pathname=rac.112/e41960/toc.htm
- Oracle Clusterware Administration and Deployment Guide
  http://www.oracle.com/pls/db112/to_toc?pathname=rac.112/e41959/toc.htm
- Oracle Data Guard Concepts and Administration
  http://www.oracle.com/pls/db112/to_toc?pathname=server.112/e41134/toc.htm
- Oracle Flashback Technology
  http://docs.oracle.com/cd/E11882_01/server.112/e40540/cncptdba.htm#CNCPT1439
- Automatic Storage Management
  http://www.oracle.com/pls/db112/to_toc?pathname=server.112/e18951/toc.htm
- Oracle Recovery Manager
  http://www.oracle.com/pls/db112/to_toc?pathname=backup.112/e10642/toc.htm
- Oracle Secure Backup
  http://docs.oracle.com/cd/E11882_01/backup.112/e10642/rcmarchi.htm#sthref236
- MOS ID 361468.1, "HugePages on Oracle Linux 64-bit"
  https://support.oracle.com/epmos/faces/DocumentDisplay?id=361468.1
- MOS ID 749851.1 "HugePages and Oracle Database 11g Automatic Memory Management (AMM) on Linux"
  https://support.oracle.com/epmos/faces/DocumentDisplay?id=749851.1
- MOS ID 1392497.1 "USE_LARGE_PAGES To Enable HugePages In 11.2"
  https://support.oracle.com/epmos/faces/DocumentDisplay?id=1392497.1
- MOS ID 151972.1 "(Dead Connection Detection (DCD) Explained"
  https://support.oracle.com/epmos/faces/DocumentDisplay?id=151972.1
- Siebel Deployment Planning Guide
  http://docs.oracle.com/cd/E14004_01/books/DeplmtPlan/DeplmtPlanTOC.html
- Clusterware on OTN
  http://www.oracle.com/technetwork/products/clusterware/overview/index.html.
- MOS ID 249213.1 "Performance problems with Failover when TCP Network goes down (no IP address)"
  https://support.oracle.com/epmos/faces/DocumentDisplay?id=249213.1

- MOS ID 781927.1 "Performance Tuning Guidelines for Siebel CRM Application on Oracle Database"
  https://support.oracle.com/epmos/faces/DocumentDisplay?id=781927.1
- BIG-IP Product Suite
  http://www.f5.com/products/big-ip/
- F5 Siebel 8.0 Deployment Guide
  http://www.f5.com/pdf/deployment-guides/f5-siebel-dg.pdf
- F5 DevCentral
  http://devcentral.f5.com/
- MOS ID 1516025.1 "How To Configure NIS Master, Slave And Client Configuration In Exalogic Virtual Environment"
  https://support.oracle.com/epmos/faces/DocumentDisplay?id=1516025.1
- MOS ID 1334857.1 "32-bit Client Install On Linux x86-64: PRVF-7532 Error Occurs For gcc-4.1.2 i386"
  https://support.oracle.com/epmos/faces/DocumentDisplay?id=1334857.1
- Oracle Grid Infrastructure Bundled Agents v2
  http://www.oracle.com/technetwork/products/clusterware/overview/ogiba-v2-1916262.pdf
- "Oracle® Fusion Middleware Installation Guide for Oracle Web Tier".
  http://docs.oracle.com/cd/E23943_01/doc.1111/e14260/preparing.htm#autoId5
- MOS ID 1375035.1 "Web server will not start for Siebel CRM version 8.0.0.13, 8.1.1.6 or 8.2.2 on Linux with OHS 10g"
  https://support.oracle.com/epmos/faces/DocumentDisplay?id=1375035.1
- Siebel Performance Tuning Guide
  http://docs.oracle.com/cd/E14004_01/books/PerformTun/PerformTunTOC.html
- MOS ID 1484925.1 "When running Siebel 8.x on Exalogic the sys and admin folder need to be moved to a local file system"
  https://support.oracle.com/epmos/faces/DocumentDisplay?id=1484925.1

8 Appendix – Case Study State Transitions

In this appendix we document the detail procedures that were followed during the case study setup and outage testing. They implement the state transitions as defined in the Siebel MAA State Model as defined in Section 3.3 Siebel MAA Site State Model and State Transitions.

8.1 Primary Site Setup

8.1.1 Siebel Database Setup

The standard Exadata configuration was deployed on the primary site.

8.1.1.1 Configure Hugepages

Hugepages were configured. It is critically important that hugepages are configured when running Siebel databases on Linux database platforms.

8.1.1.2 Run Exachk

Exachk was run after the database was created to validate the configuration, and the exceptions noted in Section 4.6.2 Exadata Exachk Best Practices were applied.

The final database parameter configuration was as follows:

```
*._always_semi_join='OFF'
*._b_tree_bitmap_plans=FALSE
*._file_size_increase_increment=2143289344
*._gc_defer_time=0
*._no_or_expansion=FALSE
*._optimizer_max_permutations=100
*._partition_view_enabled=FALSE
*.archive_lag_target=0
*.audit_sys_operations=TRUE
*.audit_trail='DB'
*.cluster_database=TRUE
siebxd1.cluster_interconnects='192.168.44.227'
siebxd2.cluster_interconnects='192.168.44.228'
*.compatible='11.2.0.3'
*.control_files='+DATA_SCAM02/siebxd/controlfile/crf1','+RECO_SCAM02/siebxd/controlfile
 /crf2'
*.db_block_checking='FALSE'
*.db_block_checksum='typical'
*.db_block_size=8192
*.db_create_file_dest='+DATA_SCAM02'
*.db_create_online_log_dest_1='+DATA_SCAM02'
*.db_domain=''
*.db_files=1024
*.db_lost_write_protect='typical'
*.db_name='SIEBXD'#Reset to original value by RMAN
*.db_recovery_file_dest='+RECO_SCAM02'
*.db_recovery_file_dest_size=1990152M
*.diagnostic_dest='/u01/app/oracle_siebel'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=ORCLXDB)'
```

```
*.fast_start_mttr_target=300
*.global_names=TRUE
siebxd1.instance_number=1
siebxd2.instance_number=2
*.job_queue_processes=1000
*.listener_networks='((NAME=network2)(LOCAL_LISTENER=LISTENER_IBLOCAL)(REMOTE_LISTENER=
 LISTENER_IBREMOTE))','((NAME=network1)(LOCAL_LISTENER=LISTENER_IPLOCAL)(REMOTE_LISTEN
 ER=LISTENER_IPREMOTE))'
*.local_listener=''
*.log_buffer=134217728
*.nls_sort='BINARY'
*.O7_DICTIONARY_ACCESSIBILITY=TRUE
*.open_cursors=1000
*.optimizer_dynamic_sampling=1
*.optimizer_index_caching=0
*.optimizer_index_cost_adj=1
*.optimizer_mode='ALL_ROWS'
*.os_authent_prefix=''
*.parallel_adaptive_multi_user=FALSE
*.parallel_degree_policy='manual'
*.parallel_threads_per_cpu=1
*.pga_aggregate_target=12884901888
*.processes=2000
*.query_rewrite_enabled='FALSE'
*.query_rewrite_integrity='enforced'
*.recyclebin='ON'
*.remote_listener='scam02-scan7'
*.remote_login_passwordfile='EXCLUSIVE'
*.resource_limit=FALSE
*.resource_manager_plan='DEFAULT_PLAN'
*.sec_case_sensitive_logon=FALSE
*.service_names=''
*.sga_target=10032385536
*.shared_servers=0
*.sql92_security=TRUE
*.star_transformation_enabled='FALSE'
*.statistics_level='typical'
siebxd1.thread=1
siebxd2.thread=2
*.undo_retention=1800
siebxd1.undo_tablespace='UNDOTBS1'
siebxd2.undo_tablespace='UNDOTBS2'
*.use_large_pages='only'
```

8.1.1.3 Siebel Database Creation

The Siebel database was copied over from another test system using RMAN duplicate functionality. The database was registered on Oracle Clusterware as follows:

```
srvctl add database -d siebxd –n siebxd -o
 /u01/app/oracle_siebel/product/11.2.0.3/dbhome_siebel
srvctl add instance -d siebxd -i “siebxd1” -n scam02db07
srvctl add instance -d siebxd -i “siebxd2” -n scam02db08
srvctl modify database -d siebxd -a "DATA_SCAM02,RECO_SCAM02"
```

