Oracle Maximum
Availability Architecture

An Oracle White Paper
February 2014

PeopleSoft MAA 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) at http://www.oracle.com/goto/maa.

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

In this paper we provide:

- An introduction to Oracle Engineered Systems
- The PeopleSoft Maximum Availability Architecture – a high-level description of the architecture and key technology components
- PeopleSoft MAA Case Study on Exalogic and Exadata – how the MAA architecture was established on our target systems
- Operational Procedures – how to operate the system in the event of planned and unplanned outages
- Outage testing and results – our tests to validate our best practices and to measure downtime in various outage scenarios, and the impact on PeopleSoft users
- Appendix – details of how PeopleSoft MAA was implemented on the Exalogic and Exadata environment

In summary, our test results show the following:

- PeopleSoft end users have minimal to no impact for most unplanned failures.

- Application server domain failures result in at most a 41 second wait with no user errors.

- Database Oracle RAC node failure results in at most 31 second wait with no user errors.

- With multiple Pure Internet Architecture (PIA) web servers, failures at this layer can be minimized.

2 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 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 high redundancy architecture with fault tolerance and zero-down-time maintenance.

2.2 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. Because Exadata is an integrated system, it offers superior price-performance, availability and supportability. Exadata frees users from the need to build, test and maintain systems and allows them to focus on higher value business problems.

Exadata uses a scale out architecture for database servers and storage. This architecture maintains an optimal storage hierarchy from memory to flash to disk. Smart Scan query offload has been added to the storage cells to offload database processing. Exadata implements Smart Flash Cache as part of the storage hierarchy. Exadata software determines how and when to use the Flash storage for reads and write as well as how best to incorporate Flash into the database as part of a coordinated data caching strategy. A high-bandwidth low-latency InfiniBand network running specialized database networking protocols connects all the components inside an Exadata Database Machine. In addition to a high performance architecture and design, Exadata offers the industry's best data compression to provide a dramatic reduction in storage needs.

3 PeopleSoft Maximum Availability Architecture Overview

PeopleSoft Maximum Availability Architecture (MAA) is a PeopleSoft high availability architecture layered on top of the Oracle Database and Oracle Fusion Middleware Maximum Availability Architectures, 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 and Oracle Fusion Middleware Maximum Availability Architectures, then we will describe how to provide high availability for the PeopleSoft application on top of that foundation, resulting in a full PeopleSoft MAA implementation.

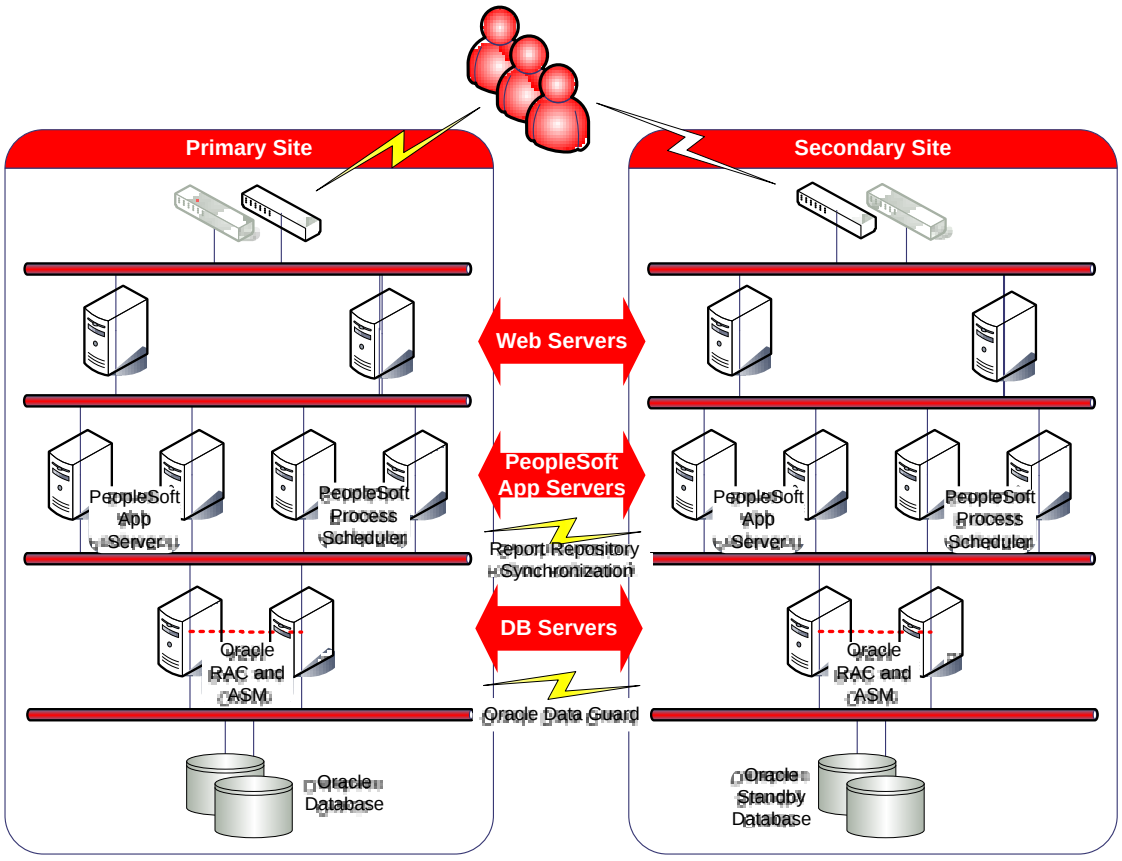

Figure 1. PeopleSoft Maximum Availability Architecture

3.1 Oracle Database Maximum Availability Architecture

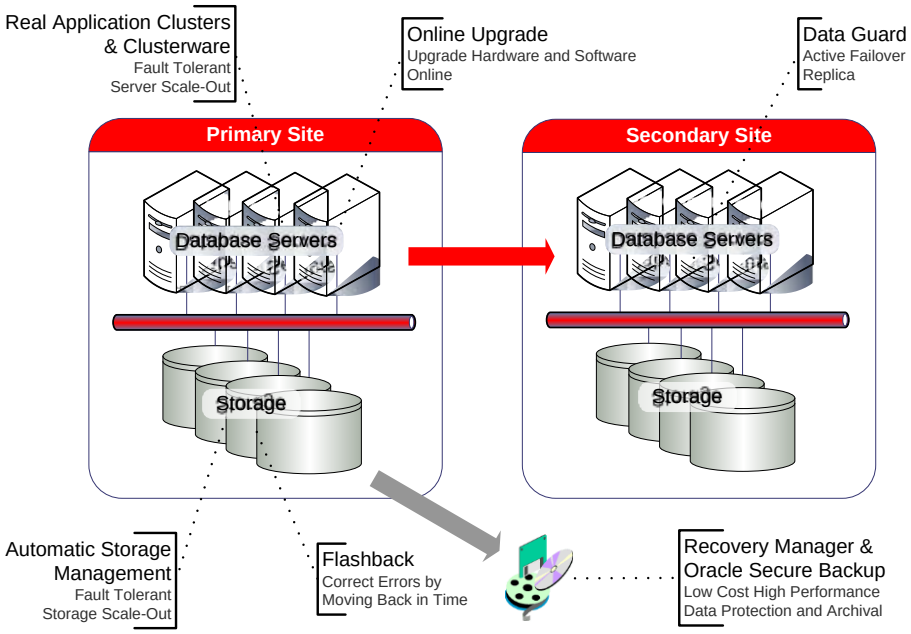

Figure 2. Oracle Database Maximum Availability Architecture

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

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

The rest of this section briefly describes each of these components. 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 and Oracle Clusterware

Oracle Real Application Clusters (Oracle 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*.

Oracle Clusterware is a cluster manager that is designed specifically for the Oracle Database. In an Oracle RAC environment, Oracle Clusterware monitors all Oracle resources (such as database instances and listeners). If a failure occurs, Oracle Clusterware will automatically attempt to restart the failed resource. During outages, Oracle Clusterware relocates the processing performed by the inoperative resource to a backup resource. For example, if a node fails, Oracle Clusterware will relocate database services being used by the application onto a surviving node in the cluster. See also: *Oracle Clusterware Administration and Deployment Guide*.

3.1.2 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.

PeopleSoft supports both physical and logical standby databases. See also: *Oracle Data Guard Concepts and Administration*.

- 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, using Redo Apply, which recovers the redo data received from the primary database and applies the redo to the physical standby database.
- 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.

It is possible (and ideal) 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 PeopleSoft Servers continue running, 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 a local and remote standby be deployed for maximum availability. For further details please see the Oracle MAA white paper "Database Rolling Upgrades Made Easy by Using a Data Guard Physical Standby Database."

3.1.2.1 Physical Standby Features

Oracle Active Data Guard is a physical standby database that receives and applies redo while it is open for read-only access, and so it may be used for other purposes as well as disaster recovery.

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 process, and return to its function as a physical standby database once the upgrade is complete.

3.1.2.2 Logical Standby Features

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.

3.1.3 Oracle Flashback

Oracle Flashback quickly rewinds an Oracle database, table or transaction to a previous point in 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 rapidly 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 "Oracle Flashback Technology" in *Oracle Database Concepts* for more information.

3.1.4 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.

3.1.5 Oracle Recovery Manager and Oracle Secure Backup

Oracle Recovery Manager (RMAN) is an Oracle Database utility that can back up, restore, and recover database files. It is a feature of 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.

Oracle Secure Backup 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, Oracle Secure Backup provides a complete tape backup solution for your IT environment. Oracle Secure Backup is tightly integrated with RMAN to provide the media management layer for RMAN.

3.1.6 PeopleSoft Database Configuration Best Practices

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

3.1.6.1 Add the PeopleSoft Database to Cluster Ready Services

If not done already, add the PeopleSoft database and Oracle RAC instances to Cluster Ready Services (CRS). The following serve as examples:

```
srvctl add database -d PAYROLL_S1 -o
/u01/app/oracle_psft/product/11.2.0.3/dbhome_psft -p
+DATA_SCAM08/PAYROL/spfilePAYROLL.ora -n PAYROLL -a "DATA_SCAM08,RECO_SCAM08"

srvctl add instance -d PAYROLL_S1 -i PAYROLL1 -n scam08db03

srvctl add instance -d PAYROLL_S1 -i PAYROLL2 -n scam08db04
```

3.1.6.2 Create FAN Enabled Role Based Database Services

PeopleSoft supports Fast Application Notification (FAN). When an Oracle RAC database instance fails, the recovering instance sends an INSTANCE DOWN event to all clients that were connected to the failed instance. The clients then break their current TCP connections and perform Transparent Application Failover (TAF).

Role-based services are created and used by the PeopleSoft application to connect to the database. These role based services are started based on the database role. See the table below.

SERVICE NAME	DATABASE ROLE	PURPOSE
PAYROLL_ONLINE	PRIMARY	Online HRMS service
PAYROL_BATCH	PRIMARY	Batch processing service
PSQUERY	PHYSICAL STANDBY	Offload queries to Oracle Active Data Guard physical standby database

The following are examples of creating role-based services:

```
srvctl add service -d PAYROLL_S1 -s PAYROLL_ONLINE -r "PAYROLL1,PAYROLL2" -m
BASIC -e SELECT -q TRUE -l PRIMARY -z 180 -w 1

srvctl add service -d PAYROLL_S1 -s PAYROLL_BATCH -r "PAYROLL1,PAYROLL2" -m
BASIC -e SELECT -q TRUE -l PRIMARY -z 180 -w 1
```

If the primary database becomes the standby, PSQUERY is started for Oracle Active Data Guard query access.

```
srvctl add service -d PAYROLL_S1 -s PSQUERY -r "PAYROLL1,PAYROLL2" -m BASIC
-e SELECT -q TRUE -l PHYSICAL_STANDBY -z 180 -w 1
```

Using the example for the service PAYROLL_ONLINE:

PARAMETER	VALUE	DESCRIPTION
-d	PAYRLLL_$1	The database unique name
-s	PAYROLL_ONLINE	The database service name
-r	"PAYROLL1,PAYROLL2"	The preferred instances the service should be started on
-m	BASIC	TAF failover method
-e	SELECT	TAF failover type
-q	TRUE	High Availability FAN events
-l	PRIMARY	The database role in which the service should be started
-z	180	The number of connection retries before the connection fails
-w	1	Time interval in seconds between each connection retry attempt

3.1.6.3 Configure HugePages (Linux Database Server Only)

PeopleSoft will typically run with many database connections and a large SGA; therefore, configuring HugePages for the PeopleSoft 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 so do not configure more than is required.

My Oracle Support note 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 Manager (AMM) cannot be used in conjunction with huge pages and so the MEMORY_TARGET and MEMORY_MAX_TARGET parameters should be unset for each database instance. See My Oracle Support note 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 My Oracle Support note 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*g* Release 2 (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 case, 4609 HugePages were used.

3.1.6.4 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. PeopleSoft 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;
```

If passwords are not managed, a non-fatal error message ORA-28002, will be emitted PASSWORD_GRACE_TIME days prior to when the password will expire. The default is 7 days in the DEFAULT profile. The PeopleSoft application server however, sees this as a fatal error and sessions will begin to fail. The application server will fail to start but the password has not yet expired. The error message will show up in the APPSRV.LOG file. This will most likely impact all database schemas that PeopleSoft uses: PS, PEOPLE and the owning schema of the PeopleSoft application objects created at install time. If you encounter this error, you will need to know or have access to the passwords for all of the schema users and reset them as follows (do not change the passwords):

```
alter user <user name> identified by <use the same password>;
```

For example:

```
alter user PEOPLE identified by <use the same password>;
```

Please refer to "Configuring Password Protection" in the *Oracle Database Security Guide.*

3.1.6.5 Use Preallocated Flashback Logs to Improve Performance

It is recommended that you enable Flashback Database so that the database can be "rewound" or flashed back to a point in time without performing a restore from backup. It also allows the database to be re-instantiated as a standby should a failover event occur.

Flashback logs are maintained in the Fast Recovery Area (FRA) and are created with a default size based on the percentage of buffer cache size in the SGA. These flashback logs are created until the

flashback retention time is reached at which point, the flashback logs may be reused. A performance impact might be noticeable before reaching steady state. To address this, it is possible to pre-allocate flashback logs – the minimum size of a single flashback log and the total space that limits how much space the flashback logs can use. In Oracle Database 11*g* Release 2 (11.2.0.3) of the database, our case study used the following parmaters to pre-allocate the flashback logs.