8.1.1.4 Database Service Creation

```
srvctl add service -d siebxd -s SIEB_PRIM -r "siebxd1,siebxd2" -e SELECT -m BASIC -P
 BASIC -w 5 -z 24 -j LONG -l PRIMARY -y AUTOMATIC
srvctl start service –d siebxd –s SIEB_PRIM
srvctl add service -d siebxd -s SIEB_STBY -r "siebxd1,siebxd2" -e SELECT -m BASIC -P
 BASIC -w 5 -z 24 -j LONG -l PHYSICAL_STANDBY -y AUTOMATIC
srvctl add service -d siebxd -s SIEB_STBY_TEST -r "siebxd1,siebxd2" -e SELECT -m BASIC
 -P BASIC -w 5 -z 24 -j LONG -l SNAPSHOT_STANDBY -y AUTOMATIC
```

8.1.1.5 Dead Connection Detection

Dead connection detection (DCD) was configured by adding the following parameter to the sqlnet.ora of the Siebel database on each database server:

```
sqlnet.expire_time = 1
```

8.1.2 Shared File System Creation

File systems were created in the Exalogic ZFS Storage Appliance for site 1 as follows:

PURPOSE	SHARE TYPE	EXPORTED AS (MOUNTED AS)	SITE STATE	COMMENTS
Siebel File System	Local Replicated	/export/siebelfs	Site 1 Primary	NFS v4, Constructed from site 2 replica on switchover/failover from site 2
Site 2 Siebel File System Replica	Replica	/export/siebelfs_site2_replica	Site 2 Primary	NFS v4
Siebel Gateway Home	Local	/export/siebel_gateway_home	All	NFS v4, no replication
Grid OCR and Vote	Local	/export/siebel_gateway_cluster	All	NFS v3, no replication

8.1.3 vServer Creation

Using the BUI, vServer types were created with the following characteristics:

PURPOSE	VSERVER TYPE NAME	VCPU PER VSERVER	RAM PER VSERVER	STORAGE PER VSERVER
Siebel Gateway	Siebel-Gateway	2	4	24
Siebel Web	Siebel-Web	4	16	24
Siebel Server	Siebel-Server	8	32	24

Distribution groups ensure that vServers are spread across the available Exalogic physical servers. vServers in the same distribution group are not allowed to run on the same physical server. Using the BUI, the following distribution groups were created:

PURPOSE	VSERVER TYPE
Siebel Gateway	Siebel-Gateway
Siebel Web	Siebel-Web
Siebel Server	Siebel-Server

Using the BUI, vServers with the following characteristics were created:

PURPOSE	QUANTITY	VSERVER TYPE	DISTRIBUTION GROUP
Siebel Gateway	2	Siebel-Gateway	Siebel-Gateway
Siebel Web	2	Siebel-Web	Siebel-Web
Siebel Server	2	Siebel-Server	Siebel-Server

When creating the vServers it was important to configure interfaces to allow access to the database servers, ZFS storage, clients and administrators. Note, the Siebel Web vServers do not need access to the storage network.

In addition, it was necessary to configure a private network between the two Gateway Servers for the Gateway Cluster private interconnect.

The vServers that were created were as follows:

PURPOSE	HOSTNAME	ZFS STORAGE NETWORK ADDRESS	PRIVATE CLUSTER NETWORK ADDRESS
Gateway Node 1	scan03vm0002-eoib1	172.17.0.255	192.168.0.2
Gateway Node 2	scan03vm0005-eoib1	172.17.0.254	192.168.0.1
Siebel Server Node 1	scan03vm0047-eoib1	172.17.0.26	
Siebel Server Node 2	scan03vm0030-eoib1	172.17.0.251	
Siebel Web Server Node 1	scan03vm0036-eoib1		
Siebel Web Server Node 2	scan03vm0052-eoib1		

After creation, each vServer was configured as follows:

- Hostname (/etc/sysconfig/network)
- DNS (/etc/resolv.conf)
- NTP was configured to synchronize the system clock
- Root partition was extended to use spare space (expanded to approximately 23GB) – see MOS ID 1516025.1 "How To Configure NIS Master, Slave And Client Configuration In Exalogic Virtual Environment" for an example of how this is done.
- NIS Client Configuration (see above support document for details)
- TCP Keepalive Parameters were configured as follows on each vServer:

```
net.ipv4.tcp_keepalive_time = 60
net.ipv4.tcp_keepalive_probes = 6
net.ipv4.tcp_keepalive_intvl = 10
```

8.1.4 Shared File Systems Mounted on vServers

Shared file systems were mounted as follows:

PURPOSE	MOUNTED ON	EXPORTED AS (MOUNTED AS)	SITE STATE	MOUNT OPTIONS
Siebel File System	All Siebel and Gateway Servers	/export/siebelfs (/siebelfs)	Site 1 Primary	nfs4 rw,bg,hard,nointr,rsize=131072,wsize=131072
Site 2 Siebel File System Replica	Optional	/export/siebelfs_site2_replica (/siebelfs_site2_replica)	Site 2 Primary	nfs4 ro,bg,hard,nointr,rsize=131072,wsize=131072
Siebel Gateway Home	Local Shared	/export/siebel_gateway_home (/siebel_gateway_home)	All	nfs4 rw,bg,hard,nointr,rsize=131072,wsize=131072
Grid OCR and Vote	Local Shared	/export/siebel_gateway_cluster (/siebel_gateway_cluster)	All	nfs rw,bg,hard,nointr,rsize=32768,wsize=32768, tcp,noac,vers=3,timeo=600,actimeo=0

8.1.5 File System Folders Created

Software folders were created as follows for site 1:

PURPOSE	FOLDER NAME	LOCATION	OS OWNER
Oracle Inventory	/u01/app/oraInventory	Created locally on all Servers	siebel
Siebel Config	/var/adm/siebel		siebel
Siebel File System	/siebelfs/siebelfs	Mounted on all Siebel Servers and Gateway Servers	siebel
Database Client Home	/u01/app/siebel/product/11.2.0/db_client_x86	Created locally on all Siebel Servers and Gateway Servers	siebel
Gateway Home	/siebel_gateway_home/app/siebel/product/8.1.1/ses	Mounted on all Gateway Servers	siebel
Cluster OCR and Vote	/siebel_gateway_cluster/registry		grid
Cluster Home	/u01/app/grid/product/11.2.0.3/grid	Created locally on all Gateway Servers	grid
Cluster Oracle Base	/u01/app/grid/base		grid
Bundled Agents	/u01/app/grid/product/2.1/xag		grid
Siebel Server Home	/u01/app/siebel/product/8.1.1/ses	Created locally on all Siebel Servers	siebel
OHS Home	/u01/app/siebel/product/11.1.1.6/ohs	Created locally on all Siebel Web Servers	siebel
Siebel Web Home	/u01/app/siebel/product/8.1.1/swe		siebel

All the above folders were owned by the oinstall OS group.

8.1.6 Database Client Software Installation and Configuration

32-bit Oracle database client software is required on the Siebel Servers and Siebel Gateway Servers to facilitate connection to the Siebel database. The following options were selected during the installation process:

- Choose: Runtime Client
- Ignore dependencies on gcc-4.1.2 - see MOS ID 1334857.1 "32-bit Client Install On Linux x86-64: PRVF-7532 Error Occurs For gcc-4.1.2 i386" for details.