```
_db_flashback_log_min_size=8G

_db_flashback_log_min_total_space=120G
```

It is recommended that you study the total space used by flashback logs before setting the above parameters. Please see My Oracle Support note 565535.1 "Flashback Database Best Practices & Performance" for further details.

3.1.6.6 Configure Dead Connection Detection

When a PeopleSoft 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 My Oracle Support note 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.

3.1.6.7 Reduce Timeout on Oracle RAC Node Failure (Exadata Only)

On Exadata, it is possible to failover 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 PeopleSoft High Availability Architecture

The following diagram illustrates a PeopleSoft high availability architecture.

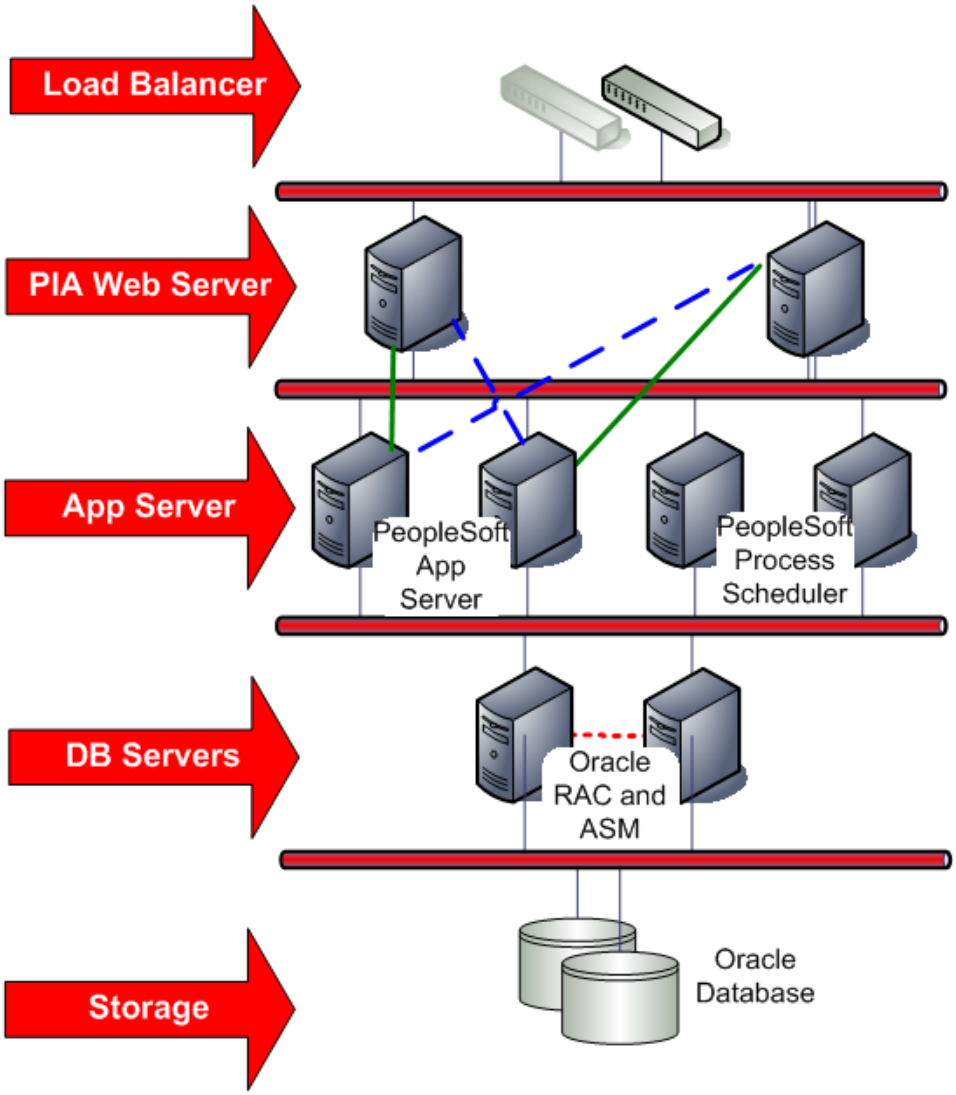

Figure 3. PeopleSoft High Availability Architecture

In this section we discuss the high availability deployment of the PeopleSoft application software that is layered on top of the Oracle Database MAA foundation.

3.2.1 PeopleSoft Application Software High Availability Deployment

PeopleSoft application components can each be deployed in a highly available manner. 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. We recommend the servers have adequate capacity to run peak load even when one server is down.

- A load balancer is used to balance web traffic across the web servers. The hardware load balancer has dual switches for redundancy.
- Two or more PeopleSoft Pure Internet Architecture (PIA) web servers for workload distribution and redundancy. Web server sessions accumulate state; thus their routings are "sticky" for a session – once a user is routed to a web server, all future requests for that session are routed to the same web server. If the web server fails the user will be routed to a new web server, but will have to re-authenticate and re-start their uncommitted work.

  As depicted in the above diagram, each PIA web server connects to a primary and secondary (active/passive) pair of application domain servers. The solid line depicts the active connection while the dashed line shows the secondary passive connection. Should a primary application domain server become lost, its requests will be routed to its secondary application domain server. Since the PIA web server does not maintain simultaneous connections to both its primary and secondary application domain server, a delay can be observed if the node hosting the primary application domain becomes unavailable and the PIA web server establishes its secondary connection.

  While the above illustration shows each PIA web server connecting to a pair of application domain servers, they can connect to any number of active and passive application domain servers.
- Any number of application domain servers can be configured to service various requests. It is at this layer where the bulk of the business logic is executed. As there is no session state at this level, loss of an application domain server does not result in a need for user rework. The application domain servers connect to the database using role-based services.
- A pair of PeopleSoft Batch Process Schedulers can be configured as master, with one being active and one idle. Any number of "slave" Batch Process Schedulers can be configured. If the active master Batch Process Scheduler goes down, the idle master takes over the task of assigning jobs to the slave process schedulers. If both go down, the slave process schedulers become stand-alone, doing the work already assigned to them, but not assuming the role of master. The process schedulers connect to the database using role-based services.

3.2.2 PeopleSoft Application File System Layout and Deployment

PeopleSoft HRMS 9.1 FP3 and PeopleTools 8.52.07 is implemented in the case study described in the next section. The PeopleSoft applications and infrastructure software components can be deployed in two different ways that affect how the system will be managed. Before describing the deployment options, a few terms are defined here:

PS_HOME: An environment variable that defines the file system location in which the PeopleTools software is installed.

PS_APP_HOME: An environment variable that defines the file system location in which the PeopleSoft application (HRMS, FIN, EPM, etc.) is installed. It is common to have installed the application into the PS_HOME (PeopleTools) location as it was not until PeopleTools 8.52 and later

when the application could be installed into a separate location. It is recommended that you install the application in a separate location with later releases of PeopleTools.

PS_CFG_HOME: An environment variable that defines the file system location for the application server domains, web server domains, their respective configuration files, and log files.

COBDIR: An environment variable that defines the file system location in which the Micro Focus Server Express COBOL compiler and run-time libraries are stored. It is required that Micro Focus Server Express be installed on all servers that will run COBOL programs. This component cannot be shared due to license key restrictions.

The PeopleSoft software can be installed in one of the following ways:

Local Homes: A deployment paradigm where all of the PeopleSoft software and its required infrastructure components such as Oracle JRockit, Oracle WebLogic Server, Oracle Tuxedo, and the PeopleSoft applications are installed on each server that will host PeopleSoft.

Shared Homes: A deployment paradigm in which all of the PeopleSoft software and its required infrastructure components such as Oracle JRockit, Oracle WebLogic Server, Oracle Tuxedo, and the PeopleSoft applications are installed in a single shared file system location that all nodes in the deployment can access. In this deployment option, the PS_HOME environment variable on all nodes point to the same file system directory location. Alternatively, you can install the PeopleSoft application in a shared location and set the PS_APP_HOME environment variable on all nodes to the shared file system location.

The following table outlines the advantages and disadvantages of shared or local homes.

TYPE	ADVANTAGES	DISADVANTAGES
Local Homes	Isolate maintenance to a single or group of PeopleSoft server nodes. Isolate diagnostics to a single or group of nodes which might require patching without affecting other nodes or requiring out of place patching.	Requires PeopleSoft software to be installed on every node which prolongs the install time. Requires patching and software upgrades to be performed on each individual nodes prolonging maintenance time. While some of the patch installs can be performed in parallel, there are components that require user interaction to respond to prompts preventing the entire process from being performed in parallel.
Shared Homes	Installation of PeopleSoft software is performed once on one node; all other nodes share the installation reducing install time. Significantly reduces setup and deployment time of the secondary DR site if the shared homes are replicated. No installation (except for COBOL) is required at the secondary site. Out-of-place patching and upgrades staged and performed once on a single node; all other nodes are restarted reducing maintenance	In-place patching and upgrades can impact all nodes. Loss of the shared storage will result in an application outage.

	time.*[1]	

As discussed in the next section, it is recommended that you deploy PeopleSoft using the shared home paradigm.

In addition to the software installation shared homes, PeopleSoft generates reports and writes them to the PeopleSoft Report Repository. This is a file system directory location where the batch jobs and the application engine place output logs and reports. The report repository is also shared across all application and web servers at the primary site.

3.2.3 Replication of PeopleSoft Shared Home and Report Repository

Data loss is not limited to only the database. Most applications have inbound and outbound data feeds, reports, etc. PeopleSoft HRMS Payroll generates pay slips in the form of files that are sent to, for example, Automated Data Processing (ADP) for electronic check processing and deposits. Should a site failure occur, if the files were not replicated, a loss of inbound and outbound files will occur and a gap in which files were sent (or not sent) must be resolved. Oracle ZFS Storage Appliance provides a replication service that can be used to minimize loss of the files needed by the application tier as discussed later in this paper.

3.2.4 PeopleSoft Application and Web Tier Configuration Best Practices

We recommend the following for configuring the PeopleSoft application:

3.2.4.1 Database Connection for Application Domain Servers

The TNS connect strings used by the application domain servers should connect to the database using the role-based database services described above. Here are examples:

```
# HR Online users
HR91FP3=
(DESCRIPTION_LIST =
  (FAILOVER=on)
  (DESCRIPTION =
    (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS=(PROTOCOL=tcp)(HOST=scam08-scan3.us.oracle.com)(PORT=1521)))
      (CONNECT_DATA=
         (SERVER=DEDICATED)
```

[1] Upgrades may require the application server domains and web server domains to be re-deployed on each node. Even so, this takes less time than performing software installs on each node.

```
         (SERVICE_NAME=PAYROLL_ONLINE)))
     )

# Batch scheduler
HRBATCH=
(DESCRIPTION_LIST =
  (FAILOVER=on)
  (DESCRIPTION =
    (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS=(PROTOCOL=tcp)(HOST=scam08-scan3.us.oracle.com)(PORT=1521)))
      (CONNECT_DATA=
         (SERVER=DEDICATED)
         (SERVICE_NAME=PAYROLL_BATCH)))
)


# Active Data Guard
PSFTADG2=
(DESCRIPTION_LIST =
  (FAILOVER=on)
  (DESCRIPTION =
    (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS=(PROTOCOL=tcp)(HOST=scam02-scan7.us.oracle.com)(PORT=1521))
      (ADDRESS=(PROTOCOL=tcp)(HOST=scam08-scan3.us.oracle.com)(PORT=1521)))
      (CONNECT_DATA=
         (SERVER=DEDICATED)
         (SERVICE_NAME=PSQUERY)))
     )
```

3.2.4.2 Configure the PS.PSDBOWNER Table

PeopleSoft treats the TNS connect string alias as the actual name of the database regardless of what the database unique name is. The PeopleTools layer checks to see if the database it is connecting to matches the "DBName" of its server domain configuration. The DBName must be set to the TNS connect string alias name. The TNS connect string alias name must be inserted into the PS.PSDBOWNER table along with the name of the PeopleSoft schema name that owns all of the PeopleSoft objects. Without these entries, the application domain server will not start.

The rows in the PS.PSDBOWNER table will be similar to the following:

```
DBNAME   OWNERID
--------------   ----------------
HR91FP3  HR91FP3
HRBATCH  HR91FP3
PSFTADG2 PSFTADG2
```

Further discussion on this topic can be found in Appendix section 10.2.5.

3.2.4.3 Ensure All Log Files Are Not Written to a Shared PS_HOME

Make sure that the application domain servers and the PIA web servers write to PS_CFG_HOME instead of PS_HOME. This ensures that no errors or loss of log files occur should the shared PS_HOME become a read-only file system after a switchover or role reversal.

3.2.4.4 Configure PeopleSoft Report Repository

Use a shared file system for the PeopleSoft report repository. Place the log_output directory in the shared report repository. Also, ensure that the PIA web servers can access the same shared file system containing the report repository. See Appendix sections 9.4.6 for further details.

3.3 PeopleSoft MAA Site State Model and State Transitions

Figure 4 shows 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 have a specific configuration in each state and there is a set of documented steps to move from one state to the next.

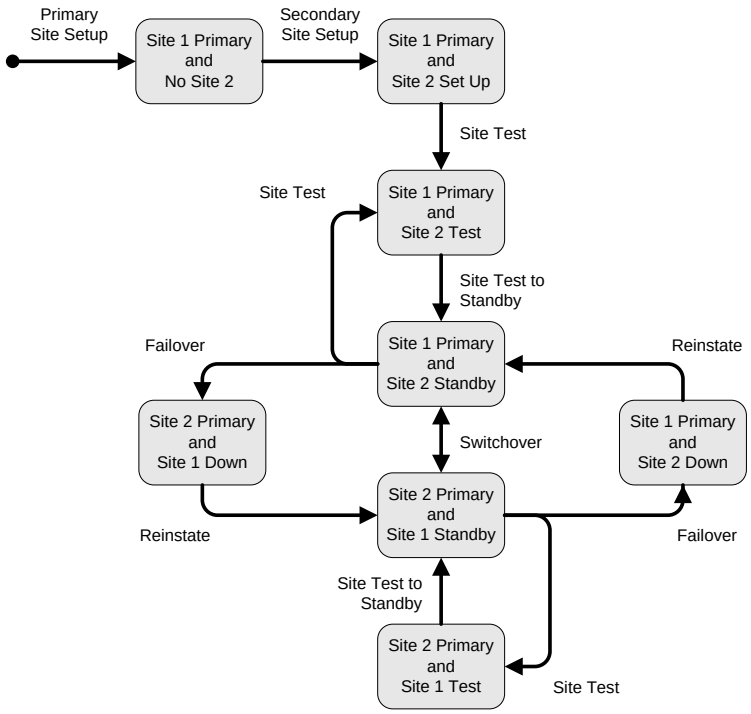

Figure 4. PeopleSoft MAA Site State Model and State Transitions

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

TABLE 1. 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 system databases and file systems are configured in each state.

SITE STATE	PEOPLESOFT DATABASE - DATA GUARD	PEOPLESOFT SHARED HOMES AND REPORT REPOSITORY - 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, (Oracle Active Data Guard) during setup.	Site 1 primary with continuous replication to site 2. Site 2 snapshot 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 snapshot created for test.
Site 1 Primary and Site 2 Standby	Site 1 primary and site 2 physical standby (Oracle Active Data Guard).	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 (Oracle Active Data Guard).	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 snapshot created for test.

PeopleSoft PeopleTools versions 8.52 and later can optionally support Oracle Active Data Guard if the PeopleSoft application domain is configured with a "secondary" PeopleSoft access ID. The above table denotes this with "(Oracle Active Data Guard)."

3.4 Planned and Unplanned Outage Solutions

This section summarizes the outages that may occur in a PeopleSoft environment and the Oracle solution that is used to minimize application downtime. In all cases, we are focused on PeopleSoft Application downtime as perceived by the end user, not the downtime of the individual component.

3.4.1 Unplanned Outage Solutions

Table 2 describes the unplanned outages that may be caused by system or human failures in a PeopleSoft environment and the technology solutions that would be used to recover and keep downtime to a minimum.

TABLE 2. UNPLANNED OUTAGE SOLUTIONS

OUTAGE TYPE	ORACLE SOLUTION	BENEFITS	RECOVERY TIME
PeopleSoft PIA Web Server Node or Component Failure	Load Balancing	Surviving nodes pick up the slack	Affected users re-authenticate and resubmit work
	Redundant Web Servers	Surviving nodes continue processing	No downtime
PeopleSoft Application Domain Server Node or Component Failure	PIA servers configured with active/passive connections Redundant application domain servers	Surviving nodes pick up the slack	No downtime
Database Node or Instance Failure	Oracle RAC	Automatic recovery of failed nodes and instances FAN events, transparent application and service failover	Users transparently fail over Updates may need to be re-submitted
Site Failure	Data Guard	Fast Start Failover	< 5 minutes
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	< 5 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
	Oracle Active Data Guard	Automatically detects and repairs corrupted blocks using the physical standby database	No downtime, transparent to application
	Data Guard	Automatic validation of redo blocks before they are applied, fast failover to an uncorrupted standby database	Seconds to 5 minutes

NOTE: The recommended solution for site failure is Data Guard with Data Guard Fast-Start Failover. As long as the standby database apply lag is within the fast start failover Lag limit, then the time to bring up the DR site will depend on the fast start failover timout threshold, the time to fail over the standby to a primary role, and the time to start all of the PeopleSoft application and PIA web servers. Our case study testing shows the following timings:

- 30 seconds to reach Fast-Start Failover timeout threshold
- 35 seconds to complete standby database failover to the primary role
- 56 seconds to start (in parallel) all PeopleSoft application domains and PIA web servers

It is possible to have a site failover to take a few minutes. In the above case it is 91 seconds.

For storage failure, if on-disk backups are kept on a separate storage array such as an Oracle ZFS Storage Appliance, it is possible to have the restore take minutes to complete.

3.4.2 Planned Maintenance Solutions

Table 3 summarizes the planned maintenance activities that typically occur in a PeopleSoft environment and the recommended technology solutions to keep downtime to a minimum.

TABLE 3. PLANNED OUTAGE SOLUTIONS

MAINTENANCE ACTIVITY	SOLUTION	PEOPLESOFT OUTAGE
Mid-Tier operating system or hardware upgrade	Hardware Load balancing, Redundance services across Web and Tuxedo Application Servers	No downtime
PeopleSoft application patching (application tier only)	PeopleSoft out-of-place patching	Minutes to hours
PeopleSoft application configuration change	PeopleSoft application rolling restart	Minutes
PeopleSoft upgrades	PeopleSoft out-of-place upgrades	Hours to days (depending on database size)[2]
Database tier operating system or hardware upgrade	Oracle RAC	No downtime
Oracle Database interim patching	Oracle RAC rolling apply, Standby-First	No downtime
Oracle Database 11*g* online patching	Online patching	No downtime
Oracle Grid and Clusterware upgrade and patches	Rolling apply / upgrade	No downtime

[2] In practice, 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 can help you plan and execute the upgrade.

Database storage migration	Oracle ASM	No downtime
Migrating to Oracle ASM or migrating a single-instance database to Oracle RAC	Data Guard	Seconds to minutes
Patch set and database upgrades	Release 11.1 or greater: Data Guard transient logical rolling upgrade	Seconds to minutes

4 PeopleSoft MAA Case Study on Exalogic and Exadata

PeopleSoft HR 9.1 FP3 with PeopleTools 8.52.07 was deployed in a full PeopleSoft MAA configuration on X3-2 Oracle Exalogic and quarter rack X3-2 Oracle Exadata Database Machines. Figure 5 is a high level illustration of the configured system:

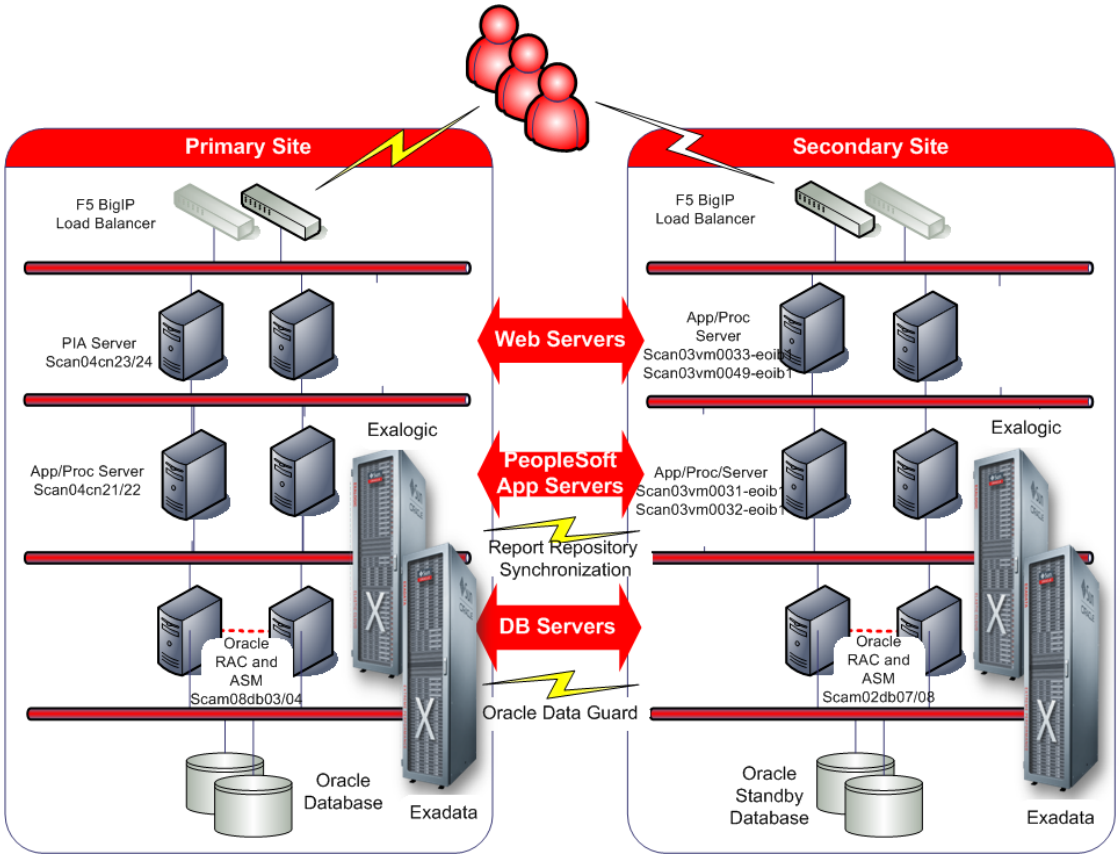

Figure 5. PeopleSoft full MAA implementation

Figure 5 illustrates a primary and a secondary site. Each site is configured such that each can assume the primary or standby role whether the switch was planned or unplanned. At each site, the web, application, and batch servers and the middle tier file system reside in the Oracle Exalogic machine; the database servers and storage reside in the Oracle Exadata Database Machine.

4.1 Hardware

4.1.1 Primary Site

The primary site has the following hardware:

TIER	HARDWARE
Database Tier	Quarter Rack X3-2 Oracle Exadata Database Machine • Two Compute nodes – scam08db03, scam08db04 o 16 cores (32 logical cores with SMT), 128GB RAM each • Three Exadata storage cells o 24 cores, 1.48TB flash cache, high performance disks each
Application Tier	X3-2 Oracle Exalogic bare metal (non-virtualized) • Two compute nodes – scan04cn21, scan04cn22 o 16 cores (32 logical cores with SMT), 256 GB RAM each o Hosting PeopleSoft Application and Process servers
Web Tier	X3-2 Oracle Exalogic bare metal (non-virtualized) • Two compute nodes -- scan04cn23, scan04cn24 o 16 cores (32 logical cores with SMT), 256 GB RAM each o Hosting PeopleSoft Pure Internet Architecture (PIA) web servers
Middle Tier File System Storage	Sun ZFS Storage 7320 for application and web server storage – scan04sn-fe

4.1.2 Secondary Site

The secondary site has the following hardware:

TIER	HARDWARE
Database Tier	Quarter Rack X3-2 Oracle Exadata Database Machine • Two Compute nodes – scam02db07, scam02db08 o 16 cores (32 SMT), 128GB RAM each • Three Exadata storage cells o 24 cores, 1.48TB flash cache, high performance disks each
Application Tier	X3-2 Oracle Exalogic (virtualized) • Two compute nodes – scan03vm0031-eoib1, scan03vm0032-eoib1 o 8 cores, 8 GB RAM each o Hosting PeopleSoft Application and Process servers
Web Tier	X3-2 Oracle Exalogic (virtualized)

	• Two compute nodes -- scan03vm0033-eoib1, scan03vm0049-eoib1 o 8 cores, 8 GB RAM each o Hosting PeopleSoft Pure Internet Architecture (PIA) web servers
Middle Tier File System Storage	Sun ZFS Storage 7320 for application and web server storage – scan03sn-fe

You will notice that the Oracle Exalogic machine configuration is not symmetric between the primary and secondary site. This is because the primary is configured as 4 shared bare metal compute nodes while the secondary site uses vServers dedicated specifically to PeopleSoft.

4.1.3 F5 Networks BIG-IP Local Traffic Manager

F5 BIG-IP hardware load balancers at each site are used for distributing traffic across PeopleSoft PIA web servers. The BIG-IP Local Traffic Manager (LTM) was configured at each site. The health of each web server within the BIG-IP server pool is monitored at the TCP layer as well as at the application layer. The application level monitor is a user-defined monitor to determine the health of the PeopleSoft web servers. The combination of the built-in TCP monitor provided by F5 Networks and the user-defined health monitor minimizes the impact on users should a web server node fail or if the web server is stalled as discussed later.

4.2 Software and Versions

The following software and versions were used for the case study:

SOFTWARE	VERSION
Oracle Exalogic	ECC 2.0.3.0.0 (site 1) ECC 2.0.4.0.0 (site 2)
PeopleSoft HRMS	9.1 FP3
PeopleSoft PeopleTools	8.52.07
Oracle Database	11.2.0.3 BP20
Oracle Exadata Storage Software	11.2.3.2.1
Oracle Database Client	11.2.0.3
Oracle Clusterware Grid Infrastructure	11.2.0.3 BP20
Sun ZFS 7320 Storage	2011.04.24.5.0,1-1.33 (Site 1) 2011.04.24.6.0,1-1.36 (Site 2)

4.3 Database Setup

Whether created afresh or migrated to the new infrastructure, the PeopleSoft database should be configured following the PeopleBooks documentation as well as implementing the Oracle Exadata Database Machine best practices. For this case study, the PeopleSoft database for HRMS 9.1 FP3 was copied from a test system onto Oracle Exadata Database Machine using the Recovery Manager

(RMAN) duplicate feature. Once there, Oracle Exadata Database Machine best practices for OLTP applications were implemented. For full details of these best practices, please see My Oracle Support note 1067527.1.

The standard Exadata configuration was deployed at both the primary and secondary sites. This includes the following:

- ASM disk groups (DATA and RECO) with NORMAL redundancy
- PeopleSoft database configured with Oracle RAC across both nodes of the X3-2 quarter rack Exadata Database Machine
- Database services registered in the Oracle Cluster Ready Services and in the SCAN listener
- For connection load balancing, the pre-configured SCAN listener is used

4.3.1 Database Features for the DR Site

The traditional role of a DR site has typically been passive – hardware and software that sits idle until it is needed. While a DR site serves to protect against data loss due to a site failure, it is a large investment for this one critical benefit. Oracle technology has evolved to leverage that investment of the DR site to further increase return on investment (ROI) and to utilize what would otherwise be idle hardware. The past several years have provided the following to achieve this benefit:

- **Oracle Active Data Guard** allows for off-loading queries for reporting, decision support, OLAP and ad-hoc queries to the physical standby database leaving more capacity for the production OLTP system. PeopleSoft, as of PeopleTools 8.52, supports Oracle Active Data Guard. If the application domain is configured to support Oracle Active Data Guard, then some Application Engine (AE) and PSQUERY reports can be configured to run against an Oracle Active Data Guard standby database.
- Oracle Active Data Guard is discussed in this paper as being implemented at the DR site. If the DR site is located at a substantial distance away from the primary (on two separate continents with several thousand miles in between), then using Oracle Active Data Guard at such a DR site might not be a viable option given the high network latency. Implementing a local standby database for Oracle Active Data Guard and a remote standby is an ideal solution. This provides a local Oracle Active Data Guard standby, a local standby database for switchover for maintenance plus a remote standby database for site failover.
- PeopleSoft supports both Data Guard logical and physical standby databases. Oracle GoldenGate one-way replication is also supported. For the purposes of this discussion, we will focus on Data Guard physical standby database.
- **Database Snapshot Standby** allows for the physical standby to be opened read-write without compromising data loss service level agreements (SLAs). This allows the entire application technology stack to be started for testing and maintenance. Once the testing is completed, the database is returned to its role as a physical standby and all outstanding changes from the production database are applied.

Once the secondary site was established, the standby database was instantiated using the RMAN duplicate feature of the primar database, Data Guard was configured with redo shipping from the primary and redo apply at the standby. Data Guard Broker was used to configure and enable Data Guard.

4.3.2 Database Service Design

The database services used by the PeopleSoft application server are

DATABASE SERVICE NAME	DATABASE ROLE	PURPOSE
PAYROL_BATCH	PRIMARY	Production process scheduler
PAYROLE_ONLINE	PRIMARY	Production application server
PSQUERY	Oracle Active Data Guard Standby	Offload read-only queries

These services are configured in the database with Fast Application Notification (FAN). If one of the Oracle RAC database instances should become unavailable, the surviving RAC instance sends all previously connected clients an out-of-band FAN event causing the clients to drop their current TCP connections and reconnect via transparent application failover (TAF) to the services running on the surviving RAC instance, expediting the failover process. This same functionality exists for a failover to a physical standby database. The application domain servers do not need to be restarted after a failover.

In addition to these services being FAN enabled, they are also defined as role based services depending upon the database role (PRIMARY, PHYSICAL STANDBY, SNAPSHOT STANDBY) that determines if a service will start (or not start).

4.4 Delegation of Roles and Administration

This section describes the operating system user accounts, groups, and the administrative role at each level: database tier on Oracle Exadata Database Machine and the application and web servers on Oracle Exalogic. These OS accounts, groups, and roles are consistent at both the primary and DR sites.

4.4.1 Administrative Roles on Oracle Exadata Database Machine

On Oracle Exadata Database Machine, the Oracle Grid Infrastructure (Oracle Clusterware and Automatic Storage Manager – ASM) manages all cluster and storage services. Although not required, it is recommended that you install the Oracle Grid Infrastructure using a separate and dedicated OS user. Application databases should be installed into their own OS user account so that the grid infrastructure is managed separately from that of the PeopleSoft application database.

The following table illustrates how this was configured in this case study.

OS USER	OS GROUPS	ROLE
oracle	oinstall, dba	Clusterware and ASM administrator (Grid)
oracle_psft	oinstall, dba_psft, dba	PeopleSoft database administrator

Notice that the oinstall and dba OS groups are common between the "oracle" OS user account and that of "oracle_psft". This allows the PeopleSoft database being managed by the oracle_psft user to access the ASM services.

4.4.2 Administrative Roles on Oracle Exalogic Machine

On Oracle Exalogic, PeopleSoft PeopleTools and HRMS have been installed into the OS user account oracle_psft with group oinstall. This installation includes all of the infrastructure needed for PeopleSoft shown in the following table.

OS USER	OS GROUPS	ROLE
oracle_psft	oinstall, dba_psft	PeopleSoft Application and PIA administration

The users and groups were registered in NIS to facilitate the NFS v4 security model. NFS v4 is recommended for use with PeopleSoft shared homes for higher performance with PeopleSoft cache files when accessed across multiple PIA web servers.

What is not shared is the Micro Focus Server Express COBOL compiler and runtime environment. This must be installed on local storage on each node as root and must have its license manager configured and running on each node.

4.5 PeopleSoft Application and Web Tier Setup

In this case study, PeopleSoft was installed using a shared PS_HOME. The application servers and PIA web servers access the same software install locations. These servers also access the PeopleSoft report repository. For this case study, HRMS 9.1 FP3 was installed in the same location as PS_HOME. While this is a common installation, it is recommended that you install the PeopleSoft application (HRMS, FIN, Enterprise PORTAL) in a different location than PS_HOME in later releases of PeopleSoft. Thus, PS_APP_HOME would point to the application while PS_HOME points to the PeopleTools install. Both homes can be shared across all mid tier nodes as shared homes.

4.6 Sun ZFS Storage Appliance and Shared File System

All of the shared and local file systems are stored on the Sun ZFS Appliance 7320, which comes installed as part of the Oracle Exalogic machine. The file systems are exported from ZFS and are mounted with NFS v4 by all mid tier servers. Specific shared file systems are replicated to the

secondary DR site using ZFS replication. The following table shows each file system, mount points and state for each site.

PURPOSE	SHARE TYPE	EXPORTED AS (MOUNTED AS)	SITE STATE	COMMENTS
PeopleSoft Software Install File System – PS_HOME / PS_APP_HOME	Local Replicated	Exported as: /export/peoplesoft/peoplesoft_u01 Mounted as NFS v4: /u01/app/oracle	Site 1 Primary Site 2 Replica	NFS v4, Replicated to site 2, primary on site 2 post switchover/failover
PIA Report Repository	Local Replicated	Exported as: /export/peoplesoft/peoplesoft_repository_scan04 Mounted as NFS v4 /peoplesoft/report_repository	Site 1 Primary Site 2 Replica	NVS v4 mounted at site 1, ZFS replicated to site 2.
Server Specific Configuration File System – PS_CFG_HOME	Local Not replicated	Mounted as local storage /peoplesoft/local/ps_config	Site 1 Primary Site 2 Primary	The application Tuxedo and the PIA web server domains are placed here. Not replicated
Required COBOLB Server Express installation-COBDIR	Local Not Replicated	Mounted as local storage /peoplesoft/local/mf/SX51_WP4	Site 1 Primary Site 2 Primary	The COBOL compiler and run-time environment. Not replicated

4.7 Workload

Oracle Application Testing Suite was used to drive a PeopleSoft application workload during application tests. The workload consisted of 1,000 concurrent virtual users. The transaction spread across these 1,000 users consisted of:

- Employee pay slip review
- Employee benefits review
- Employee personal information update
- Employee benefits updates
- Administrator HR users performing updates
- Administrator HR review
- Manager HR review

During outage testing the behavior of this workload was monitored and any anomalies were noted.

5 PeopleSoft Outage Testing and Results

5.1 Failover Behavior

Table 4 summarizes the PeopleSoft behavior during Oracle RAC or Data Guard failover when client failover is configured. Except for a short pause as the failover occurs, the failure is transparent to the end user in most cases:

TABLE 4. PEOPLESOFT FAILOVER BEHAVIOR DURING ORACLE RAC OR DATA GUARD FAILOVER

PEOPLESOFT CLIENT OPERATIONS	BEHAVIOR
End user is inserting, updating, or deleting data and submits or saves the inserts/updates/deletes during or just after the database failure.	The data manipulation language (DML) will fail. Transactions will not get resubmitted. Oracle reconnects and reconstructs the database session on a surviving node and the end user must resubmit the transaction.
End user is paging through queried data (SELECTs) when the database failure occurs.	Oracle reconnects and reconstructs the database session on a surviving node, re-executes the query, repositions the SQL cursor, and returns the next set of rows.
End user is issuing a new query (SELECTs) or switching screens just after the database failure.	Oracle reconnects and reconstructs the database session on a surviving node.

Table 5 summarizes the failover behavior of the PeopleSoft Batch Process Scheduler, Application Engine (AE) jobs, Structured Query Report (SQR), PeopleSoft Query (PSQuery), XML Publisher (XMLP), and COBOL programs.

TABLE 5. PEOPLESOFT FAILOVER BEHAVIOR DURING CLIENT BATCH OPERATIONS

PEOPLESOFT CLIENT BATCH OPERATION	BEHAVIOR
Process Scheduler	Oracle reconnects and reconstructs the session on a surviving node. The process scheduler fails over with no administration intervention required.
Application Engine (AE) job submitted just *BEFORE* primary instance failure	Oracle reconnects and reconstructs the session on a surviving node but AE jobs may fail and show up in the PeopleSoft Process Monitor as "No Success". These jobs must be resubmitted. If the AE job has been implemented to be restartable, then the process scheduler will automatically restart the job.[3]

[3] If the AE job was not in an open transaction and the job was performing only SELECT statements, then it will fail over and complete successfully.

TABLE 5. PEOPLESOFT FAILOVER BEHAVIOR DURING CLIENT BATCH OPERATIONS

Application Engine (AE) submitted *during* or just *after* primary instance failure	Oracle reconnects and reconstructs the session on a surviving node, the AE job is then submitted on the surviving RAC node (or the new primary database failover) and completes successfully.
COBOL jobs just *before* primary instance failure	If the COBOL program runs pure queries (SELECT statements), then it will fail over to the surviving node and complete successfully. If the COBOL program executes INSERTs, UPDATEs, and DELETEs, it will *not* complete successfully on the surviving node. Manual intervention is required to restart the COBOL jobs.
Crystal and SQR reports	The behavior is the same as COBOL
PSQUERY, Tree Viewer, XMLP Viewer	These PeopleSoft components will fail over and complete successfully.

5.2 Unplanned PeopleSoft PIA Web Server Outages

The workload was ramped up before simulating the outage.

5.2.1 Instance Outage

The PeopleSoft PIA web server was stopped abruptly using the stopPIA.sh script or by killing the Oracle WebLogic Server instance wls process.

5.2.2 Node Outage

In the virtualized case, the vServer was stopped abruptly using the command "xm destroy" from the VM server (DOM0).

In the bare metal case, the server was stopped abruptly using a power reset from the iLOM service processor.

5.3 Unplanned PeopleSoft Application Server Outages

5.3.1 Instance Outage

The application server domains were stopped with the command:

```
psadmin -c shutdown!
```

Note the "!" which causes a forced shutdown.

5.3.2 Node Outage

In the virtualized case, the vServer was stopped abruptly using the command `xm destroy`. In the bare metal case, the server was stopped abruptly using a power reset from the iLOM service processor.

5.4 Unplanned Database Outages

5.4.1 Instance Outage

An Oracle RAC database instance was abruptly terminated with shutdown abort.

5.4.2 Node Outage

The server was stopped abruptly using a power reset from the iLOM service processor.

5.5 Unplanned Site Outage

5.5.1 Results

COMPONENT	BARE METAL/ VIRTUALIZED	UNPLANNED OUTAGE	DOWNTIME	OBSERVATIONS
PeopleSoft PIA Web Server	Bare Metal	Instance	N/A	256 (or 25%) OATS users encountered errors
	Bare Metal	Node	N/A	47 (or 4.7%) OATS users encountered errors
	Virtualized	Instance	N/A	252 (or 25%) OATS users encountered errors
	Virtualized	Node	N/A	17 (or 1.7%) OATS users encountered errors
PeopleSoft App Server	Bare Metal	Instance	1 seconds	No errors encountered by OATS users
	Bare Metal	Node	41 seconds	No errors encountered by OATS users
	Virtualized	Instance	1 seconds	No errors encountered by OATS users
	Virtualized	Node	6 seconds	No errors encountered by OATS users
Database Server	N/A	Instance	< 5 seconds	No errors encountered by OATS users
		Node	31 seconds	Depends on node eviction, the amount of outstanding redo to recover and when FAN events are sent Note that the cluster css miscount is set to 30 seconds on Exadata.
Site			35 seconds for database to become primary after the failover command was issued 55 seconds for	The Data Guard FAILOVER took 35 seconds to complete. The application startup took 55 seconds. To fully start and

			PeopleSoft application startup	accept logins.

Notes

- The PIA web server maintains session state for all active sessions. Failure of a given PIA web server will result in loss of session state and errors for those sessions. Therefore, downtime is not measured here as they will need to log back on and re-authenticate. When front-ended by the F5 load balancer, the number of users who encounter errors depends on the nature of the failure. Here are the following configuration scenarios:

  Scenario 1: Active user sessions logged directly on a web server that fails: All such users will fail and must re-login to a separate server or wait until the service is restored on the failed server.

  Scenario 2: Active user sessions logged on via F5 load balancer and a PIA web server instance fails, but the hardware remains up: In this case the number of failures depends on the implemented F5 health monitor, its timeout values and the number of active sessions at the time of the failure. In our case, we have implemented a simple HTTP "GET /" with a timeout value of 60 seconds. If this timeout is reached, the load balancer removes the node from the server pool and returns all outstanding HTTP requests with "reject" errors. In this scenario, users are blocked until the timeout timer expires. The longer the timeout, the higher the error count will be.

  Scenario 3: Users logged on via F5 load balancer and PIA web server node fails: In this scenario, the F5 load balancer internally pings the servers with ICMP pings to determine the health of the network TCP layer. This ping occurs at a fixed interval of 5 seconds. If the pings fail to return for up to 16 seconds, the node is removed from the server pool, and all outstanding HTTP requests are returned with "reject" errors. This case presents a smaller window of time for blocked sessions from piling up and consequently, fewer errors are seen as illustrated by the test results.

  Testing and care must be taken when implementing a health monitor to check the status of the web servers. Remember that this health monitor is designed for when the PIA web server experiences a failure or have higher response times such as when its primary application server domain fails, thus, response times can increase while the PIA reconnects to its backup application domain server.

  Setting the timeout value depends on the average length of time it takes to recover the web services, not including node or hardware failures, where ICMP pings would fail and force the server to be removed from the server pool.

  A PIA web instance failure will always result in all active (not idle) sessions encountering failures no matter what the health monitor timeout is. We want to treat the WebLogic instance failure like a hardware failure. It may not be possible in this case because there is no ICMP type functionality to validate that the WebLogic instance is down, not just stalled.

- The PeopleSoft Application Server failures result in some amount of downtime depending on the nature of the failure. An abrupt and forced shutdown of the application server (with `shutdown!`) brings down the Oracle Tuxedo application services along with all of the application threads that were running. In doing so, the network sockets used by the connected JOLT sessions are removed

causing the PIA web server to detect and react to the failure. Our results show a one second increase in response times at the PIA web server during the one second time period the PIA server took to reconnect to its backup application server.

- A failure of the node hosting one of the application servers (for bare metal) shows a much longer downtime – 41 seconds in our testing. In this case, there are no network events that the PIA web server can react to. The JOLT sessions will hang until the JOLT layer times out at which point the PIA re-establishes its JOLT connections to its backup application server. No online user encountered errors.
- A failure of the vServer (virtualized) node hosting an application server shows different downtime results. To cause a failure, the `xm destroy` Xen command was used. When this command was used to destroy a vServer running an active application domain server, the PIA JOLT sessions were able to re-establish connections to the backup application server with in six seconds. As part of the vServer tear down, the virtual network interfaces is de-allocated that causes a socket close network event. No online users experienced any errors.
- An Oracle RAC database instance failure via a `shutdown abort` shows that the application domain server was able to reconnect to the remaining Oracle RAC instance in less than 5 seconds. The database services that the application uses to connect are FAN enabled. The shutdown abort of the databases causes the instance to issue an INSTANCE DOWN event that the application server receives and then reconnects to the remaining Oracle RAC instance.
- A RAC node failure results in on average, a 31 second downtime in our testing. The Oracle RAC database resides on a quarter rackExadata Database Machine where Oracle Cluster Ready Services (CRS) is pre-configured. Best practices for CRS MISSCOUNT is 60 seconds, but for critical applications this can be decreased to no more than 30 seconds which is what it is set to for these tests. Once the cluster determines that a node cluster heart beat is no longer present after the 30 second period, the node is evicted from the cluster and one of the remaining RAC instances will recover the failed instance. The instance performing the recovery sends an INSTANCE DOWN event to all previously connected clients so that these clients can commence transparent application failover (TAF).

6 Best Practices

6.1 Exadata and Database Best Practices

- Review all of the relevant My Oracle Support notes. Specifically review My Oracle Support note 888828.1. This My Oracle Support note contains links to other My Oracle Support notes that should be reviewed.
- Configure Linux HugePages on each database server node. See My Oracle Support note 361468.1, "HugePages on Oracle Linux 64-bit".
- Set the database parameter USE_LARGE_PAGES='ONLY' to utilize HugePages.
- Run the latest version of Exachk on a regular basis and review its output report. See My Oracle Support note 1070954.1 latest version of Exachk. Correct any critical issues that are identified. Reviewed any items marked as "Failed" and if appropriate, correct those items. See Appendix A for items that can be ignored.
- If the application is critical and requires a faster Oracle RAC node failover, the CSS MISSCOUNT can be set from the default of 60 to 30 seconds. It is not recommended to set this parameter lower than 30 seconds. See Appendiix A for further details.
- Configure database role based services that are FAN enabled.
- Deploy PeopleSoft on an Oracle RAC database and use role base services for database connections.
- Enable Flashback on both primary and standby databases. This allows for quick re-instantiation of a failed primary database if the database itself was not lost. This feature also allows for error correction without having to perform a restore from backups.
- Preallocate your flashback logs to improve performance.
- Use Recovery Manager (RMAN) to backup the PeopleSoft database on a regular backup schedule. For example, perform full level 0 backups weekly and incremental level 1 backups daily.

6.2 Exalogic and PeopleSoft Application Best Practices

- Make sure all required RPMS needed to support 32 bit based applications are present. For our test, the COBOL license manager requires 32 bit libraries that required applying the July 2013 patch.
- Configure NIS to take advantage of NFS v4 higher performance benefits.
- Do not use NFS v4 mount options actime=0 or noac mount options as this will impact performance.
- Make sure all PeopleSoft application domains and PIA web server logs are written to the PS_CFG_HOME/ directory location and not PS_HOME or PS_APP_HOME which are shared homes.
- Use shared homes to reduce the time needed to deploy the secondary site and for performing out-of-place patching which also should reduce maintenance time.

- Ensure that the application server domains use FAN enabled role-based database services in their TNS connect strings.
- Configure the PIA web servers to connect to a primary and backup application domain server.
- Enable PIA_access.log with extended log format which can be configured using the Oracle WebLogic Server Administration console on the PIA servers.
- Take regular backups of PS_HOME, PS_APP_HOME, each local PS_CFG_HOME, and the report repository.

6.3 High Availability and Disaster Recovery Best Practices

- Deploy a second geographically separate site to serve as the disaster recovery site which can run the PeopleSoft workload.
- Always use multiple PeopleSoft servers for application, batch, and PIA web servers at both the primary and secondary DR site.
- Implement a physical standby database for failover in case of a site failure or disaster.
- Use Oracle Active Data Guard with PeopleSoft so that higher leverage of the standby database can be realized by offloading reporting and adhoc queries to the standby database. If the DR site is a substantial distance away from the primary (on two different continents thousands of miles apart) and network latency is high, implement a local standby and configure it for Oracle Active Data Guard along with the remote Data Guard standby.
- Use Data Guard Broker to simplify the Data Guard standby database configuration.
- Use ZFS replication for PeopleSoft PS_HOME, PS_APP_HOME and the report repository.
- Make sure the surviving nodes can handle the load that was on the failed node.
- Export the PS_HOME, PS_APP_HOME, and the report repository shared file systems at the DR site. It is not necessary to remount the file systems at the DR site when the DR site assumes the primary role. Their state will change from read-only to read-write.
- Set F5 load balancer health monitor timeouts appropriately according to your environment and testing.
- Conduct annual or semi-annual switchover testing between the primary and DR sites to validate processes and procedures.
- Leverage the DR site for testing using database snapshot standby and ZFS cloning of the PeopleSoft report repository.
- Always test patches and configuration changes in a test environment before promoting to production. This practice reduces risks by following proper change control and validation processes.

7 Conclusion

Deploying PeopleSoft on both Oracle Exadata Database Machine and Oracle Exalogic serves as a common standard platform where unified deployments and management can be leveraged. MAA technologies and best practices are integrated at all layers of the stack. The use of both Data Guard and ZFS replication reduces the time to deploy the secondary site for disaster recovery as well as for switchover, while simplifying the requirements for failover. PeopleSoft has adapted database technologies such as FAN, TAF, and SCAN support, and is capable of remaining up for rolling maintenance activity reducing downtime. For performance, Exadata smart logging, smart scans, and write back flash cache provide higher throughput for PeopleSoft batch processes, and online users enjoy a consistent reliable performance. PeopleSoft also supports the SDP protocol over InfiniBand that provides low latency and high bandwidth network for non-set oriented batch jobs.

The PeopleSoft applications deployed on Oracle Exadata Database Machine and Oracle Exalogic realize a high level of HA when MAA best practices are followed. Our testing shows that zero users are impacted for Oracle RAC database instance failures and application domain server failures. The time to failover and recover is several seconds for such failures and in many cases go unnoticed by end users.

With Oracle Exalogic and ZFS storage, physical deployment footprint can be reduced and at lower cost. ZFS replication provides the much needed safeguards for the middle tier where critical data feeds (inbound and outbound) can be replicated to a secondary site should a lost of the primary site occur.

8 References

- MAA on Oracle Technology Network (OTN)
  http://www.oracle.com/goto/maa
- Oracle Database High Availability Overview
  http://www.oracle.com/pls/topic/lookup?ctx=db112&id=HAOVW
- Oracle Real Application Clusters Administration and Deployment Guide
  http://www.oracle.com/pls/topic/lookup?ctx=db112&id=RACAD
- Oracle Data Guard Concepts and Administration
  http://www.oracle.com/pls/topic/lookup?ctx=db112&id=SBYDB
- Oracle Flashback Technology
  http://www.oracle.com/pls/topic/lookup?ctx=db112&id=CNCPT1439
- Oracle Automatic Storage Management Administrator's Guide
  http://www.oracle.com/pls/topic/lookup?ctx=db112&id=OSTMG
- Oracle Database Backup and Recovery User's Guide (Oracle Recovery Manager)
  http://www.oracle.com/pls/topic/lookup?ctx=db112&id=BRADV
- Oracle Secure Backup Administrator's Guide
  http://www.oracle.com/pls/topic/lookup?ctx=db112&id=OBADM
- My Oracle Support note 361468.1, "HugePages on Oracle Linux 64-bit"
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=361468.1
- My Oracle Support note 749851.1 "HugePages and Oracle Database 11g Automatic Memory Management (AMM) on Linux"
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=749851.1
- My Oracle Support note 1392497.1 "USE_LARGE_PAGES To Enable HugePages In 11.2"
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1392497.1
- My Oracle Support note 151972.1 "Dead Connection Detection (DCD) Explained"
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=151972.1
- BIG-IP Product Suite
  http://www.f5.com/products/big-ip/
- F5 DevCentral
  http://devcentral.f5.com/
- My Oracle Support note 1516025.1 "How To Configure NIS Master, Slave And Client Configuration In Exalogic Virtual Environment"
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1516025.1

- My Oracle Support note 1334857.1 "32-bit Client Install On Linux x86-64: PRVF-7532 Error Occurs For gcc-4.1.2 i386"
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1334857.1
- PeopleSoft Enterprise PeopleTools 8.52 Documentation Library
  http://docs.oracle.com/cd/E25741_01/psft/html/docset.html
- PeopleSoft: Implementing Oracle Active Data Guard
  http://docs.oracle.com/cd/E38689_01/pt853pbr0/eng/pt/tadm/task_ImplementingOracleActiveDataGuard-3b7d04.html
- Sun ZFS Storage 7000 System Administration Guide
  http://docs.oracle.com/cd/E26765_01/html/E26397/shares__projects__replication.html
- Oracle Exalogic Elastic Cloud Administrator's Guide: Appendix F Creating Server Templates from vServers
  http://docs.oracle.com/cd/E18476_01/doc.220/e25258/appendix.htm#sthref118
- My Oracle Support note 565535.1 "Flashback Database Best Practices & Performance"
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=565535.1
- My Oracle Support note 1274318.1 "Oracle Sun Database Machine Setup/Configuration Best Practices"
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1274318.1
- PeopleSoft MAA Best Practices
  http://www.oracle.com/technetwork/database/features/availability/maa-peoplesoft-bestpractices-134154.pdf
- PeopleSoft on Exadata
  http://www.oracle.com/technetwork/database/features/availability/maa-wp-peoplesoft-on-exadata-321604.pdf

9 Appendix A: Primary Site Setup

9.1 Primary Site Description

This section describes the steps taken in the project to set up the primary site to include Oracle Exadata Database Machine, database setup, Oracle Exalologic, Sun ZFS storage and the PeopleSoft application deployment. The primary site employed Oracle Exalogic bare metal (non-virtualized) compute nodes.

9.2 Database Server Setup on Oracle Exadata Database Machine

The standard Oracle Exadata Database Machine configuration was deployed on the primary site. As described earlier, the Oracle Exadata Database Machine at the primary site is an X3-2 quarter rack with high performance disks.

9.2.1 Grid Home and Database Home

The Grid Home was installed following the Exadata installation convention by Exadata OneCommand and is installed on all database nodes in the location:

/u01.app/11.2.0.3/grid

The Grid Home is owned by the 'oracle' user and is in the oinstall and dba groups.

The Oracle Database software for the PeopleSoft database is installed into its own ORACLE_HOME location, separate from where the OneCommand installed the DBM database. There are two ways to install the database software into a separate ORACLE_HOME:

- Download the Oracle Database 11*g* Release 2 (11.2.0.3) software and install it using Oracle Universal Installer. See My Oracle Support note 888828.1 for details.
- Clone the existing DBM ORACLE_HOME over to the new ORACLE_HOME location for PeopleSoft.

This project chose the second option of cloning from the DBM home as this would give us an 11.2.0.3 database software version with the current bundle patch BP15 as well.

The Oracle Database software home for the PeopleSoft database was installed in:

/u01/app/oracle_psft/product/11.2.0.3/dbhome_psft

It is owned by the oracle_psft user and in the oinstall and dba groups.

To clone from the DBM home, follow these steps:

1. As root on each database node, create the /u01/app/oracle_psft directory and make it owned by oracle_psft:oinstall:

```
mkdir –p /u01/app/oracle_psft

chown oracle_psft:oinstall /u01/app/oracle_psft
```

2. As the software owner (oracle_psft), make the appropriate directories:

```
mkdir –p /u01/app/oracle_psft/admin/PAYROLL_S1/adump

mkdir –p /u01/app/oracle_psft/cfgtoollogs

mkdir –p /u01/app/oracle_psft/diag

mkdir –p /u01/app/oracle_psft/product/11.2.0.3
```

3. As root on one of the compute nodes, zip up the dbmhome_1 ORACLE_HOME:

```
cd /u01/app/oracle/product/11.2.0.3

zip -r 11203_BP15_dbhome_1.zip dbhome_1
```

4. As the software owner (oracle_psft) on each database node, copy the 11203_BP15_dbhome_1.zip to the new ORACLE_HOME location, unzip it, then move the dbhome_1 to dbhome_psft:

```
cd /u01/app/oracle_psft/product/11.2.0.3

cp /u01/app/oracle/product/11.2.0.3/11203_BP15_dbhome_1.zip .

unzip –d 11203_BP15_dbhome_1.zip

mv dbhome_1 dbhome_psft
```

5. Create a small shell script to run the clone.pl procedure. The script should look something like the following example, but replace the host names and Oracle home path to match the environment it will run on.

```
echo "Clone started at `date`" | tee -a clone.log
perl /u01/app/oracle_psft/product/11.2.0.3/dbhome_psft/clone/bin/clone.pl
\ ORACLE_BASE=/u01/app/oracle_psft \
ORACLE_HOME=/u01/app/oracle_psft/product/11.2.0.3/dbhome_psft \
ORACLE_HOME_NAME=dbhome_psft1 '-O"CLUSTER_NODES={scam08db03,scam08db04}"'
\   '-O"LOCAL_NODE=scam08db03"' OSDBA_GROUP=”dba”
echo "Clone ended at `date`" | tee -a clone.log
```

   Place the above in to a script say clone.sh, add execute privileges and run it. Do this step on each database node.

6. As the software owner on each node, relink the database software. Make sure that ORACLE_HOME is defined. Make sure you include the ipc_rds option.