Once installed, the siebel user environment was configured:

```
cat > ~/.bashrc <<'EOF!'
export ORACLE_HOME=/u01/app/siebel/product/11.2.0/db_client_x86
export PATH=$PATH:$ORACLE_HOME/bin
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
EOF!
```

Configure connectivity to the database:

```
mkdir –p $ORACLE_HOME/network/admin
cat > $ORACLE_HOME/network/admin/tnsnames.ora <<EOF!
SIEBXD =
  (DESCRIPTION =
    (CONNECT_TIMEOUT=5)
    (TRANSPORT_CONNECT_TIMEOUT=3)
    (RETRY_COUNT=3)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS = (PROTOCOL = TCP)(HOST = scam02-scan7)(PORT = 1521)))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = sieb_prim))
  )
EOF!
```

8.1.7 Clustered Siebel Gateway Configuration

8.1.7.1 Gateway Server Software Installation

The following actions were performed as user siebel on only one of the Gateway Server vServers:

```
cd …/image/8.1.1.9/Linux/Server/Siebel_Enterprise_Server/Disk1/install
export DISPLAY=scan03vm0055-eoib1:4.0
./runInstaller -oneclick -invPtrLoc /u01/app/oraInventory/oraInst.loc
```

DIALOG	RESPONSE
Siebel Home	/siebel_gateway_home/app/siebel/product/8.1.1/ses

8.1.7.2 Gateway Server Configuration

The following actions were performed as user siebel on only one of the Gateway Server vServers:

```
cd /siebel_gateway_home/app/siebel/product/8.1.1/ses
. gtwysrvr/cfgenv.sh
export DISPLAY=scan03vm0055-eoib1:4.0
cd config
./config.sh -mode enterprise
```

DIALOG	RESPONSE
Main Task Selection	Create New Configuration
Configuration Task Selection	Configure a New Gateway Name Server
Port	2320

8.1.7.3 Siebel Enterprise Configuration

The following actions were performed as user siebel on scm03vm0005-eoib1:

```
cd /siebel_gateway_home/app/siebel/product/8.1.1/ses
. gtwysrvr/cfgenv.sh
export DISPLAY=scan03vm0055-eoib1:4.0
cd config
./config.sh -mode enterprise
```

DIALOG	RESPONSE
Main Task Selection	Create New Configuration
Configuration Task Selection	Configure a New Enterprise in a Gateway Name Server
Gateway Name Server Authentication User Account Name	sadmin
Gateway Name Server Authentication User Account Password	sadmin
Gateway Name Server Host Name	scan03vm0005-eoib1
Gateway Name Server TCP/IP Port	2320
Siebel Enterprise Name	SBA
Enterprise Description	SBA
Primary Siebel File System	/siebelfs/siebelfs
RDBMS Platform	Oracle Database Enterprise Edition
Database Table Owner	ORAPERF
Oracle SQLNet Connect String	siebxd
Siebel Database User Account Name	SADMIN
Siebel Database User Account Password	SADMIN
Enterprise Security Authentication Profile	Database Authentication (development only)
Security Adapter Name	DBSecAdpt
Propagate Authentication Settings to the Gateway Name Server	true
Additional Enterprise Task Selection	

8.1.7.4 Gateway Server Shutdown

The following actions were performed as user siebel on only one of the Gateway Server vServers:

```
cd /siebel_gateway_home/app/siebel/product/8.1.1/ses/gtwysrvr
. siebenv.sh
cd ./bin/stop_ns
```

8.1.7.5 Oracle Clusterware Installation

Virtual IP addresses were allocated in EMOC as follows:

PURPOSE	NAME	ADDRESS
Gateway Node1 Virtual Host Name	scan03vm0056-eoib1.us.oracle.com	10.133.227.29
Gateway Node 2 Virtual Host Name	scan03vm0027-eoib1.us.oracle.com	10.133.227.0
Siebel Gateway VIP		10.133.226.230
Cluster SCAN (temporary)	dummyscan	10.133.226.232

A dummy scan address entry was created in /etc/hosts file on each Gateway Server:

```
10.133.226.232   dummyscan
```

Clusterware was installed on the two Gateway Servers by executing the following commands as user grid on scan03vm0002-eoib1:

```
cd …/grid
export DISPLAY=scan03vm0055-eoib1:4.0
./runInstaller
```

INSTALLER OPTION	RESPONSE
Software Updates	Skip software updates
	Install and Configure Grid Infrastructure for a Cluster
	Typical Installation
SCAN Name	dummyscan
	Add the second host name and add the appropriate virtual node names (VIP) as per the earlier allocation
	Configure the network interfaces:
	Public: Client network with virtual host names
	Private: Private cluster network
	Setup SSH connectivity through the installer and configure
Oracle Base	/u01/app/grid/base
Oracle Home	/u01/app/grid/product/11.2.0.3/grid
	Let the install fix the parameters it can and ignore the swap space issue.
	Error reported during the cluster verification stage due to the dummy scan address may be ignored.

8.1.7.6 Oracle Clusterware Bundle Agent Installation and Configuration

Bundled Agents 2.1 (the latest at that time) was downloaded and installed following the document "Oracle Grid Infrastructure Bundled Agents v2".

The following command was run as grid on scan03vm0002-eoib1 to install the bundled agent software on all the Gateway Servers:

```
cd …/xagpack_2.1/xag
./xagsetup.sh --install --directory /u01/app/grid/product/2.1/xag --all_nodes
```

As root on scan03vm0002-eoib1:

```
export CRS_HOME=/u01/app/grid/product/11.2.0.3/grid
cd /u01/app/grid/product/2.1/xag
./agctl.pl add siebel_gateway SBA --siebel_home
 /siebel_gateway_home/app/siebel/product/8.1.1/ses/gtwysrvr --oracle_home $ CRS_HOME -
 -oracle_client_home /u01/app/siebel/product/11.2.0/db_client_x86/lib --nodes
 scan03vm0002-eoib1,scan03vm0005-eoib1 --network 1 --ip 10.133.226.230 --user siebel -
 -group oinstall
./agctl.pl config siebel_gateway SBA
```

The configuration was tested by executing the following commands on each Gateway Server as user siebel:

```
export CRS_HOME=/u01/app/grid/product/11.2.0.3/grid
cd /u01/app/grid/product/2.1/xag
./agctl.pl config siebel_gateway SBA
./agctl.pl start siebel_gateway SBA –-node scan03vm0002-eoib1

cd /siebel_gateway_home/app/siebel/product/8.1.1/ses/gtwysrvr
. siebenv.sh
./bin/srvrmgr /g 10.133.226.230 /e SBA /u sadmin /p sadmin
```

8.1.8 Siebel Server Configuration

8.1.8.1 Install Siebel Server Software

The following was executed on each Siebel Server machine as user siebel:

```
cd …/image/8.1.1.9/Linux/Server/Siebel_Enterprise_Server/Disk1/install
export DISPLAY=scan03vm0055-eoib1:4.0
./runInstaller -oneclick -invPtrLoc /u01/app/oraInventory/oraInst.loc
```

INSTALL OPTION	RESPONSE
Installation type	New Installation
Home:	/u01/app/siebel/product/8.1.1/ses
Modules:	Siebel Server, and Database Utils (needed for siebelfs files)

Note, the 8.1.1.0 installer runs first and then the patch to 8.1.1.9 is applied. Be careful not to cancel the install before it completes.

8.1.8.2 Configure and Start Siebel Server

The following was executed on each Siebel Server machine as user siebel:

```
cd /u01/app/siebel/product/8.1.1/ses
. siebsrvr/cfgenv.sh
export DISPLAY=scan03vm0055-eoib1:4.0
cd config
./config.sh -mode siebsrvr
```

CONFIGURATION OPTION	RESPONSE
	Create New Configuration
Gateway Auth:	sadmin/sadmin
Gateway Name:	10.133.226.230
Gateway Port:	2320
Siebel Enterprise Name:	SBA
Siebel Server Name:	SBA1, SBA2, etc. (note, no hyphens)
Component Groups:	EAI, Communications, Loyalty, Loyalty Engine, Siebel Financial Services
Port:	2321
Sync Manager Port:	40400
	No additional server tasks
Register External Oracle DB ODBC Driver:	uncheck

8.1.8.3 Copy Seed Files to Siebel File System

The following was executed on each Siebel Server machine as user siebel:

```
cp /u01/app/siebel/product/8.1.1/sestemp/dbsrvr/files/* /siebelfs/siebelfs/att/
```

8.1.8.4 Test

The following was executed on each Siebel Server machine as user siebel:

```
cd /u01/app/siebel/product/8.1.1/ses/siebsrvr
. siebenv.sh
cd ./bin/odbcsql
ODBC> set source SBA_DSN
ODBC> connect sadmin/sadmin
ODBC> select count(*) from oraperf.s_evt_act;
COUNT(*)
----------------------------------------------------
6099058
----------------------------------------------------
(query+fetch time: 7.2s, rows fetched: 1, time per row: 7.2s)
ODBC> exit

./srvrmgr /g 10.133.226.230 /e SBA /u sadmin /p sadmin
srvrmgr> list server
```

8.1.8.5 Start the Siebel Server

The following was executed on each Siebel Server machine as user siebel:

```
cd /u01/app/siebel/product/8.1.1/ses/siebsrvr
. siebenv.sh
cd ./bin/start_server all
```

8.1.9 Siebel Web Server Configuration

The 32-bit Oracle HTTP Server, part of the Oracle Fusion Middleware 11.1.1.6.0 Web Tier Utilities x86, was downloaded. Note, this must be the 32-bit version for Siebel. The following documents were referenced during the installation:

- "Oracle® Fusion Middleware Installation Guide for Oracle Web Tier".
- MOS ID 1375035.1 "Web server will not start for Siebel CRM version 8.0.0.13, 8.1.1.6 or 8.2.2 on Linux with OHS 10g".

8.1.9.1 Install Web Server Software

The following was executed on each Web Server machine as user siebel:

```
linux32 bash
cd /stage2/stage/FMW_Web_Tier_11.1.1.6_Linux_x86/Disk1
export DISPLAY=scan03vm0055-eoib1:4.0
./runInstaller
```

INSTALL OPTION	RESPONSE
Inventory Location:	/u01/app/oraInventory
Run:	/u01/app/oraInventory/createCentralInventory.sh
Install option:	Install and Configure
Home:	/u01/app/siebel/product/11.1.1.6/ohs
Product:	Select Oracle HTTP Server
Instance home:	/u01/app/siebel/product/11.1.1.6/ohs/OHS/instances/instance1
Port Configuration:	Auto

There was a linking error during the installation. The workaround was follows and detailed steps in the official install guide (link above). As root:

```
mv /usr/bin/gcc /usr/bin/gcc.orig
cat >/usr/bin/gcc41 <<'EOF!'
#!/bin/sh
exec /usr/bin/gcc.orig -m32 -static-libgcc -B /usr/lib/gcc/x86_64-redhat-linux/4.1.1/
 $*
EOF!
chmod 755 /usr/bin/gcc41
ln -s -f /usr/bin/gcc41 /usr/bin/gcc
```

8.1.9.2 Create a SWSE Logical Profile

As Siebel on the active Gateway Server:

```
cd /siebel_gateway_home/app/siebel/product/8.1.1/ses
. gtwysrvr/cfgenv.sh
export DISPLAY=scan03vm0055-eoib1:4.0
cd config
```

./CONFIG.SH -MODE ENTERPRISECONFIGURATION OPTION	RESPONSE
Select:	Create New Configuration
Select:	Configure a New Siebel Web Server Extension Logical Profile
Siebel Enterprise Name :	SBA
Profile Name:	/siebel_gateway_home/app/siebel/product/8.1.1/ses/gtwysrvr/admin/Webserver

Collect Application Specific Stats:	uncheck
Compression Type:	None
HTTP 1.1:	Checked
Login Timeout:	300
Active Session Timeout:	900
HTTP Port:	8000
HTTPS Port:	443
FQDN:	Blank
HI Login:	sadmin/sadmin
LI Login:	sadmin/sadmin
Siebel Enterprise Security Token:	token
Stats Page:	_stats.swe
Connection protocol:	TCPIP
SSL/TLS:	None

The eapps.cfg file was edited to have correct syntax:

```
vi /siebel_gateway_home/app/siebel/product/8.1.1/ses/gtwysrvr/admin/Webserver/eapps.cfg
```

Replacing:

```
VirtualHostsFile   = $(SWSERoot)\admin\lbconfig.cfg
```

With:

```
VirtualHostsFile   = $(SWSERoot)/admin/lbconfig.txt
```

The file was copied to web servers:

```
cd /siebel_gateway_home/app/siebel/product/8.1.1/ses/gtwysrvr/admin
scp  -r Webserver siebel@scan03vm0036-eoib1:
scp  -r Webserver siebel@scan03vm0052-eoib1:
```

8.1.9.3 Generate Load Balancing Configuration File

As user siebel on one of the Siebel servers:

```
cd /u01/app/siebel/product/8.1.1/ses/siebsrvr
. siebenv.sh
cd ./bin/srvrmgr /g 10.133.226.230 /e SBA /u sadmin /p sadmin
srvrmgr> generate lbconfig
```

The file was edited to remove any object managers that were not being used:

```
vi /u01/app/siebel/product/8.1.1/ses/siebsrvr/admin/lbconfig.txt
```

The file was copied to the web servers:

```
cd /u01/app/siebel/product/8.1.1/ses/siebsrvr/admin
scp lbconfig.txt siebel@scan03vm0036-eoib1:Webserver/
scp lbconfig.txt siebel@scan03vm0052-eoib1:Webserver/
```

8.1.9.4 Install Siebel Web Server Software

The following was executed as user siebel on each Web Server:

```
cd …/image/8.1.1.9/Linux/Server/Siebel_Web_Server_Extension/Disk1/install
export DISPLAY=scan03vm0055-eoib1:4.0
./runInstaller -oneclick -invPtrLoc /u01/app/oraInventory/oraInst.loc
```

INSTALL OPTION	RESPONSE
	New Installation
Home:	/u01/app/siebel/product/8.1.1/swe
Language:	English

Note, the 8.1.1.0 installer runs first and then the patch to 8.1.1.9 is applied. Be careful not to cancel the install before it completes.

8.1.9.5 Configure and Start SWE

The following was executed as siebel on each Web Server:

```
cd /u01/app/siebel/product/8.1.1/swe
. cfgenv.sh
export DISPLAY=scan03vm0055-eoib1:4.0
cd config
./config.sh -mode swse
```

CONFIGURATION OPTION	RESPONSE
Option:	Apply a SWSE Logical Profile
Load balancing:	Siebel Native
Location:	/siebel/home/Webserver
Web Server Instance Location:	/u01/app/siebel/product/11.1.1.6/ohs/OHS/instances/instance1/OHS/ohs1
Restart Web Server:	checked

8.1.9.6 Configuring the Web Server:

The httpd.conf file was edited as follows:

```
vi /u01/…/11.1.1.6/ohs/OHS/instances/instance1/config/OHS/ohs1/httpd.conf
```

PARAMETER	VALUE
KeepAliveTimeout	15
MaxKeepAliveRequests	0
UseCanonicalName	Off
In the section entitled <IfModule mpm_worker_module>:	
ThreadLimit	256
ServerLimit	5
StartServers	1
MaxClients	1200
MinSpareThreads	1
MaxSpareThreads	257
ThreadsPerChild	256
MaxRequestsPerChild	0

As per the "Siebel Performance Tuning Guide", the following line was added to the opmn.xml file on each web server immediately after the SIEBEL_CODEPAGE line:

```
<variable id="SIEBEL_OSD_PTHREAD_STACK_SIZE" value="65536"/>
```

The opmn.xml file was located at:

```
/u01/app/siebel/product/11.1.1.6/ohs/OHS/opmn/conf/opmn.xml
```

The following is a section of the file after editing:

```
    <variable id="SIEBEL_CODEPAGE" value="1252"/>
    <variable id="SIEBEL_OSD_PTHREAD_STACK_SIZE" value="65536"/>
  </environment>
```

8.1.9.7 Test

As user siebel on each Web Server:

- To get status:

```
/u01/app/siebel/product/11.1.1.6/ohs/OHS/instances/instance1/bin/opmnctl status
```

- To stop:

```
/u01/app/siebel/product/11.1.1.6/ohs/OHS/instances/instance1/bin/opmnctl stopall
```

- To start:

```
/u01/app/siebel/product/11.1.1.6/ohs/OHS/instances/instance1/bin/opmnctl startall
```

8.1.10 Web Server Load Balancing Configuration

The web server load balancing was configured in the F5 load balancer. The configuration was performed in accordance with the second chapter of the F5 "F5 Siebel Deployment Guide", but with the following changes in order to achieve the logic outlined in the section entitled "3.2.2.1 Optimal Siebel Web Server Load Balancing":

- HTTP health monitor:

PROPERTY	VALUE
Interval	5
Timeout	16
Send String	GET /index.html \r\n

- Pool:

PROPERTY	VALUE
Load Balancing Method	Round Robin
Action on Service Down	Reject

- iRule:

```
when HTTP_REQUEST {
  set lb_retries 0
  set lb_failures 0
}

when LB_FAILED {
  if { $lb_failures == 0 } {
    set lb_retries [expr [active_members [LB::server pool] ] * 3]
  }
  incr lb_failures
  if { $lb_retries > 0 } {
    if { $lb_failures <= $lb_retries } {
      LB::reselect
    } else {
      log local0.error "Client [IP::client_addr]:[TCP::client_port] load balance failed
 after $lb_retries reselects"
    }
  }
}
```

- Virtual Server:

PROPERTY	VALUE
Default Persistence Profile	None

8.2 Secondary Site Setup

In this case study the secondary site was located on Exalogic and Exadata, but with a bare metal (non-virtualized) Exalogic configuration. Many of the steps are the same as the primary site setup on the virtualized Exalogic and so here we will only describe where the process was different.