```
export ORACLE_HOME=/u01/app/oracle_psft/product/11.2.0.3/dbhome_psft

cd $ORACLE_HOME/rdbms/lib

make -f ins_rdbms.mk ioracle ipc_rds
```

7. As root on each database node, run the $ORACLE_HOME/root.sh script.
8. Set up the environment for each compute node.

```
set -o vi

alias scr='cd $HOME/framework/scripts'

export ORACLE_BASE=/u01/app/oracle_psft

#export ORACLE_HOME=/u01/app/oracle/product/11.2.0.2/dbhome_1

export ORACLE_HOME=/u01/app/oracle_psft/product/11.2.0.3/dbhome_psft

export ORACLE_DB=payroll_s1

export local_db=PAYROLL

case `hostname -s` in
        *03 ) ORACLE_SID=${local_db}1;;
        *04 ) ORACLE_SID=${local_db}2 ;;

esac

export ORACLE_SID

export PATH=$PATH:$ORACLE_HOME/bin

export LFRAME_PATH=/home/oracle_psft/psft_lframe

export PATH=$PATH:$LFRAME_PATH/bin:$LFRAME_PATH/gnuplot/bin:$HOME/bin

export PATH=$PATH:$COBDIR/bin:.
```

At this point, all nodes should be ready for use.

NOTE: Be sure that whether you createa new database or copy one over, as in the case for this project, that the database parameter AUDIT_FILE_DEST points to the correct location. In our case it points to:

/u01/app/oracle_psft/admin/PAYROLL_S1/adump

9.2.2 Linux HugePages Configuration

HugePages were configured on each database node. It is critically important that HugePages are configured when running PeopleSoft on Linux database platforms. Please see My Oracle Support note 361323.1 for further details on HugePages and how to calculate the proper value for your environment. For our X3-2 database nodes, the HugePages were set in the sysctl.conf:

```
vm.nr_hugepages=25213
```

9.2.3 ASM Disk Groups

The following table describes the ASM storage configuration:

ASM DISK GROUP NAME	REDUNDANCY	TOTAL SIZE (GB)
+DATA_SCAM08	NORMAL	15,593
+RECO_SCAM08	NORMAL	3,896
+DBFS_DG	NORMAL	894

9.2.4 Exadata Exachk

Exachk is an extremely valuable utility to run on Oracle Exadata Database Machine to identify potential issues at the OS, database, and storage layers. This utility should be run at regular intervals and after patching and software upgrades are performed. Exachk was run after the database was created to validate the configuration. However, some items that Exachk flags may conflict with PeopleSoft requirements and documentation. These items are:

PROCESSES may be flagged if not set to the Exadata recommended value of 1024. The recommendation is designed to mitigate risks of exhausting memory resources or having too high a run queue for each CPU which can adversely impact performance and the system overall.

However, for a heavly loaded PeopleSoft deployment with several application server domains, it may be necessary to increase the PROCESS parameter above 1024. For each Oracle RAC database instance, do not exceed the value of 2048 for PROCESSES on an each Oracle RAC instance running an X3-2 compute node.

Hidden parameters may be flagged, because in most cases these parameters are not required for proper operation and functioning of the database. However, some hidden parameters may be required to ensure performance or functional behavior expected by PeopleSoft. These parameters are:

```
_unnest_subquery=false

_ignore_desc_in_index=true

_gby_hash_aggregation_enabled=false
```

The above parameters are documented in the PeopleBooks PeopleTools documentation which can be found at: http://docs.oracle.com/cd/E25741_01/psft/html/docset.html.

In addition, the following parameters have been found to improve Flashback performance during normal operations if Flashback is enabled:

```
_db_flashback_log_min_size=8G

_db_flashback_log_min_total_space=120G
```

9.2.5 Database Initialization Parameters

The following is the full list of database initialization parameters for the primary database:

```
*._db_flashback_log_min_size=8589934592

*._db_flashback_log_min_total_space=128849018880

*._file_size_increase_increment=2143289344

*._gby_hash_aggregation_enabled=false

*._ignore_desc_in_index=true

*._unnest_subquery=false

*.archive_lag_target=0

*.audit_file_dest='/u01/app/oracle_psft/admin/PAYROLL/adump'

*.audit_trail='db'

*.cluster_database=TRUE

PSFT1.cluster_interconnects='192.168.44.227'

PSFT2.cluster_interconnects='192.168.44.228'

PAYROLL1.cluster_interconnects='192.168.218.130'

PAYROLL2.cluster_interconnects='192.168.218.131'

*.compatible='11.2.0.3.0'

*.control_files='+DATA_SCAM08/payroll/controlfile/current.308.811798725','+RE
CO_SCAM08/payroll_s1/controlfile/current.1969.821533763'

*.db_block_checking='MEDIUM'

*.db_block_checksum='FULL'

*.db_block_size=8192

*.db_create_file_dest='+DATA_SCAM08'

*.db_domain=''

*.db_lost_write_protect='TYPICAL'

*.db_name='PAYROLL'#Reset to original value by RMAN
```

```
*.db_recovery_file_dest='+RECO_SCAM08'

*.db_recovery_file_dest_size=107374182400000

*.db_unique_name='PAYROLL_S1'

*.dg_broker_config_file1='+DATA_SCAM08/PAYROLL/dr1payroll.dat'

*.dg_broker_config_file2='+RECO_SCAM08/PAYROLL/dr2payroll.dat'

*.dg_broker_start=TRUE

*.diagnostic_dest='/u01/app/oracle_psft'

*.dispatchers='(PROTOCOL=TCP) (SERVICE=PAYROLLXDB)'

*.fal_server='payroll_s2'

*.fast_start_mttr_target=300

PAYROLL1.instance_number=1

PAYROLL2.instance_number=2

*.listener_networks=''

*.log_archive_config='dg_config=(PAYROLL_S1,PAYROLL_S2)'

*.log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST'

*.log_archive_dest_2='service="payroll_s2"','LGWR ASYNC NOAFFIRM delay=0
optional compression=disable max_failure=0 max_connections=1 reopen=300
db_unique_name="PAYROLL_S2"
net_timeout=30','valid_for=(all_logfiles,primary_role)'

PAYROLL1.log_archive_dest_3='location="+RECO_SCAM08"','
valid_for=(STANDBY_LOGFILE,STANDBY_ROLE)'

PAYROLL2.log_archive_dest_3='location="+RECO_SCAM08"','
valid_for=(STANDBY_LOGFILE,STANDBY_ROLE)'

*.log_archive_dest_state_2='ENABLE'

PAYROLL1.log_archive_dest_state_3='ENABLE'

PAYROLL2.log_archive_dest_state_3='ENABLE'

PAYROLL1.log_archive_format='%t_%s_%r.dbf'
```

```
PAYROLL2.log_archive_format='%t_%s_%r.dbf'
*.log_archive_max_processes=4
*.log_archive_min_succeed_dest=1
PAYROLL1.log_archive_trace=0
PAYROLL2.log_archive_trace=0
*.log_file_name_convert=’+RECO_SCAM02’,’+REDO_SCAM08’
*.log_buffer=134217728
*.nls_length_semantics='CHAR'
*.open_cursors=1000
*.parallel_adaptive_multi_user=FALSE
*.parallel_threads_per_cpu=1
*.pga_aggregate_target=8G
*.processes=1024
*.remote_listener='scam08-scan3'
*.remote_login_passwordfile='exclusive'
*.sessions=2258
*.sga_target=12884901888
*.sql92_security=TRUE
*.standby_file_management='MANUAL'
PAYROLL1.thread=1
PAYROLL2.thread=2
PAYROLL1.undo_tablespace='UNDOTBS1'
PAYROLL2.undo_tablespace='UNDOTBS2'
*.use_large_pages='ONLY'
```

9.2.6 PeopleSoft Database Creation

The PeopleSoft HRMS payroll database was copied from the performance test lab. It contains the HRMS 9.1 application installed schemas and HR data for 500,000 employees. The database size is 750 GB. Once the database was copied over and configured for Oracle RAC, it was then registered with Oracle Cluster Ready Services as follows:

```
srvctl add database -d PAYROLL_S1 -o
/u01/app/oracle_psft/product/11.2.0.3/dbhome_psft -p
+DATA_SCAM08/PAYROL/spfilePAYROLL.ora -n PAYROLL -a "DATA_SCAM08,RECO_SCAM08"

srvctl add instance -d PAYROLL_S1 -i PAYROLL1 -n scam08db03

srvctl add instance -d PAYROLL_S1 -i PAYROLL2 -n scam08db04
```

Before proceding, make sure that the catbundle.sql script is executed to match the bundle patch by logging in to the database as SYS and running the catbundle.sql as follows:

```
cd $ORACLE_HOME

sqlplus / as sysdba

SQL> @?/rdbms/admin/catbundle.sql exa apply
```

9.2.7 Database Service Setup

The following commands set up the required FAN-enabled role-based database services to be used by the application server domains.

```
srvctl add service -d PAYROLL_S1 -s PAYROLL_ONLINE -r "PAYROLL1,PAYROLL2" -m
BASIC -e SELECT -q TRUE -l PRIMARY -z 180 -w 1

srvctl add service -d PAYROLL_S1 -s PAYROLL_BATCH -r "PAYROLL1,PAYROLL2" -m
BASIC -e SELECT -q TRUE -l PRIMARY -z 180 -w 1
```

If the primary database becomes the standby, PSQUERY is started for Oracle Active Data Guard query access.

```
srvctl add service -d PAYROLL_S1 -s PSQUERY -r "PAYROLL1,PAYROLL2" -m BASIC
-e SELECT -q TRUE -l PHYSICAL_STANDBY -z 180 -w 1
```

9.2.8 Cluster Ready Service Configuration

The cluster synchronization service (CSS) is the process and mechanism that determines how nodes within an Oracle RAC cluster are synchronized. If a node fails, CSS will evict the node and initiate cluster re-configuration. Its timeout timer set by MISSCOUNT is set to 60 seconds on Oracle Exadata Database Machine. This is a conservative value so that if a node experiences a brief stall for some reason, and if it can recover with in 60 seconds, it will not be evicted. However, for critical

applications that need a faster Oracle RAC node failover, 30 seconds can be used. It is NOT recommended that you set the CSS timeout timer to a value less than 30 seconds.

It is a business and administration judgement call to set the CSS timer value less than 60 seconds. In this case study, we set it to 30 seconds. On an Oracle RAC node failure, cluster reconfiguration will take place after 30 seconds and FAN-enabled clients will receive an "INSTANCE DOWN" FAN event to expodite reconnecting to the surviving Oracle RAC instances.

To change the CSS timeout timer, do the following steps on only one database (scam08db03) logged on as root. It is assumed that GRID_HOME is set to your Grid Infrastructure location, typically /u01/app/11.2.0.3/grid.

1. Get the current CSS timeout timer setting.

   ```
   $GRID_HOME/bin/crsctl get css miscount
   ```

2. Set the timeout timer to 30 seconds if it is not already in step 1 above.

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

   The CSS miscount can be set while the cluster is up.

3. Verify the new setting.

   ```
   $GRID_HOME/bin/crsctl get css miscount
   ```

9.3 Oracle Exalogic Bare Metal Server Setup for PeopleSoft Applications

The primary site used non-virtualized, bare metal Oracle Exalogic machine compute nodes to host the PeopleSoft application components. Two nodes were used for the application domain servers and two were used for the PeopleSoft Pure Internet Architecture (PIA) web servers.

The nodes are outlined in the table below:

NODE NAME	PURPOSE
scan03cn21	Application domain server 1
scan03cn22	Application domain server 2
scan03cn23	PIA web server 1
scan03cn24	PIA web server 2

9.3.1 Application Host Setup

Each of the above hosts were set up identically. The only difference was which components were started on which hosts according to the table above.

Environment configuration is identical for all four hosts:

```
export ORACLE_HOME=/u01/app/oracle/product/11.2.0.3/dbhome_client

export
TNS_ADMIN=/u01/app/oracle/product/11.2.0.3/dbhome_client/network/admin/payrol
l_s1

export local_db=PAYROLL_S1

export TUXDIR=/u01/app/oracle/product/Middleware/tuxedo10gR3 ;

export JAVA_HOME=/u01/app/oracle/product/jrockit-jdk1.6.0_22

export WLS_HOME=/u01/app/oracle/product/Middleware/wlserver_10.3

export PS_HOME=/u01/app/oracle/product/peoplesoft/PT852

export PS_APP_HOME=/u01/app/oracle/product/peoplesoft/PT852

export PS_CFG_HOME=/peoplesoft/local/ps_config

export COBDIR=/peoplesoft/local/mf/SX51_WP4

#Setup PeopleSoft configuration
. /u01/app/oracle/product/peoplesoft/PT852/psconfig.sh

export PS_JRE=${PS_HOME}/jre

export LD_LIBRARY_PATH=${TUXDIR}/lib:${COBDIR}/lib

export PATH=$PATH:$ORACLE_HOME/bin:${TUXDIR}/bin:$PS_HOME/bin

export LFRAME_PATH=/home/oracle/psft_lframe

export PATH=$PATH:$LFRAME_PATH/bin:$LFRAME_PATH/gnuplot/bin:$HOME/bin

export PATH=$PATH:$PS_HOME/jre/bin:$COBDIR/bin:$JAVA_HOME/bin:.

export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PS_HOME/bin:$ORACLE_HOME/lib:$ORACLE_HOME/j
dk/jre/lib/amd64/server
```

9.3.2 Sun ZFS Storage Appliance and Shared File System Creation for PeopleSoft Applicaiotns

All of the shared and local file systems are stored on the Sun ZFS 7320 Appliance which comes installed as part of the Oracle Exalogic machine. The file systems are exported from ZFS and are mounted with NFS v4 by all bare metal mid tier servers. Shared file systems were mounted as follows:

PURPOSE	MOUNTED ON	EXPORTED AS (MOUNTED AS)	SITE STATE	MOUNT OPTIONS
PeopleSoft shared home File System	All PeopleSpft app and Web Servers and	Exported as: /export/peoplesoft/peoplesoft_software_u01/app Mounted on all vServers as: /u01/app	Site 1 Primary Site 2 Read-Only	nfs4 rw,rsize=131072,wsize=131072,bg,hard,timeo=600
PeopleSoft Report Repository	All PeopleSpft App and Web Servers	Exported as: /export/psft_repository_scan04 Mounted on all vServers as: /peoplesoft/report_repository	Site 1 Primary Site 2 Read-Only	nfs4 rw,rsize=131072,wsize=131072,bg,hard,timeo=600

9.3.3 Create the ZFS Shares

To create a share on ZFS, make sure that all of the requirements for configuring NIS are met. See Appendix B for detailed steps for configuring NIS.

On the ZFS filer, we have created a project called "peoplesoft_software" and a share with in this project called "peoplesoft_software_u01" whose exported mount point is: /export/peoplesoft/peoplesoft_software_u01.

From the ZFS BUI, we made sure of the following:

1. On the **Projects** pane, select **peoplesoft_software**, and the share peoplesoft_software_u01 should be displayed.
2. Click **General**.
3. Make sure you enter the proper user and group numbers. From above, the oracle_psft UID is 1003 and the group oinstall UID is 1001.
4. Make sure that the privelidges are: owner rwx, group rwx and other: r-x.
5. Click **SHARES** towards the top of the UI, where it says: SHARES, PROJECTS, SCHEMA
6. Scroll down the list and find **peoplesoft_software / peoplesoft_software_u01**
7. Mouse over this row. The **Edit** (pinsel icon) and **Delete** (trashcan icon) will only display once you mouse over.
8. Click the **Edit** icon then click **Protocol** at the top.
9. Under **NFS**, make sure that the check box for **Inherit from project** is checked.
10. Make sure that **Share** mode is **Read/write.**
11. Uncheck the box **Disable setuid and setgid for file creation.**
12. Uncheck the box for **Prevent clients from mounting subdirectories.**

13. For **Anonymous user mapping**, enter `1003` which is the UID for oracle_psft.
14. Accept the defaults for **Character encoding** and **Security mode**.
15. For **NFS Exceptions**, if there are no exception listed, click the plus sign (+) to add an exception.
16. For the network **TYPE**, select **Network** from the list.
17. For **ENTITY**, our network as discussed earlier is 172.17.0.0/16.
18. For **ACCESS MODE**, select **Read/write** from the list.
19. For **CHARSET**, select **Default** from the list.
20. For **ROOT ACCESS**, check the check box.
21. Accept the defaults for **Share Level ACLs.**
22. Make sure all check boxes for **Inherit from project** are checked for all protocols (HTTP, SMB, FTP, SFTP, TFTP).
23. Click **Access** at the top.
24. Under **Root Directory Access**, the **User** should be oracle_psft and the **Group** should be oinstall.
25. For **Permissions**, **User** should be rwx, **Group**: rwx and **Other**: r-x.
26. Make sure the **Inherit from project** checkbox is checked, accept the defaults for the rest of this page, and click **APPLY** at the top to save changes.

We created another ZFS project called peoplesoft_repository with a share called peoplesoft_repository_scan04. All of the steps above were used to create this ZFS share.

To mount the shares, each of the four bare metal Oracle Exalogic compute nodes has the following entries in /etc/fstab:

```
192.168.219.207:/export/peoplesoft/peoplesoft_software_u01/app/oracle
/u01/app/oracle nfs4 rw,rsize=131072,wsize=131072,bg,hard,timeo=600

192.168.219.207:/export/peoplesoft/peoplesoft_repository_scan04
/peoplesoft/report_repository nfs4
rw,rsize=131072,wsize=131072,bg,hard,timeo=600
```

On each of the four bare metal Oracle Exalogic servers, the following directories were created:

```
mkdir –p /u01/app/oracle

mkdir –p /peoplesoft/report_repository
```

Then, the shares were mounted on all four servers:

```
mount /u01/app/oracle

mount /people/report_repository
```

9.4 PeopleSoft Application Software Installation

For this project, we cloned (copied) an existing PeopleSoft installation on a previous system over to the ZFS mounted share peoplesoft_software_u01 mounted by all bare metal Oracle Exalogic servers.

PeopleSoft on Oracle Exalogic can be deployed in one of two ways:

- PeopleSoft and all required software installed and run locally on each vServer.
- Using shared PS_HOME, ORACLE_HOME, and shared infrastructure that is required, but server specific configurations (PS_CFG_HOME) and MicroFocus Server Express COBOL are installed locally.

For minimizing maintenance downtime, the second option was implemented for this case study.

NOTE: This paper does not provide details for installing all of the necessary software components required for PeopleSoft. It also does not provide details on the PeopleTools Windows utilities such as Application Designer or Change Management.

The required software components for PeopleSoft HRMS in this project are:

- Oracle WebLogic Server 10.3
- Oracle JRockit 1.6.0.22
- Oracle Tuxedo 10.3
- Oracle Database 11*g* Release 2 (11.2.0.3) client software (64 bit)
- PeopleSoft PeopleTools 8.52.07
- PeopleSoft HRMS 9.1 FP3
- Micro Focus COBOL Server Express 5.1 WP4

Except for Micro Focus COBOL, all of the above software components were installed in the shared directory structure:

/u01/app/oracle/product/

Under the above directory:

11.2.0.3/dbhome_client (database client software)

jrockit-jdk1.6.0_22 (JRockit Java and JRE)

Middleware/coherence_3.6 (web server cluster)

Middleware/ tuxedo10gR3 (Oracle Tuxedo)

Middleware/utils

Middleware/ wlserver_10.3 (Oracle WebLogic Server)

peoplesoft/PT852

The Micro Focus COBOL should be installed on a local file system. It is often installed in the directory /usr/local/mf and owned by root. Later, you will see we chose to install it in /u02/app/local/mf/SX51_WP4. It must have the Micro Focus License Manager (mflman) configured and running regardless of where it is installed.

9.4.1 Install Micro Focus Server Exress COBOL

On each application server node that will run COBOL programs, install the Micro Focus Server Express 5.1 WP4 COBOL compiler and license manager facility following the instructions in the accompanying README. This must be installed as root. In our case study, it is installed in the /peoplesoft/local/mf/SX51_WP4 directory. It should also be configured so that the license manager is started after each node reboot. Ensure that the COBDIR environment variable is set to the install directory location.

9.4.2 Move PS_CFG_HOME to Local Directories

When configuring PeopleTools, the PS_CFG_HOME should be configured to point to a directory location local to the host running PeopleSoft application server or the PIA web server. Host and domain specific configuration files are stored in this location. Log files are also stored in this location. The local file system on each node should have the same directory structure layout.

On scan04, PS_CFG_HOME was set on all nodes to: /peoplesoft/local/ps_config.

1. Create the local directories on each server.

   As root:

   ```
   mkdir -p /peoplesoft/local/ps_config

   chown oracle_psft:oinstall /peoplesoft/local/ps_config
   ```

   Because we cloned the PeopleSoft software installation from an existing system where local homes were used, on that system PS_CFG_HOME pointed to

   /u01/app/oracle/product/peoplesoft/PT852/ps_config

   The above directory is on the ZFS share.

2. Copy it to the local directory on each compute node.

   ```
   cp –r  /u01/app/oracle/product/peoplesoft/PT852/ps_config/*
   /peoplesoft/local/ps_config/.
   ```

   These steps were repeated for each of the four bare metal servers.

   It is optional to remove the ps_config directory from the ZFS share.

9.4.3 PeopleSoft Application Server Domain Database Connection

To configure the application server domain, two items need to be configured.

- The tnsnames.ora file for connecting to the database

- The application server domain

The tnsnames.ora file needs to be configured. The following provides an example of a working tnsnames.ora usied in our project.

The environment variable TNS_ADMIN listed above for the primary nodes points to:

u01/app/oracle/product/11.2.0.3/dbhome_client/network/admin/payroll_s1

The secondary nodes at the DR site will point to:

u01/app/oracle/product/11.2.0.3/dbhome_client/network/admin/payroll_s2

Each has its own tnsnames.ora file specific to the database depending on the site which they will connect to. For the primary site, the tnsnames.ora file entries are:

```
#HR Online users
HR91FP3=
(DESCRIPTION_LIST =
  (FAILOVER=on)
  (DESCRIPTION =
    (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS=(PROTOCOL=tcp)(HOST=scam08-scan3.us.oracle.com)(PORT=1521)))
      (CONNECT_DATA=
         (SERVER=DEDICATED)
         (SERVICE_NAME=PAYROLL_ONLINE)))
     )

# Batch scheduler
HRBATCH=
(DESCRIPTION_LIST =
  (FAILOVER=on)
  (DESCRIPTION =
    (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS=(PROTOCOL=tcp)(HOST=scam08-scan3.us.oracle.com)(PORT=1521)))
      (CONNECT_DATA=
         (SERVER=DEDICATED)
         (SERVICE_NAME=PAYROLL_BATCH)))
)
```

Validate that your connect string alias works by using SQL*Plus to ensure proper connections:

```
sqlplus "/@HR91FP3 as sysdba"
```

PeopleTools treats the above two TNS connect strings as actual database names and refers to them as the DBName when configured in the PSADMIN utility for the application server and the process scheduler. For security purposes, when PeopleTools connects to the database and queries the

PS.PSDBOWNER table to validate that the database it is attempting to connect to is the correct database. The table has two columns: DBNAME and OWNER. The DBNAME must match the TNS connect string alias, and the OWNER is the owning schema of the application tables. The entries in our PSDBOWNER table are:

```
DBNAME   OWNERID
--------------   ---------------
HR91FP3  HR91FP3
HRBATCH  HR91FP3
```

Make sure that the TNS connect alias names are in the PSDBOWNER table in the DBNAME column as shown above.

To insert the rows into the PSDBOWNER table:

1. Log onto the database with SQL*Plus as SYS:

   ```
   sqlplus "/ as sysdba"
   ```

2. Insert the rows:

   ```
   INSERT INTO PS.DBOWNER VALUES ('HR91FP3',' HR91FP3');
   INSERT INTO PS.DBOWNER VALUES ('HRBATCH,' HR91FP3');
   COMMIT;
   ```

9.4.4 Configuring the Application Domain

On every node that runs the PeopleSoft application server, you must configure the Oracle Tuxedo domain server. In this project, we use the same domain configuration on all nodes. The domain name is HR91FP3. In the $PS_CFG_HOME/appserv/HR91FP3 directory you will find the psappsrv.cfg file. Make any configuration changes using the $PS_HOME/appserv/psadmin utility before re-deploying the domain HR91FP3 on each node. The steps to deploy the HR91FP3 domain are:

1. Run $PS_HOME/appsrv/psadmin utility.

   ```
   cd $PS_HOME/appsrv
   ./psadmin
   ```
   This starts the psadmin utility.

2. Select option 1: **Application Donain**.

3. Select option 1: **Administer a Domain**.

4. Select the domain name, in our case HR91FP3.

5. Select option 4: **Configure this domain**.

6. Enter Y to shut down the domain.

7. Review the configuration and ensure that all of your TNS connct string aliases are correct for DBName. In our case it is HR91FP3.

8. Select option 13: **Load domain as shown**.

The domain configuration is loaded and all of the required domain files for Oracle Tuxedo are regenerated. It should now be possible to start the application server domain from the PSADMIN utility.

9.4.5 Configuring the Process Scheduler

The PeopleSoft process scheduler (or batch process) must also be configured.

The steps for configuring the process scheduler are similar to thosee for the application server domain.

1. Run $PS_HOME/appsrv/psadmin utility.

   ```
   cd $PS_HOME/appsrv
   ./psadmin
   ```

   This starts the psadmin utility.

2. Select option 2: **Process Scheduler**.
3. Select option 1: **Administer a Domain.**
4. Select the domain name. In our case it is HR91FP3.
5. Select option 4: **Configure this domain.**
6. Enter Y to shut down the domain.
7. Review the configuration and ensure that all of your TNS connct string aliases are correct for DBName. In our case it is HRBATCH.
8. Select option 3: **Load domain as shown**.
9. On each node that runs the process scheduler, create a symbolic link so that its report output points to the shared report repository:

   ```
   cd $PS_CFG_HOME/appsrv/prcs/HR91FP3

   mv log_output log_output_backup

   ln –s /peoplesoft/report_repository/log_output log_output
   ```

   The domain configuration is loaded and all of the required domain files for Oracle Tuxedo are regenerated. It should now be possible to start the process scheduler from the PSADMIN utility.

9.4.6 PeopleSoft PIA Web Server Domain Configuration

The PIA web server requires more configuration attention. Because the scan03 and scan04 local file system directory structures are different, both host names and directory structure paths must be changed, as well as other changes in the following files. Note that the domain name for the PIA is peoplesoft.

1. Change the directory paths and host names in $PS_CFG_HOME/webserv/peoplesoft/bin:

   ```
   setEnv.sh

   startPIA.sh

   stopPIA.sh
   ```

2. In $PS_CFG_HOME/ webserv/peoplesoft/piaconfig/properties/logging.properties file, set

   ```
   java.util.logging.FileHandler.pattern =
   /peoplesoft/local/ps_config/webserv/peoplesoft/servers/PIA/logs/PIA_servle
   ts%u.log
   ```

3. Set the primary and backup application server domains and the report repository:

   ```
   $PS_CFG_HOME/ webserv/peoplesoft/applications/peoplesoft/PORTAL.war/WEB-
   INF/psftdocs/ps/ configuration.properties and set (per our example):
   psserver=scan04cn21:9000{scan04cn22:9000}
   ReportRepositoryPath= /peoplesoft/report_repository/log_output
   ```

4. Change or correct the host name and the correct port number in:

   ```
   $PS_CFG_HOME/peoplesoft/ps_config/webserv/peoplesoft/applications/peopleso
   ft/PORTAL.war/WEB-INF/weblogic.xml
   <cookie-name>scan03cn23-8080-PORTAL-PSJSESSIONID</cookie-name>
   ```

5. Change or correct the host name in $PS_CFG_HOME/webserv/peoplesoft/config/config.xml

   ```
   <server-private-key-alias>scan04cn23.us.oracle.com</server-private-key-
   alias>
   ```

6. In the same file, change or correct the listener port (not the one under ssl):

   ```
   <listen-port>8080</listen-port>
   ```

On all nodes ensure that the host names are correct. Note that the domain pertains only to the node it is running on.

9.4.7 Application and PIA Web Server Scripts

Simple startup and shutdown scripts were created to make it easy to start and stop each PeopleSoft component.

Application Domain Server:

startAPP.sh:

```
#!/bin/sh
export ORACLE_HOME=/u01/app/oracle/product/11.2.0.3/dbhome_client
export PS_HOME=/u01/app/oracle/product/peoplesoft/PT852
source /home/oracle_psft/payroll.env
export domain=HR91FP3
```

```
cd ${PS_HOME}/appserv
date
echo "------ Starting Apps Server for domain: $domain ----"
${PS_HOME}/appserv/psadmin -c boot -d $domain
```

stopAPP.sh:

```
#!/bin/sh
export ORACLE_HOME=/u01/app/oracle/product/11.2.0.3/dbhome_client
export PS_HOME=/u01/app/oracle/product/peoplesoft/PT852
source /home/oracle_psft/payroll.env
export domain=HR91FP3
cd ${PS_HOME}/appserv
date
echo "------ Starting Apps Server for domain: $domain ----"
#Note the shutdown! Is a forced shutdown.
${PS_HOME}/appserv/psadmin -c shutdown! -d $domain
```

startPS.sh:

```
#!/bin/sh
export ORACLE_HOME=/u01/app/oracle/product/11.2.0.3/dbhome_client
export PS_HOME=/u01/app/oracle/product/peoplesoft/PT852
export domain=HR91FP3
cd ${PS_HOME}/appserv
date
echo "------ Starting Process Scheduler for domain: $domain ----"
${PS_HOME}/appserv/psadmin -p start -d $domain
```

stopPS.sh:

```
#!/bin/sh
export ORACLE_HOME=/u01/app/oracle/product/11.2.0.3/dbhome_client
export PS_HOME=/u01/app/oracle/product/peoplesoft/PT852
export domain="HR91FP3"
cd ${PS_HOME}/appserv
echo "`date` Stopping Process Scheduler for domain: $domain"
${PS_HOME}/appserv/psadmin -p kill -d ${domain}
```

PIA Web Server

startWS.sh:

```
export ORACLE_HOME=/u02/app/oracle/product/11.2.0.3/dbhome_client
export PS_HOME=/u01/app/oracle/product/peoplesoft/PT852
source /home/oracle_psft/payroll.env
cd ${PS_CFG_HOME}/webserv/peoplesoft/bin
./startPIA.sh
```

stopWS.sh :

```
export ORACLE_HOME=/u02/app/oracle/product/11.2.0.3/dbhome_client
```

```
export PS_HOME=/u01/app/oracle/product/peoplesoft/PT852
source /home/oracle_psft/payroll.env
cd ${PS_CFG_HOME}/webserv/peoplesoft/bin
./startPIA.sh
```

9.5 F5 BIG-IP Load Balancer

In this case study, an F5 BIG-IP hardware load balancer was used to distribute the web traffic across both PIA web servers. A server pool called scan04_peoplesoft_web was created with two servers in the pool: scan04cn23 and scan04cn24. As explained previously, a health monitor was implemented to check the health of both web servers. The health monitor is a simple HTTP GET function as described here:

```
Name:  PeopleSoft_Web
Type: HTTP
Parent Monitor:  http
Interval:  10 seconds
Timeout: 60 seconds
Send String:  "GET / HTTP/1.0\r\n\r\n"
Receive String: "200 OK"
```

The above function monitors scan04cn23 and scan04cn24, which are in the F5 local traffic server pool. There are some considerations to keep in mind:

- Reducing the timeout time can result in false service down when the web server is stalled or has high response times due to other component failure and recovery.
- Setting the timeout too high will result in a higher number of user failures if the web server is down due to an abrupt shutdown or crash. In this case, existing user sessions will block and new sessions can be routed to the impacted web server only to eventually receive errors when the timeout period is reached.

One other important consideration is the capacity of each web server and application server to take on load due to a failure of either component. Ensure that testing and production analsyses are conducted to understand the required capacity requirements (CPU, memory) needed for each of the components to assume the additional load.

10 Appendix B: Secondary (DR) Site Setup

This section describes the secondary site setup for DR. This site employs Oracle Exalogic vServers (virtual) compared to the primary site which uses bare metal (physical) servers.

10.1 Database Server Setup on Oracle Exadata

An Oracle Exadata Database Machine quarter rack was provisioned at the disaster recovery site. The two database nodes were set up identical to that of the primary Oracle Exadata Database Machine to include HugePages, Grid Home, the database home for PeopleSoft, and database services. Exachk was also run to validate and correct any items not configured correctly with the exceptions noted in Appendix A.

For the database software setup, please refer to Appendex A-1.1: Grid Home and Database Home Setup. Follow the steps for cloning ORACLE HOMEs in that section.

10.1.1 ASM Disk Groups

The following table describes the ASM storage configuration at the disaster recovery site:

ASM DISK GROUP NAME	REDUNDANCY	TOTAL SIZE (GB)
+DATA_SCAM02	NORMAL	15,593
+RECO_SCAM02	NORMAL	3,896
+DBFS_DG	NORMAL	894

10.1.2 PeopleSoft Standby Database Instantiation

The primary database was copied from the Oracle Exadata Database Machine node scam08db03 over to the standby Oracle Exadata Database Machine node scam02db07 using the Recovery Manager (RMAN) duplicate feature. To use this process several prerequisite steps are required.

1. On the standby node scam02db07, make sure your environment set set up correctly including the environment variables ORACLE_HOME and ORACLE_SID.
2. Create a small init.ora file.

   On the standby Oracle Exadata Database Machine, scam02db07 in $ORACLE_HOME/dbs, create a small init.ora file. For this example name the file init_rman.ora and include the following parameters:

   ```
   db_name='PAYROLL'

   sga_size=4G

   Save the file and attempt to startup the instance with NOMOUNT:

   Sqlplus / as sysdba
   ```

```
SQL>  startup pfile=init_rman.ora nomount
```

Note that the ORACLE_SID is PAYROLL1.

3. Copy the orapwd<sid>.ora file on all primary nodes to all database nodes at the stanby if they are not already copied as part of the software cloning process.

   On the primary scam08db03:

   ```
   cd $ORACLE_HOME/dbs
   cp orapwdPAYROLL1.ora \
   oracle_psft@scam02db07:
   /u01/app/oracle_psft/product/11.2.0.3/dbhome_psft/dbs/.
   ```

4. On the standby (scam02db07) configure a static SID listener in the Grid listener.ora file

   ```
   SID_LIST_LISTENER=
     (SID_LIST=
      (SID_DESC=
      (SID_NAME=PAYROLL1)
        (GLOBAL_DBNAME=PAYROLL_S2)
        (ORACLE_HOME=/u01/app/oracle_psft/product/11.2.0.3/dbhome_psft)
      )
     )
   ```

   You must restart the listener:

   ```
   srvctl stop listener –n scam02db07
   srvctl stop listener –n scam02db07
   ```

5. On the primary (scam08db03) add a TNS connect string alias in the tnsnames.ora file.

   ```
   SCAM02DB07_PAYROLL =
     (DESCRIPTION =
     (SDU=32767)
     (LOAD_BALANCE = on)
     (FAILOVER=on)
       (ADDRESS = (PROTOCOL = TCP)(HOST = scam02db07)(PORT = 1521))
     (CONNECT_DATA =
         (SERVER = DEDICATED)
         (SERVICE_NAME = PAYROLL_S2)
       )
     )
   ```

   Specify the specific host name (scam02db07) and do not use the SCAN listener for the above connect string.

6. On the primary (scam08db03) check that you can successfully connect to the NOMOUNT instance running on scam02db07:

   ```
   sqlplus sys/<password>@SCAM02DB07_PAYROLL as sysdba
   ```

It should connect successfully. If not, on the standby (scam02db07) check to verify that the static SID is registered correctly with the listener. The lsnrctl status command should show something like:

```
Service "PAYROLL_S2" has 2 instance(s).
  Instance "PAYROLL2", status UNKNOWN, has 1 handler(s) for this
service...
```

The key is that PAYROLL_S2 has at least one instance hander PAYROLL1 whoes status is UNKNOWN.

7. On the primary (scam08db03) create an RMAN script that will duplicate the primary database to the standby database on scam02db07:

```
# Run RMAN to duplicate your database.
date
rm dup.log
time rman <<EOF! > dup.log
set echo on
connect target sys/welcome1; # Primary
connect auxiliary sys/welcome1@SCAM02DB07_PAYROLL; # destination
run {
# Create channels for parallism
     allocate channel prmy1 type disk;
     allocate channel prmy2 type disk;
     allocate channel prmy3 type disk;
     allocate channel prmy4 type disk;
     allocate channel prmy5 type disk;
     allocate auxiliary channel stby1 type disk;
duplicate target database for standby from active database
spfile
     parameter_value_convert
'+DATA_SCAM08','+DATA_SCAM02','+RECO_SCAM08','+RECO_SCAM02'
     set audit_file_dest='/u01/app/oracle_psft/admin/PAYROLL_S2/adump'
     set db_unique_name='PAYROLL_S2'
     set log_file_name_convert='+RECO_SCAD08','+RECO_SCAM02'
     set db_create_file_dest='+DATA_SCAM02'
     set db_recovery_file_dest='+RECO_SCAM02'
     set DB_RECOVERY_FILE_DEST_SIZE='10000G'
     set control_files='+DATA_SCAM02'
     set log_archive_dest_1='LOCATION=+RECO_SCAM02'
     set cluster_database='FALSE'
     set use_large_pages='TRUE'
     set standby_file_management='AUTO'
     set sga_target='4G'
     set listener_networks=''
nofilenamecheck;
}
EOF!
Date
```

Usage notes:

- It may be necessary to vary the number of channels allocated depending on the network link being used.
- If the CLUSTER_INTERCONNECTS parameter is set on the primary, this parameter must be cleared in the SPFILE at the primary BEFORE running the RMAN DUPLICATE command. Once the DUPLICATE completes, then the CLUSTER_INTERCONNECTS parameter can be set back to their original value. The DUPLICATE command will fail if this parameter is not reset on the primary. To reset the CLUSTER_INTERCONNECTS, execute the following commands on the primary database

  alter system reset cluster_interconnects scope=spfile sid='PAYROLL1';
  alter system reset cluster_interconnects scope=spfile sid='PAYROLL2';
- Ensure that in the RMAN script, the CLUSTER_DATABASE parameter is set to FALSE.
- Make sure that the directory that the audit_file_dest parameter points to exists.
- The time to complete the duplicate operation depends on the network bandwidth and current network usage.

8. Check for any errors in the log file dup.log created by the above script.
9. Do not open the standby database at this time. It should remain mounted.

10.1.3 PeopleSoft Standby Database Configuration

In this section, we place configuration files where they belong and add the database to Oracle Cluster Ready Serivices. The next section will discuss Data Guard configuration.

All of the following steps are completed on the standby (scam02db07):

1. Create a database parameter file from the spfile

   Log into SQL*Plus as SYS and issue:

   ```
   SQL> create init_standby.ora from SPFILE;
   ```

2. Review the init_standby.ora file and make any appropriate changes.

   ```
   LOG_ARCHIVE_DEST_1='LOCATION=USE_DB_RECOVERY_FILE_DEST'
   ```

   HugePages should already be configured.

   ```
   USE_LARGE_PAGES='ONLY'
   ```

   Change LOG_FILE_NAME_CONVERT

   ```
   LOG_FILE_NAME_CONVERT='+REDO_SCAM08','+REDO_SCAM02'
   ```

3. Create new SPFILE into ASM.

```
SQL> create SPFILE='+DATA_SCAM08/PAYROLL/spfilePAYROLL.ora' from
pfile='init_standby.ora';
```

4. Create initPAYROLL1.ora and initPAYROLL2.ora files.

   The content of these two files is:

   ```
   SPFILE='+DATA_SCAM08/PAYROLL/spfilePAYROLL.ora'
   ```

   Place initPAYROLL2.ora onto scam02db08: in $ORACLE_HOME/dbs directory.

5. Shut down the standby database, then do startup mount on both scam02db07 and scam02db08.

   ```
   SQL> shutdown immediate
   SQL> startup mount
   ```

6. Add the database to Oracle Cluster Ready Services (CRS):

   ```
   srvctl add database -d PAYROLL_S2 -o
   /u01/app/oracle_psft/product/11.2.0.3/dbhome_psft -p
   +DATA_SCAM08/PAYROL/spfilePAYROLL.ora -n PAYROLL -a
   "DATA_SCAM02,RECO_SCAM02"

   srvctl add instance -d PAYROLL_S1 -i PAYROLL1 -n scam02db07

   srvctl add instance -d PAYROLL_S1 -i PAYROLL2 -n scam02db08
   ```

   Make sure CRS can stop and restart the database on both nodes:

   ```
   srvctl stop database –d payroll_s2 –o immediate

   srvctl start database -d payroll_s2  -o mount
   ```

10.1.4 Standby Database Service Setup

```
srvctl add service -d PAYROLL_S2 -s PAYROLL_ONLINE -r "PAYROLL1,PAYROLL2" -m
BASIC -e SELECT -q TRUE -l PRIMARY -z 180 -w 1

srvctl add service -d PAYROLL_S2 -s PAYROLL_BATCH -r "PAYROLL1,PAYROLL2" -m
BASIC -e SELECT -q TRUE -l PRIMARY -z 180 -w 1
```

If the primary database becomes the standby, PSQUERY is started for Oracle Active Data Guard query access.

```
srvctl add service -d PAYROLL_S2 -s PSQUERY -r "PAYROLL1,PAYROLL2" -m BASIC
-e SELECT -q TRUE -l PHYSICAL_STANDBY -z 180 -w 1
```

10.2 Data Guard Broker Configuration and Setup

This section provides the steps used to enable Oracle Active Data Guard. The Data Guard Broker is used for much of the Data Guard configuration work. We do not describe the manual configuration of Data Guard.

10.2.1 Data Guard Broker Prerequisite Configuration

The following steps must be executed on both the primary (scam08db03) and standby (scam02db07) databases.

1. Create standby redo logs on both the primary and standby databases. If the primary database already has standby redo logs, the RMAN duplicate will have recreated them on the standby database. If so, skip this step.

   On the primary using SQL*Plus logged on as "/ as sysdba" determine the number of redo threads, online redo log:groups and size:

```
SQL> select group#,thread#,bytes
  2  from v$log;

    GROUP#    THREAD#      BYTES
---------- ---------- ----------
         1          1 8589934592
         2          1 8589934592
         3          1 8589934592
         4          1 8589934592
         5          2 8589934592
         6          2 8589934592
         7          2 8589934592
         8          2 8589934592

8 rows selected.
```

   The above results show that we have 2 threads with 4 groups each, 8 groups with one member each, and all logs are 8 GB in size. For Data Guard redo apply, we must create N standby logs where N is:

   N = (maximum # of logfile groups +1) * (maximum # of threads)

   In our case, we need 9 standby logs: (4 groups per thread) * (2 threads) + 1 = 9.

   Please refer to the *Oracle Database High Availability Best Practices* for more details.

   We now create the 9 standby redo logs on both the primary and standby using SQL*Plus:

```
alter database add standby logfile group 11 '+RECO_SCAM08' size 8G;

alter database add standby logfile group 12 '+RECO_SCAM08' size 8G;

alter database add standby logfile group 13 '+RECO_SCAM08' size 8G;

alter database add standby logfile group 14 '+RECO_SCAM08' size 8G;
```

```
alter database add standby logfile group 15 '+RECO_SCAM08' size 8G;

alter database add standby logfile group 16 '+RECO_SCAM08' size 8G;

alter database add standby logfile group 17 '+RECO_SCAM08' size 8G;

alter database add standby logfile group 18 '+RECO_SCAM08' size 8G;

alter database add standby logfile group 19 '+RECO_SCAM08' size 8G;
```

Repeat this process on the standby (scam02db07) and replace SCAM08 with SCAM02 in the above ALTER DATABASE statements.

2. Create a static SID entry for the Broker on each Oracle RAC node

   On each Oracle RAC instance for both the primary and standby add a static SID entry for the Broker in the Grid Infrastructure listener.ora making sure the SID_NAME is correct for each node, as follows:

```
SID_LIST_LISTENER=
  (SID_LIST=
   (SID_DESC=
   (SID_NAME=PAYROLL1)
     (GLOBAL_DBNAME=PAYROLL_S1_DGMGRL)
     (ORACLE_HOME=/u01/app/oracle_psft/product/11.2.0.3/dbhome_psft)

(ENVS="TNS_ADMIN=/u01/app/oracle_psft/product/11.2.0.3/dbhome_psft/network
/admin")
   ) )
```

   If the parameter DB_DOMAIN is set, then you must append the domain into the GLOBAL_NAME above. Once you have added the above to the listener.ora, restart the listener. You can issue an `lsnrctl` status to verify that the entry is listed. If DB_DOMEAIN is set, the domain will appear in the output list.

3. Add the TNS connect string alias to all of the tnanames.ora files on all of the nodes on the primary and standby. These entries are required in the next step.

```
PAYROLL_S1 =
  (DESCRIPTION =
  (SDU=32767)
  (LOAD_BALANCE = on)
  (FAILOVER=on)
    (ADDRESS = (PROTOCOL = TCP)(HOST = scam08-scan3)(PORT = 1521))
  (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = PAYROLL_S1)
    )
  )
```

```
PAYROLL_S2 =
  (DESCRIPTION =
  (SDU=32767)
  (LOAD_BALANCE = on)
  (FAILOVER=on)
    (ADDRESS = (PROTOCOL = TCP)(HOST = scam02-scan7)(PORT = 1521))
  (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = PAYROLL_S2)
    )
  )
```

Test the above connect strings (modify for your environment) to make sure they work correctly.

4. Configure the Broker database parameters on both primar and standby. The following example is SQL*Plus on the primary (scam08db03).

```
alter system set dg_broker_config_file1='
+DATA_SCAM08/payroll_s1/dr1payrol1.dat' sid='*' scope=both;

alter system set dg_broker_config_file2='
+RECO_SCAM08/payroll_s1/dr2payrol1.dat' sid='*' scope=both;

alter system set dg_broker_start=true sid='*' scope=both;
```

Repeat the above on the standbmy (scam02db07) and change payroll_s1 to payroll_s2.

5. Create and enable the Data Guard Broker configuration

Ensure that the database parameters for Data Guard parameters have been set properly as in the previous step. Before starting the Broker command line interface (CLI), it is recommended that you log on to each node at the primary and standby site as oracle_psft (for this case study) and use "tail –f" on the alert_<SID>.log file located in $ORACLE_HOME/diag/rdbms/payroll_s1/PAYROLL1/trace directory for the first Oracle RAC instance.

Run `dgmgrl` on the primary database (scam08db03) and add the primary and standby database using the TNS connect string aliases in the connect identifier, from the above in step 4:

```
% dgmgrl
DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production

Copyright (c) 2000, 2009, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/<password>
Connected.
DGMGRL> create configuration payroll_dgb
  as primary database is 'PAYROLL_S1'
```

```
  connect identifier is PAYROLL_S1;
```

Now add the standby:

```
DGMGRL> add database 'PAYROLL_S2'
 as connect identifier is PAYROLL_S2;
```

Note that the name in quotes is the db_unique_name of the two databases.

Enable the configuration. When you issue the following command, watch the tail –f sessions. You will see a lot of activity as the Broker configures each database.

```
DGMGRL> enable configuration;
Configuration enabled.
```

If the command was successful issue the following:

```
DGMGRL> show configuration
Configuration - payroll_dgb

  Protection Mode: MaxPerformance
  Databases:
    PAYROLL_S1 - Primary database
    PAYROLL_S2 - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS
```

If the result is not SUCCESS and you receive errors, refer to the Data Guard Broker documentation. Most likely, you have connect strings that cannot be resolved. Use the `show database` command within the Broker to provide more details.

Assuming that enabling the configuration was successful, the standby is brought into synch with the primary.

10.2.2 Oracle Active Data Guard Setup

PeopleSoft supports Oracle Active Data Guard. The following steps show how to set up a database service for offloading queries to the Oracle Active Data Guard database.

1. Disable managed recovery process (MRP) from the Broker:

```
DGMGRL> edit database 'PAYROLL_S2' set state='APPLY-OFF';
Succeeded.
```

2. Open the standby database on scam02db07 read-only.

```
SQL> alter database open read only;
Database altered.
```

3. Re-enable MRP from the Broker.

```
DGMGRL> edit database 'PAYROLL_S2' set state='APPLY-ON';
Succeeded.
DGMGRL> show database 'PAYROLL_S2';

Database - PAYROLL_S2

  Role:            PHYSICAL STANDBY
  Intended State:  APPLY-ON
  Transport Lag:   0 seconds
  Apply Lag:       0 seconds
  Real Time Query: ON
  Instance(s):
    PAYROLL1 (apply instance)
    PAYROLL2

Database Status:
SUCCESS
```

Oracle Active Data Guard is now enabled.

4. Configure the PSQUERY service.

PSQUERY is already added as a service to both the primary and standby CRS if you have followed the steps for the primary database setup and standby database setup above. It is necessary now to finish the configuration for PSQUERY. To complete the configuration, do the following steps:

5. On the primary database execute the following PL/SQL package:

```
EXECUTE DBMS_SERVICE.CREATE_SERVICE('PSQUERY', 'PSQUERY', NULL, NULL,TRUE,
'BASIC', 'SELECT', 180, 1, NULL);
```

DBMS_SERVICE.CREATE_SERVICE executed on the primary database replicates the service definition in the redo stream onto the standby. Note that the parameters passed to DBMS_SERVICE.CREATE_SERVICE must match those passed to the `srvctl` command that was performed earlier on the standby shown here:

```
srvctl add service -d PAYROLL_S2 -s PSQUERY -r "PAYROLL1,PAYROLL2" -m
BASIC  -e SELECT -q TRUE -l PHYSICAL_STANDBY -z 180 -w 1
```

6. On the primary, start and stop the PSQUERY service.

```
srvctl start service –d payroll_s1 –s PSQUERY

srvctl stop service –d payroll_s1 –s PSQUERY
```

7. Start the PSQUERY service on the standby (scam02db07).

```
srvctl start service –d payroll_s2 –s PSQUERY
```

10.3 Exalogic vServer Creation and Setup for PeopleSoft Applications

This section describes the vServer creation process for the PeopleSoft application on Oracle Exalogic. Each vServer is configured such that no one vServer resides on the same Oracle Exalogic physical compute node with distribution groups.

10.3.1 PeopleSoft vServer Creation

Using the Enterprise Manager Ops Center (EMOC) browser user interface (BUI), vServer types were created with the following vServer characteristics:

PURPOSE	VSERVER TYPE NAME	VCPU PER VSERVER	RAM PER VSERVER (GB)	STORAGE PER VSERVER (GB)
PeopleSoft App server	PSFT-App	8	8	24
PeopleSoft PIA Server	PSFT-Web	4	8	24

Distribution groups defined on Oracle Exalogic ensure that vServers are spread across the available Oracle Exalogic physical compute nodes. vServers in the same distribution group are not allowed to run on the same physical server. Using the browser-based user interface (BUI), the following distribution groups were created:

PURPOSE	DISTRIBUTION GROUP
PeopleSoft App	PSFT-App
PeopleSoft PIA	PSFT-Web

Using the above vServer types and vServer distribution groups, the following vServers for PeopleSoft were created.

VSERVER NAME	PURPOSE / HOSTING	VCPUS	RAM (GB)	LOCAL STORAGE (GB)
scan03vm0031-eobi1	Application Server	8	8	24
scan03vm0032-eobi1	Application Server	8	8	24
scan03vm0033-eobi1	PIA Web Server	4	8	24
scan03vm0049-eobi1	PIA Web Server	4	8	24

10.3.2 vServer Network Assignment and Configuration

When creating the vServers it was important to configure interfaces to allow access to the database servers, ZFS storage, clients, and administrators. During the vServer creation process, the network page prompts you to include each nectwork interface and the network (address space) that are required. For example, the Exadata database client side network is 10.133.44.0/22, the private network to the Oracle Exadata Database Machine compute nodes is 192.168.40.0/21 and the ZFS storage is on the network 172.17.0.0/16. Normally, these networks should have already been added to the Oracle

Exalogic virtual data center (vDC) so that you can add these interfaces during the vServer creation process. The BUI will present a list of available networks during this process.

When adding network interfaces, the BUI will allow for either automatically allocated IP addresses or you can manually enter them. In this project, we chose to allocate them automatically.

Once the vServers are created, the following table shows each vServer's characteristics:

SITE	PURPOSE	CLIENT, ADMIN, DATABASE NETWORK ADDRESS	CLIENT NETWORK IP ADDRESS TO ACCESS ZFS STORAGE
SCAN03	PeopleSoft App-Server Node 1	scan03vm0031-eoib1	172.17.0.29
	PeopleSoft App-Server Node 2	scan03vm0032-eoib1	172.17.0.18
	PIA Web Server Node 1	scan03vm0033-eoib1	172.17.0.16
	PIA Web Server Node 2	scan03vm0049-eoib1	172.17.0.27

10.3.3 Post vServer Creation Steps

After the vServers are created, each vServer must be configured using the following steps. Once you have the vServer IP address and its DNS host name (see step 1 below) all of the remaining steps must be performed as root on the vServer. The following steps serve as examples and you must modify them according to your environment.

1. Obtain the host name as it is registered in DNS.

   The host name allocated to each vServer will correspond with the vServer name in the BUI. In our case we wanted the host name to correspond with the client network address as configured in DNS, so we changed the host name on each vServer using the following procedure:

   a. Get the IP address from the **Network** tab of the vServer.

   b. Get the hostname from DNS.