8.2.1 Establish Standby Database

8.2.1.1 Database Server Setup

The standard Exadata configuration was deployed on the secondary site with disk groups +DATA_SCAM08 and +RECO_SCAM08.

It was decided to use the following naming for the database and instances on the secondary site:

SITE	DB_UNIQUE_NAME	DB_NAME	INSTANCES
1 (primary)	siebxd	siebxd	siebxd1 on scam02db03, and siebxd2 on scam02db04
2 (secondary)	s2_siebxd	siebxd	siebxd1 on scam08db07, and siebxd2 on scam08db08

8.2.1.2 Oracle Home Installation and Initial Database Configuration

A database software home (Oracle home) was installed on the secondary site database server nodes with the following attributes:

- Created under the same user name oracle_siebel and with the same folder name as the primary for simplicity.
- Same version and patch level as the primary.

The oracle_siebel OS environment was configured on each database server, for example:

```
cat >siebxd.env <<'EOF'
export ORACLE_HOME=/u01/app/oracle_siebel/product/11.2.0.3/dbhome_siebel
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_BASE=/u01/app/oracle_siebel
export ORACLE_SID=siebxd1
EOF
```

User equivalence was established for the oracle_siebel user on the database servers.

The database folders were created in the +DATA_SCAM08 disk group, as user grid on scam08db03:

```
asmcmd <<EOF
mkdir +DATA_SCAM08/s2_siebxd
mkdir +DATA_SCAM08/s2_siebxd/datafile
mkdir +DATA_SCAM08/s2_siebxd/onlinelog
mkdir +DATA_SCAM08/s2_siebxd/controlfile
EOF
```

Init.ora files were created on all the site 2 nodes, for example as oracle_siebel on scam08db03:

```
cd /u01/app/oracle_siebel/product/11.2.0.3/dbhome_siebel/dbs
cat >initsiebxd1.ora <<'EOF'
spfile='+DATA_SCAM08/s2_siebxd/spfilesiebxd.ora'
EOF
```

Aliases were added to the tnsnames.ora file on each node to allow the database instances to locate their local listeners, for example on scam08db03 (note, different on each node):

```
LISTENER_IPLOCAL =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = scam0803-vip.us.oracle.com)(PORT = 1521))
    )
  )
```

The database parameters were configured on the standby with the same considerations as on the primary. To simplify the configuration process the parameters were dumped on the primary, copied to the secondary, edited, and then an spfile was created on the secondary site:

As oracle_siebel on scam02db07:

```
cd $ORACLE_HOME/dbs
sqlplus / as sysdba <<EOF
create pfile='standby_params' from spfile;
EOF
scp standby_params oracle_siebel@scam08db07:$ORACLE_HOME/dbs/
```

As oracle_siebel on scam08db03:

```
cd $ORACLE_HOME/dbs
vi standby_params
sqlplus / as sysdba <<EOF
create spfile='+DATA_SCAM08/s2_siebxd/spfilesiebxd.ora' from pfile='standby_params';
EOF
```

The Data Guard parameters necessary for establishing a standby database were considered separately and later on in the configuration process.

The following parameters were found to need different settings on the standby:

```
siebxd1.cluster_interconnects='192.168.218.130'
siebxd2.cluster_interconnects='192.168.218.131'
*.control_files='+DATA_SCAM08/s2_siebxd/controlfile/crf1','+RECO_SCAM08/s2_siebxd/contr
 olfile/crf2'
*.db_create_file_dest='+DATA_SCAM08'
*.db_create_online_log_dest_1='+DATA_SCAM08'
*.db_recovery_file_dest='+RECO_SCAM08'
*.remote_listener='scam08-scan3'
```

8.2.1.3 Dead Connection Detection

Dead connection detection (DCD) was configured by adding the following parameter to the sqlnet.ora of the Siebel database on each database server:

```
sqlnet.expire_time = 1
```

8.2.1.4 Networking Configuration

The following aliases were added to tnsnames.ora in each Oracle home on site 1 and site 2 to allow Data Guard service and broker connectivity between sites:

```
siebxd_DGMGRL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = scam02-scan7)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = siebxd_DGMGRL)
    )
  )

dg_siebxd =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = scam02-scan7)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = siebxd)
    )
  )

s2_siebxd_DGMGRL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = scam08-scan3)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = s2_siebxd_DGMGRL)
    )
  )
```

```
dg_s2_siebxd =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = scam08-scan3)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = s2_siebxd)
    )
  )
```

An instance definition was added to the grid infrastructure listener.ora file on each node to allow broker connectivity to the instances, for example on scam08db03:

```
SID_LIST_LISTENER=
  (SID_LIST=
   (SID_DESC=
     (SID_NAME=siebxd1)
     (GLOBAL_DBNAME=s2_siebxd_DGMGRL)
     (ORACLE_HOME=/u01/app/oracle_siebel/product/11.2.0.3/dbhome_siebel)
     (ENVS=
  "TNS_ADMIN=/u01/app/oracle_siebel/product/11.2.0.3/dbhome_siebel/network/admin")
   )
)
```

Note, the settings were different for each node as per the following table:

NODE	SID_NAME	GLOBAL_DBNAME
scam02db07	siebxd1	siebxd_DGMGRL
scam02db08	siebxd2	siebxd_DGMGRL
scam08db03	siebxd1	s2_siebxd_DGMGRL
scam08db04	siebxd2	s2_siebxd_DGMGRL

The listener was restarted to bring these changes into effect.

8.2.1.5 Enable Archive Log Mode, Flashback Database and Force Logging on Primary

Archive log mode, flashback database and force logging were enabled on the primary, as user oracle_siebel on scam02db07:

```
srvctl stop database –d siebxd

sqlplus / as sysdba <<EOF
startup mount
alter database archivelog;
alter database flashback on;
alter database open;
alter database force logging;
EOF

srvctl start database –d siebxd
```

8.2.1.6 Establish Password Files

A password file was created on scam02db07 and copied to all other nodes on the primary and secondary sites, as oracle_siebel on scam02db07:

```
cd $ORACLE_HOME/dbs
orapwd file=orapwsiebxd1 password=welcome1
scp orapwsiebxd1 oracle_siebel@scam02db08:$ORACLE_HOME/dbs/orapwsiebxd2
scp orapwsiebxd1 oracle_siebel@scam08db03:$ORACLE_HOME/dbs/orapwsiebxd1
scp orapwsiebxd1 oracle_siebel@scam08db04:$ORACLE_HOME/dbs/orapwsiebxd2\
```

8.2.1.7 Create Standby Redo Logs on Primary

The following was executed as user oracle_siebel on scam02db07.

To determine the number and size of redo logs the following was executed:

```
sqlplus / as sysdba <<'EOF'
select thread#, count(*) log_count, max(bytes) log_size from v$log group by
 thread#;
EOF
```

The output was:

```
   THREAD#  LOG_COUNT   LOG_SIZE
---------- ---------- ----------
         1          4 4294967296
         2          4 4294967296
```

Standby redo logs were created of the same size for each log file in each thread, plus one extra for each thread:

```
sqlplus / as sysdba <<'EOF'
alter database add standby logfile thread 1 size 4294967296;
alter database add standby logfile thread 1 size 4294967296;
alter database add standby logfile thread 1 size 4294967296;
alter database add standby logfile thread 1 size 4294967296;
alter database add standby logfile thread 1 size 4294967296;

alter database add standby logfile thread 2 size 4294967296;
alter database add standby logfile thread 2 size 4294967296;
alter database add standby logfile thread 2 size 4294967296;
alter database add standby logfile thread 2 size 4294967296;
alter database add standby logfile thread 2 size 4294967296;
EOF
```

8.2.1.8 Configure Data Guard Parameters on Primary

The following was executed as user oracle_siebel on scam02db07 (the instance is assumed to be running):

```
sqlplus / as sysdba <<'EOF'
alter system set dg_broker_start=false scope=both sid='*';
alter system set log_archive_config='dg_config=(siebxd,s2_siebxd)' scope=both sid='*';
alter system set log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST' scope=both
 sid='*';
alter system set log_archive_dest_2='SERVICE=dg_s2_siebxd
 valid_for=(online_logfiles,primary_role) db_unique_name=s2_siebxd LGWR ASYNC=20480
 OPTIONAL REOPEN=15 NET_TIMEOUT=30' scope=both sid='*';
alter system set log_archive_dest_state_2='defer' scope=both sid='*';
alter system set standby_file_management='AUTO' scope=both sid='*';
alter system set DG_BROKER_CONFIG_FILE1='+DATA_SCAM02/siebxd/dg1.dat' scope=both
 sid='*';
alter system set DG_BROKER_CONFIG_FILE2='+RECO_SCAM02/siebxd/dg2.dat' scope=both
 sid='*';
EOF
```

8.2.1.9 Configure Data Guard Parameters on Secondary

The following was executed as user oracle_siebel on scam08db03:

```
sqlplus / as sysdba <<'EOF'
startup nomount force;
alter system set dg_broker_start=false scope=both sid='*';

alter system set db_unique_name='s2_siebxd' scope=spfile sid='*';
alter system set log_archive_config='dg_config=(siebxd,s2_siebxd)' scope=both sid='*';
alter system set log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST' scope=both
 sid='*';
alter system set log_archive_dest_2='SERVICE=dg_siebxd
 valid_for=(online_logfiles,primary_role) db_unique_name=siebxd LGWR ASYNC=20480
 OPTIONAL REOPEN=15 NET_TIMEOUT=30' scope=both sid='*';
alter system set log_archive_dest_state_2='defer' scope=both sid='*';
alter system set standby_file_management='AUTO' scope=both sid='*';
alter system set DG_BROKER_CONFIG_FILE1='+DATA_SCAM08/s2_siebxd/dg1.dat' scope=both
 sid='*';
alter system set DG_BROKER_CONFIG_FILE2='+RECO_SCAM08/s2_siebxd/dg2.dat' scope=both
 sid='*';
shutdown immediate
EOF
```

8.2.1.10 Startup Standby nomount

```
sqlplus / as sysdba <<'EOF'
startup nomount force;
EOF
```

8.2.1.11 Instantiate the Standby Database

There are a number of different ways to instantiate the standby database from the primary. In this case the RMAN duplicate feature was used. As user oracle_siebel on scam02db03:

```
rman <<EOF!
connect target sys/welcome1@siebxd_dgmgrl
connect auxiliary sys/welcome1@s2_siebxd_dgmgrl
run {
allocate channel s1a type disk;
allocate channel s1b type disk;
allocate channel s1c type disk;
allocate channel s1d type disk;
allocate auxiliary channel s2 type disk;
duplicate target database for standby from active database; }
EOF!
```

8.2.1.12 Register the Database in CRS

```
srvctl add database -d s2_siebxd –n siebxd -o
  /u01/app/oracle_siebel/product/11.2.0.3/dbhome_siebel
  srvctl add instance -d s2_siebxd -i "siebxd1" -n scam08db03
  srvctl add instance -d s2_siebxd -i "siebxd2" -n scam08db04
srvctl modify database -d s2_siebxd -a "DATA_SCAM08,RECO_SCAM08"
```

8.2.1.13 Create Database Services

As user oracle_siebel on scam08db03:

```
srvctl add service -d s2_siebxd -s SIEB_PRIM -r "siebxd1,siebxd2" -e SELECT -m BASIC -P
  BASIC -w 5 -z 24 -j LONG -l PRIMARY -y AUTOMATIC
srvctl add service -d s2_siebxd -s SIEB_STBY -r "siebxd1,siebxd2" -e SELECT -m BASIC -P
  BASIC -w 5 -z 24 -j LONG -l PHYSICAL_STANDBY -y AUTOMATIC
srvctl add service -d s2_siebxd -s SIEB_STBY_TEST -r "siebxd1,siebxd2" -e SELECT -m
  BASIC -P BASIC -w 5 -z 24 -j LONG -l SNAPSHOT_STANDBY -y AUTOMATIC
```

The standby services were started and stopped on the primary so that the service definitions are created in the database and then synchronized to the standby. As user oracle_siebel on scam02db07:

```
srvctl start service -d siebxd -s SIEB_STBY
srvctl stop service -d siebxd -s SIEB_STBY
srvctl start service -d siebxd -s SIEB_STBY_TEST
srvctl stop service -d siebxd -s SIEB_STBY_TEST
```

8.2.1.14 Mount Remaining Standby Instances

As user oracle_siebel on scam08db03:

```
srvctl start database -d siebxd -o mount
```

8.2.1.15 Enable Flashback Database on Standby

As user oracle_siebel on scam08db03:

```
sqlplus / as sysdba <<EOF
alter database flashback on;
EOF
```

8.2.1.16 Configure and Start Data Guard Broker

On Primary and Standby as user oracle_siebel:

```
sqlplus/ as sysdba <<EOF
alter system set dg_broker_start=true scope=both sid='*';
EOF
```

On the primary, as user oracle_siebel:

```
dgmgrl sys/welcome1@dg_siebxd
create configuration config as primary database is siebxd connect identifier is
 dg_siebxd;
add database s2_siebxd as connect identifier is dg_s2_siebxd;
enable configuration;
```

8.2.1.17 Validate Standby Operation

As user oracle_siebel on any database server:

```
dgmgrl –silent sys/welcome1@dg_siebxd <<EOF
show configuration
show database siebxd
show database s2_siebxd
EOF
```

The following output was seen when Data Guard was operating normally:

```
Configuration - config

  Protection Mode: MaxPerformance
  Databases:
    siebxd    - Primary database
    s2_siebxd - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

Database - siebxd

  Role:            PRIMARY
  Intended State:  TRANSPORT-ON
  Instance(s):
    siebxd1
    siebxd2

Database Status:
SUCCESS

Database - s2_siebxd

  Role:            PHYSICAL STANDBY
  Intended State:  APPLY-ON
  Transport Lag:   0 seconds
  Apply Lag:       0 seconds
  Real Time Query: ON
  Instance(s):
    siebxd1
    siebxd2 (apply instance)

Database Status:
SUCCESS
```

8.2.1.18 Customize the Broker

As user oracle_siebel on any database server:

```
dgmgrl –silent sys/welcome1@dg_siebxd <<EOF
edit database s2_siebxd set property LogArchiveMaxProcesses=5;
edit database siebxd set property LogArchiveMaxProcesses=5;
edit database s2_siebxd set property StandbyFileManagement=auto;
edit database siebxd set property StandbyFileManagement=auto
EOF
```

8.2.1.19 Start the Database in Snapshot Standby Mode

The standby database was started in snapshot standby mode to complete and test the middle tier configuration. As user oracle_siebel on any database server:

```
dgmgrl sys/welcome1@dg_s2_siebxd <<EOF
convert database s2_siebxd to snapshot standby
EOF
```

The standby was periodically converted back to physical standby mode to apply any accumulated redo. As user oracle_siebel on any database server:

```
dgmgrl sys/welcome1@dg_s2_siebxd <<EOF
convert database s2_siebxd to physical standby
EOF
```

Once the site 2 configuration was completed the database was returned to physical standby mode.

8.2.2 Shared File System Creation

File systems were created on the scan04 ZFS Storage Appliance as follows:

PURPOSE	SHARE TYPE	EXPORTED AS	SITE STATE	COMMENTS
Siebel File System	Local Replicated	/export/siebelfs	Site 2 Primary	NFS v4, Constructed from site 1 replica on switchover/failover from site1
Site 2 Siebel File System Replica	Replica	/export/siebelfs_site1_replica	Site 1 Primary	NFS v4
Site 2 Test Siebel File System	Local	/export/siebelfs_test	Site 2 Setup, Site 2 Test	NFS v4, Copied from site 1 replica
Siebel Gateway Home	Local	/export/siebel_gateway_home	All	NFS v4, no replication
Grid OCR and Vote	Local	/export/siebel_gateway_cluster	All	NFS v3, no replication
Clusterware Home	Local	/export/ grid_scan04cn23	All	NFS v4, no replication
Siebel Server Home	Local	/export/siebel_scan04cn23	All	NFS v4, no replication
Clusterware Home	Local	/export/ grid_scan04cn24	All	NFS v4, no replication
Siebel Server Home	Local	/export/siebel_scan04cn24	All	NFS v4, no replication
Siebel Web Home	Local	/export/siebel_scan04cn21	All	NFS v4, no replication
Siebel Web Home	Local	/export/siebel_scan04cn22	All	NFS v4, no replication

8.2.3 Exalogic Bare Metal Servers

PURPOSE	HOSTNAME	ZFS STORAGE NETWORK ADDRESS	CLUSTER NETWORK ADDRESS
Gateway Node 1	scan04cn23	192.168.219.199	192.168.219.199
Gateway Node 2	scan04cn24	192.168.219.200	192.168.219.200
Siebel Server Node 1	scan04cn23	192.168.219.199	
Siebel Server Node 2	scan04cn24	192.168.219.200	
Siebel Web Server Node 1	scan04cn21	192.168.219.197	
Siebel Web Server Node 2	scan04cn22	192.168.219.198	

NIS Client Configuration was performed on all servers to enable the NFSv4 security model.

TCP Keepalive Parameters were configured on each Exalogic Server as follow:

```
net.ipv4.tcp_keepalive_time = 60
net.ipv4.tcp_keepalive_probes = 6
net.ipv4.tcp_keepalive_intvl = 10
```

8.2.4 Shared File Systems Mounted on Exalogic Servers

Shared file systems were mounted as follows:

PURPOSE	MOUNTED ON	EXPORTED AS (MOUNTED AS)	MOUNTED AS	SITE STATE	NFS
Siebel File System	Siebel and Gateway Servers	/export/siebelfs	/siebelfs	Site 2 Primary	v4
Siebel File System	Siebel and Gateway Servers	/export/siebelfs_test	/siebelfs	Site 2 Setup, Site 2 Test	v4
Site 2 Siebel File System Replica	Optional	/export/siebelfs_site2_replica	/siebelfs_site2_replica	Site 1 Primary	v4
Siebel Gateway Home	Local Shared	/export/siebel_gateway_home	/siebel_gateway_home	All	v4
Grid OCR and Vote	Local Shared	/export/siebel_gateway_cluster	/siebel_gateway_cluster	All	v3
Grid Home	scan04cn23	/export/ grid_scan04cn23	/u01/app/grid	All	v4
Siebel Server Home	scan04cn23	/export/siebel_scan04cn23	/u01/app/siebel	All	v4
Grid Home	scan04cn24	/export/ grid_scan04cn24	/u01/app/grid	All	v4
Siebel Server Home	scan04cn24	/export/siebel_scan04cn24	/u01/app/siebel	All	v4
Siebel Web Home	scan04cn21	/export/siebel_scan04cn21	/u01/app/siebel	All	v4
Siebel Web Home	scan04cn22	/export/siebel_scan04cn22	/u01/app/siebel	All	v4

NFS mount options were implemented as follows:

NFS VERSION	MOUNT OPTIONS
v3	nfs rw,bg,hard,nointr,rsize=32768,wsize=32768,tcp,noac,vers=3,timeo=600,actimeo=0
v4	nfs4 rw,bg,hard,nointr,rsize=131072,wsize=131072

8.2.5 File System Folders Created

The software folders were created the same as for site 1.

8.2.6 Database Client Software Installation and Configuration

The 32-bit Oracle database client software was installed on scan04cn23 and scan04cn24 in the same way as on site 1. Once the software was installed, the siebel user environment was configured as for site 1.

Connectivity to the database was configured as follows:

```
mkdir –p $ORACLE_HOME/network/admin
cat > $ORACLE_HOME/network/admin/tnsnames.ora <<EOF!
SIEBXD =
  (DESCRIPTION =
    (CONNECT_TIMEOUT=5)
    (TRANSPORT_CONNECT_TIMEOUT=3)
    (RETRY_COUNT=3)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS = (PROTOCOL = TCP)(HOST = scam08-scan3)(PORT = 1521)))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = sieb_stby_test))
  )
EOF!
```

8.2.7 Clustered Siebel Gateway Configuration

8.2.7.1 Gateway Server Installation and Configuration

The Gateway Server installation and configuration was performed on server scan04cn23 in the same way as on site 1.

8.2.7.2 Install Oracle Clusterware

Clusterware installation was performed on the scan04cn23 and scan04cn24 servers in the same way as on site 1, with virtual IP addresses allocated as follows:

PURPOSE	NAME	ADDRESS
Gateway Node1 Virtual Host Name		10.133.219.234
Gateway Node 2 Virtual Host Name		10.133.219.235
Siebel Gateway VIP		10.133.218.83
Cluster SCAN (temporary)	dummyscan	10.133.218.84

8.2.7.3 Install Oracle Clusterware Bundled Agents

Bundled Agents 2.1 installation and configuration was performed on scan04cn23 and scan04cn24 in the same way as on site 1.

8.2.8 Siebel Server Configuration

Siebel Server installation and configuration was performed on scan04cn23 and scan04cn24 in the same way as on site 1, with two exceptions:

- It was unnecessary to seed the Siebel File System because a copy from site 1 was used.
- The additional configuration outlined in this document was applied: MOS ID 1484925.1 "When running Siebel 8.x on Exalogic the sys and admin folder need to be moved to a local file system".

8.2.9 Siebel Web Server Configuration

Siebel Server installation and configuration was performed on scan04cn21 and scan04cn22 in the same way as on site 1.

8.2.10 Web Server Load Balancing Configuration

The F5 load balancer on site 2 was configured in the same way as site 1.

8.3 Site Test

8.3.1 Physical Standby Database converted to Snapshot Standby

Use to Data Guard Broker to convert the standby database to a snapshot standby, for example:

```
dgmgrl sys/welcome1@dg_s2_siebxd <<EOF
convert database siebxd to snapshot standby
EOF
```

The database service that was configured for snapshot standby mode (SIEB_STBY_TEST) will be started automatically.

8.3.2 Point the Siebel Processes to the Snapshot Standby Service

Edit the tnsnames.ora file on each Siebel Server and Gateway Server to point to the snapshot standby service (SIEB_STBY_TEST), for example:

```
SIEBXD =
  (DESCRIPTION =
    (CONNECT_TIMEOUT=5)
    (TRANSPORT_CONNECT_TIMEOUT=3)
    (RETRY_COUNT=3)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS = (PROTOCOL = TCP)(HOST = scam02-scan7)(PORT = 1521)))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = sieb_stby_test))
  )
```

Note, the same alias name (SIEBXD) is used as for primary operation and so the Siebel configuration does not need to be changed.

8.3.3 Create Clone of Siebel File System Replica on Standby Site

- Log into BUI on standby
- Select the REPLICA project, for example “scan04sn02: siebelfs”, and hit the "Replication" tab
- Hit the "Clone most recently received project snapshot" icon (labeled with the + sign)
- Enter the new project name, for example “siebelfs_test”
- Enter an override mount point “/export/siebelfs_test”, and hit CONTINUE
- Select the new LOCAL project - siebelfs_test

8.3.4 Mount the Siebel File System Clone

On each Siebel Server and Gateway Server, edit the /etc/fstab to point the cloned siebel file system and mount on the /siebelfs mount point, for example:

```
172.17.0.9:/export/siebelfs_test /siebelfs nfs4
 rw,bg,hard,nointr,rsize=131072,wsize=131072
```

As root, mount the file system:

```
mount /siebelfs
```

Note, the same mount point is used as for primary operation and so the Siebel configuration does not need to be changed.

8.3.5 Siebel Startup and Test

Start up the Siebel application using the regular process and then Siebel testing could begin.

8.4 Site Test to Standby

8.4.1 Shutdown Siebel

Shut down Siebel using the regular process.

8.4.2 Snapshot Standby Database Converted to Physical Standby

Use to Data Guard Broker to convert the snapshot standby database to a physical standby, for example:

```
dgmgrl sys/welcome1@dg_s2_siebxd <<EOF
convert database siebxd to physical standby
EOF
```

8.4.3 Point the Siebel Processes to the Primary Mode Service

Edit the tnsnames.ora file on each Siebel Server and Gateway Server to point to the primary mode service (SIEB_PRIM), for example:

```
SIEBXD =
  (DESCRIPTION =
    (CONNECT_TIMEOUT=5)
    (TRANSPORT_CONNECT_TIMEOUT=3)
    (RETRY_COUNT=3)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS = (PROTOCOL = TCP)(HOST = scam02-scan7)(PORT = 1521)))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = sieb_prim))
  )
```

Note, the same alias name (SIEBXD) does not change and so the Siebel configuration does not need to be changed.

8.4.4 Prepare to Mount the Primary Siebel File System

As root, unmount the cloned Siebel File System used for testing:

```
umount /siebelfs
```

So that we are ready for primary operation when necessary, edit the /etc/fstab to point the primary siebel file system on each Siebel Server and Gateway Server, for example:

```
172.17.0.9:/export/siebelfs /siebelfs nfs4 rw,bg,hard,nointr,rsize=131072,wsize=131072
```

Do not attempt to mount the file system at this time as it will not be available. Note, the mount point (/siebelfs) does not change and so the Siebel configuration does not need to be changed.

8.4.5 Remove the Clone of Siebel File System Replica on Standby Site

- Log into ZFS BUI on the standby site
- Select the LOCAL project, for example "siebelfs_test"
- Confirm that you have the correct project
- Hit the "Remove of Destroy Entry" trash can icon
- Hit OK to confirm

8.5 Switchover

8.5.1 Shutdown Siebel on Primary Site

Shutdown Siebel using the standard procedure and unmount the Siebel File System, for example as root on each Siebel Server and Gateway Server:

```
umount /siebelfs
```

8.5.2 Perform Database Switchover

Use Data Guard Broker to perform the database switchover, for example:

```
dgmgrl sys/welcome1@dg_s2_siebxd <<EOF
switchover to s2_siebxd
EOF
```

8.5.3 Stop Siebel File System Replication at Source

- Login to the ZFSSA BUI on the old primary (source) site.
- Locate the Siebel File System project, for example siebelfs.
- Navigate to the "Replication" tab and confirm that replication is up-to-date – the "Last Sync" time should be later than when the Siebel File System was dismounted.
- Click the "Enable/disable action" button to disable replication, and wait for the "STATUS" column to indicate a status of "disabled".

8.5.4 Perform Siebel File System Role Reversal at Target

- Login to the ZFSSA BUI on the new primary site.
- Locate the replica project on the standby (target) site, for example scan04sn01:siebelfs.
- Navigate to the Replication tab and confirm that replication is up-to-date – the "Last Sync" time should be later than when the Siebel File System was dismounted on the old primary site.
- Click the "Reverse Direction of Replication" button.
- Enter the new project name "siebelfs".

8.5.5 Mount the Siebel File System

This procedure should be performed on each Siebel Server and Gateway Server.

As root, make sure the current Siebel File System is not mounted:

```
umount /siebelfs
```

Check the /etc/fstab file to confirm that the server is mounting the Siebel File System from the primary export (/export/siebelfs), for example:

```
172.17.0.9:/export/siebelfs /siebelfs nfs4 rw,bg,hard,nointr,rsize=131072,wsize=131072
```

As root, mount the Siebel File System. Note, the mount point (/siebelfs) does not change and so the Siebel configuration does not need to be changed.

8.5.6 Startup Siebel as Prod on new primary site

Use the standard procedure to start Siebel.

8.5.7 Start Siebel File System Replication to New Standby Site

- Login to the ZFSSA BUI on the new primary site.
- Locate the Siebel File System project, for example siebelfs.
- Navigate to the Replication tab and click the "Edit Entry" button.
- Enable the "Send Updates: Continuous" radio button and hit the "Apply" button.
- Wait until the sync completes and the "Last Sync" time is updated.

8.5.8 Delete Old Siebel File System Project

It is important to delete the old Siebel File System project after the switchover so that a subsequent switchover or failover will not be slowed down by this work. To clean up:

- Login to the ZFSSA BUI on the old primary (new standby) site.
- Locate the Siebel File System project, for example siebelfs.
- Confirm that there are no shares in this project, and then delete the project.

8.6 Failover

In the event that the primary site is down a failover is performed so that the application can be started on the secondary site. After the failover has been completed it is important to reestablish a standby site as quickly as possible, either by performing the Reinstate procedure – Section 8.7 Reinstate Standby, or by establishing a new secondary site – Section 8.2 Secondary Site Setup.

8.6.1 Perform Database Failover

This activity can be performed in parallel with the Siebel File System Role Reversal. Using Data Guard Broker the database switchover was performed, for example:

```
dgmgrl sys/welcome1@dg_siebxd <<EOF
failover to siebxd
EOF
```

8.6.2 Perform Siebel File System Role Reversal on Replica

This activity could be performed in parallel with the database failover. The following steps were performed:

- Login to the ZFSSA BUI on the new primary site.
- Locate the replica project on the standby (target) site, for example scan04sn01:siebelfs.
- Navigate to the Replication tab and make a note of the "Last Sync" time.
- Click the "Reverse Direction of Replication" button.
- Enter the new project name "siebelfs".

8.6.3 Mount the Siebel File System

This procedure was performed on each Siebel Server and Gateway Server.

As root, make sure the current Siebel File System is not mounted.

Check the /etc/fstab file to confirm that the server is mounting the Siebel File System from the primary export (/export/siebelfs), for example:

```
172.17.0.9:/export/siebelfs /siebelfs nfs4 rw,bg,hard,nointr,rsize=131072,wsize=131072
```

As root, mount the Siebel File System. Note, the mount point (/siebelfs) does not change and so the Siebel configuration does not need to be changed.

8.6.4 Startup Siebel as Prod on new primary site

Use the standard procedure to start Siebel.

8.7 Reinstate Standby

8.7.1 Perform database reinstate

Startup one database instance on the new standby (old primary) site:

```
srvctl start instance –d s2_siebxd –i siebxd1
```

Use Data Guard Broker to reinstate the old primary as a physical standby database:

```
dgmgrl sys/welcome1@dg_siebxd <<EOF
reinstate database s2_siebxd
EOF
```

8.7.2 Start Siebel File System Replication to New Standby Site

- Login to the ZFSSA BUI on the new primary site.
- Locate the Siebel File System project, for example siebelfs.
- Navigate to the Replication tab and click the "Edit Entry" button.
- Enable the "Send Updates: Continuous" radio button and hit the "Apply" button.
- Wait until the sync completes and the "Last Sync" time is updated.

8.7.3 Delete Old Siebel File System Project

It is important to delete the old Siebel File System project after the switchover so that a subsequent switchover or failover will not be slowed down by this clutter. To clean up:

- Login to the ZFSSA BUI on the new standby site.
- Locate the Siebel File System project, for example siebelfs.
- Confirm that there are no shares in this project.
- Delete the project.

Siebel MAA with Case Study on Exalogic and Exadata
October 9, 2013

Author:
Richard Exley (Oracle)

Contributing Authors:
Chris Akker (F5),
Lyn Pratt (Oracle),
Mathew Steinberg (Oracle)

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com