```
[oracle_psft@scan03vm0031-eoib1 FSFO]$ nslookup 10.133.227.4
Server:         130.35.249.52
Address:        130.35.249.52#53
4.227.133.10.in-addr.arpa       name = scan03vm0031-
eoib1.us.oracle.com.
```

2. Change the vServer host name logged on as root.

   a. Edit the /etc/sysconfig/network file.

```
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=scan03vm0031-eoib1
```

   b. Execute as root: hostname xxx.

```
hostname scan03vm0031-eoib1
```

3. Create the /etc/resolv.conf file.

```
cat >/etc/resolv.conf <<EOF!
# Search domain and name server
search us.oracle.com
# Commented it out, because OUI complains about it
#domain us.oracle.com
nameserver 130.35.249.52
nameserver 130.35.249.41
EOF!
ping scam02db07
```

The `ping` command tests that the database node scam02db07 can be reached to validate the configuration.

4. Create the /u01 file system.

   a. View current partitions.

```
[root@scan03vm0031-eoib1 ~]# cat /proc/partitions
major minor  #blocks  name
 202        0   25165824 xvda
 202        1     104391 xvda1
 202        2    4835565 xvda2
 202        3     514080 xvda3
```

   b. Use `fdisk` to view free cylinders.

   The number of cylinders that are free and available depends on the vServer Type.

```
[root@scan03vm0031-eoib1 ~]# fdisk /dev/xvda
The number of cylinders for this disk is set to 3133.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
Command (m for help): p
Disk /dev/xvda: 25.7 GB, 25769803776 bytes
255 heads, 63 sectors/track, 3133 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

    Device Boot      Start         End      Blocks   Id  System
/dev/xvda1   *           1          13      104391   83  Linux
/dev/xvda2              14         615     4835565   83  Linux
/dev/xvda3             616         679      514080   82  Linux swap /
Solaris

Command (m for help):q
```

NOTE: The entire disk has 3133 cylinders. The last cylinder used is 679, so our free cylindars are from 680 to 3134 (not 3133).

c. Create the new partition.

```
[root@scan03vm0031-eoib1 ~]# parted -s /dev/xvda mkpart primary 680cyl
3134cyl
```

d. View the partitions again.

```
[root@scan03vm0031-eoib1 ~]# parted -s /dev/xvda unit cyl print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvda: 3133cyl
Sector size (logical/physical): 512B/512B
BIOS cylinder,head,sector geometry: 3133,255,63.  Each cylinder is
8225kB.
Partition Table: msdos

Number  Start   End      Size     Type     File system  Flags
 1      0cyl    12cyl    12cyl    primary  ext3         boot
 2      13cyl   614cyl   602cyl   primary  ext3
 3      615cyl  678cyl   64cyl    primary  linux-swap
 4      680cyl  3133cyl  2453cyl  primary  partd –s /dev/xvda unit cyl
print
```

e. A new device /dev/xvda4 should now exist.

```
[root@scan03vm0031-eoib1 ~]# cat /proc/partitions
major minor  #blocks  name
 202        0   25165824 xvda
 202        1     104391 xvda1
 202        2    4835565 xvda2
 202        3     514080 xvda3
 202        4   19703724 xvda4
```

f. Create the file system on /dev/xvda4.

```
[root@scan03vm0054-eoib1 ~]# mkfs.ext3 /dev/xvda4
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
4300576 inodes, 8595947 blocks
429797 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
263 block groups
32768 blocks per group, 32768 fragments per group
16352 inodes per group
Superblock backups stored on blocks:
```

```
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632,
2654208,
        4096000, 7962624

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 23 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to
override.mkfs.ext3 /dev/xvda4
```

g. Add it to /etc/fstab and mount it.

```
[root@scan03vm0054-eoib1 ~]# cat >> /etc/fstab <<EOF!

/dev/xvda4              /u01                    ext3    defaults        1 2

EOF!
[root@scan03vm0054-eoib1 ~]# mkdir /u01
[root@scan03vm0054-eoib1 ~]# mount /u01

[root@scan03vm0031-eoib1 ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/xvda2            4.5G  3.6G  764M  83% /
/dev/xvda1             99M   22M   73M  23% /boot
tmpfs                 4.0G     0  4.0G   0% /dev/shm
/dev/xvda4             19G  177M   17G  1% /u02
```

5. Configure NIS on the vServers.

   To support NFS v4 the NIS configuration must be completed first. See My Orcle Support note 1516025.1 for details about configuring the NIS master and slave vServers. If NIS is already configured, you must ensure that the Sun ZFS Storage Appliance is configured and has access to the NIS server. In addition, the OS users and groups that will be used on the vServers must also be added to the NIS server. If these requirements are met, then configure the PeopleSoft vServers as NIS clients. The following steps below must be performed as root on each of the vServers.

   a. Configure the NIS domain name.

      Edit the /etc/sysconfig/network file and add the NISDOMAIN entry:

      ```
      NETWORKING=yes
      NETWORKING_IPV6=no
      HOSTNAME=scan03vm0031-eoib1
      NISDOMAIN=us.oracle.com
      ```

   b. Configure the Name Service to use NIS.

Edit /etc/nsswitch.conf to use NIS after local files for passwd, shadow, and group name resolution, for example:

```
passwd:     files nis
shadow:     files nis
group:      files nis
```

c. Register the NIS Name Servers into yp.conf.

Edit /etc/yp.conf as shown in following example using the two servers scan03vm0051-eoib1.us.oracle.com and scan03vm0029-eoib1 as the master and slave NIS Name servers:

```
domain us.oracle.com server scan03vm0051-eoib1.us.oracle.com
domain us.oracle.com server scan03vm0029-eoib1.us.oracle.com
```

d. Configure the domain name in the ID Mapping Daemon.

Edit the /etc/idmapd.conf file as follows:

```
Domain = us.oracle.com
```

e. Restart the ID Mapping Daemon.

```
service rpcidmapd restart
```

f. Start ypbind.

```
service ypbind start
```

g. Ensure that ypbind restarts on vServer reboot .

```
chkconfig ypbind on
```

h. Test that NIS is running.

```
[root@scan03vm0031-eoib1 ~]# ypcat passwd
oracle:$1$TkfGSH8K$RLWlZLCB9NDS6O1aai1mB1:1000:1001::/home/oracle:/bin/bash
grid:$1$TwLWI4I/$rgxIHaQ7TwE4hVjknUKCE1:1001:1001::/home/grid:/bin/bash
oracle_psft:$1$aLTS7NYU$I4jzJjH7O.UFX5GMnStx2.:1003:1001::/home/oracle_psft:/bi
n/bash
[root@scan03vm0031-eoib1 ~]# ypcat group
oracle:!:1000:
oinstall:!:1001:
```

6. Add the PeopleSoft OS user and groups, and configure the group membership on each vServer.

```
groupadd -g 1001 oinstall
groupadd -g 1011 dba_psft
useradd --uid 1003 -g oinstall oracle_psft
usermod -d /home/oracle_psft -G dba_psft -g oinstall -s /bin/bash
oracle_psft
```

Again, ensure that the OS user and groups are registered in NIS.

7. Set up the environment logged on as oracle_psft.

   Create a shell file called payroll.env with the following:

```
set -o vi
export ORACLE_HOME=/u01/app/oracle/product/11.2.0.3/dbhome_client
export
TNS_ADMIN=/u01/app/oracle/product/11.2.0.3/dbhome_client/network/admin/pay
roll_s2
export local_db=payroll
export TUXDIR=/u01/app/oracle/product/Middleware/tuxedo10gR3
export JAVA_HOME=/u01/app/oracle/product/jrockit-jdk1.6.0_22
export WLS_HOME=/u01/app/oracle/product/Middleware/wlserver_10.3
export PS_HOME=/u01/app/oracle/product/peoplesoft/PT852
export PS_APP_HOME=/u01/app/oracle/product/peoplesoft/PT852
#export PS_CFG_HOME=/u01/app/oracle/product/peoplesoft/PT852/ps_config
export PS_CFG_HOME=/u02/app/local/oracle_psft/peoplesoft/ps_config
export COBDIR=/u02/app/local/mf/SX51_WP4
export PS_JRE=${PS_HOME}/jre ;
export ORACLE_SID
export LD_LIBRARY_PATH=${TUXDIR}/lib:${COBDIR}/lib
export PATH=$PATH:$ORACLE_HOME/bin:${TUXDIR}/bin:$PS_HOME/bin
export LFRAME_PATH=/home/oracle/psft_lframe
export PATH=$PATH:$LFRAME_PATH/bin:$LFRAME_PATH/gnuplot/bin:$HOME/bin
export PATH=$PATH:$PS_HOME/jre/bin:$COBDIR/bin:$JAVA_HOME/bin:.
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PS_HOME/bin:$ORACLE_HOME/lib:$ORACLE_HOM
E/jdk/jre/lib/amd64/server
. /u01/app/oracle/product/peoplesoft/PT852/psconfig.sh
```

   This shell script should be executed each time you log onto oracle_psft.

   At this point, the vServer should be ready to install PeopleSoft.

10.4 Configure and Enable ZFS Replication from the Primary Site

The ZFS replication facility can be used to replicate ZFS projects and their shares to a remote ZFS appliance. Once the target ZFS appliance is configured into both primary and secondary sites, the project can be replicated to the new site.

Replicaton of two projects were established between scan04sn at the primary site and scan03sn at the secondary site. These projects are:

ZFS PROJECT NAME	SHARE NAME
peoplesoft_software	peoplesoft_software_u01
peoplesoft_repository	peoplesoft_repository_scan04

These projects on scan04sn ZFS were configured to replicate to scan03sn as its target. The peoplesoft_software project is configured to replicate manually – on demand, and the peoplesoft_repository project is configured to replicate continuously.

As described earlier, the peoplesoft_software project with its peoplesoft_software_u01 share contains the common infrastructure software, ORACLE_HOME for the database client software and the PeopleSoft PeopleTools install (PS_HOME). These are deployed as shared homes at the primary site.

The peoplesoft_repository project with its peoplesoft_repository_scan04 is where the process scheduler runs jobs that write to the report repository, and where the PIA web servers access these reports.

Once replication has started, the "replicas" at the standby site can be exported as read-only file systems or create a "clone" of the most recent replicated snapshot.

For further details on ZFS replication, please see:
http://docs.oracle.com/cd/E26765_01/html/E26397/shares__projects__replication.html

10.4.1 Start ZFS Replication

To configure and enable ZFS replication, these steps were followed:

1. On the ZFS appliance at the primary site, scan04sn, using the BUI, select the first project, **peoplesoft_software**.
2. Click **Replication** .
3. Click the plus sign (+) to add a replication action.
4. Select the target (scan03sn01) from the pull-down list.
5. It is optional to select **SSL** (not used in this project.)
6. Select**Scheduled**.Click **Apply** without adding a schedule.
7. Click the update icon next to **manual**. this should start the first replication snapshot. Depending on the content size of the source share and the bandwidth and latency of the network this can take some time to complete.

For the peoplesoft_repository project repeat steps 1 through 6. In step 7 choose **Continuous** then click **Apply**. The project should show a continuous replication status.

10.4.2 Export ZFS Replicas at the Standby Site

1. On the ZFS appliance at the standby (scan03sn01), click **Projects** on the left side, then click **REPLICA**.
2. Select the replica **peoplesoft_software.**
3. To the right on the line with the share **peoplesoft_software_u01**, click the **Edit** icon (you must mouse over it to see it.)
4. Make sure the check box for **Inherit from project**, is checked.

5. Check the **Export** check box.
6. Note the export path (it will be the same as that on the primary.)
7. Click **Apply**.

Follow the same procedures for the peoplesoft_repository project.

10.4.3 Mount the ZFS Shares

The peoplesoft_software_u01 ZFS share contains the shared home deployment of PeopleSoft. All servers will have their PS_HOME access this share rather than the local file system. During the vServer post creation process, a local partition was created on /dev/xvda4 and mounted as /u01. This poses a small problem. We want to have the share to be mounted as /u01. To address this problem the following steps were performed:

1. Remount the local file system under /u02 on all vServers.

   As root:

   ```
   [root@scan03vm0031-eoib1 ~]$ mkdir /u02

   [root@scan03vm0031-eoib1 ~]$ umount /u01
   ```

   Edit /etc/fstab and change:

   ```
   /dev/xvda4              /u01               ext3    defaults        1 2
   ```

   To:

   ```
   /dev/xvda4              /u02               ext3    defaults        1 2

   [root@scan03vm0031-eoib1 ~]$ mount /u02
   ```

2. Mount the two ZFS shares.

   As root:

   ```
   [root@scan03vm0031-eoib1 ~]$ mkdir –p /u01/app

   [root@scan03vm0031-eoib1 ~]$ mkdir –p /peoplesoft/report_repository
   ```

   Add the following file system entries to /etc/fstab:

   ```
   172.17.0.9:/export/peoplesoft/peoplesoft_software_u01/app /u01/app nfs4
   rw,rsize=131072,wsize=131072,bg,hard,timeo=600

   172.17.0.9:export/peoplesoft/peoplesoft_repository_scan04
   /peoplesoft/report_repository nfs4 rw, rsize=131072, wsize=131072,
   bg,hard,timeo=600

   [root@scan03vm0031-eoib1 ~]$ mount /u01/app
   ```

```
[root@scan03vm0031-eoib1 ~]$ mount /peoplesoft/report_repository
```

10.5 PeopleSoft Application and PIA Web Server Installation on vServers

To install PeopleSoft at the secondary DR site, we use ZFS replication to replicate the PeopleSoft shared homes. There is no software installation required and thus, reduces the time to set up the application at the secondary site.

10.5.1 Move PS_CFG_HOME to the Local File System

On each vServer, create the directories and copy PS_CFG_HOME. The following example shows the procedure on scan03vm0031-eoib1.

As root:

```
[root@scan03vm0031-eoib1 ~]$ mkdir –p /u02/app/local/oracle_psft

[root@scan03vm0031-eoib1 ~]$ chown oracle_psft:oinstall
/u02/app/local/oracle_psft
```

As oracle_psft:

```
[oracle_psft@scan03vm0031-eoib1 ~]$ cd /u02/app/local/oracle_psft

[oracle_psft@scan03vm0031-eoib1 ~]$ mkdir –p peoplesoft/ps_config

[oracle_psft@scan03vm0031-eoib1 ~]$ cp –r
oracle_psft@scan04cn21:/peoplesoft/local/ps_config/*
/u02/app/local/oracle_psft/peoplesoft/ps_config/.
```

In payroll.env, change PS_CFG_HOME:

```
export PS_CFG_HOME=/u02/app/local/oracle_psft/peoplesoft/ps_config
```

To complete the installation and configuration, we followed the steps in Appendix A-9.4.4 through A-9.4.7 for configureing the application domain server and the PIA web server.

If you want to create a VM template, proceed to section B-4.1.

10.5.2 Create an Oracle Exalogic VM Template (Optional)

PeopleSoft is delivered in a couple of different forms. The first form is the traditional multiple CD downloads in addition to requiring the separate infrastructure download of Oracle WebLogic Server, Oracle Tuxedo, Oracle JRockit, Coherence and the Oracle Database client software. Each software component must be installed separately prior to installing PeopleSoft PeopleTools and the PeopleSoft application. A second way PeopleSoft is delivered is in the form of a VM (VirtualBox Appliance) image (a VM template) designed to be imported into a generic Oracle Linux VM, or in the case of Oracle Exalogic, used to create one or more vServers. The PeopleSoft VM template contains all of the required infrastructure software and PeopleSoft PeopleTools -- all pre-installed. This substantially

reduces the time required to deploy PeoleSoft. Once a vServer has been created based on the PeopleSoft VirtualBox image, all that is needed is to install Micro Focus Server Express COBOL and install the PeopleSoft application. There may be other 3rd party components that your environment might require such as enVision and Crystal Reports that must be installed and configured separately.

It is also possible to create a VM image (or Oracle Exalogic vServer template) from your own PeopleSoft install on Oracle Exalogic. This template can then be used to provision additional vServers as needed.

There are several reasons for creating a VM template of the PeopleSoft installation. They include:

- Provosioning of new production vServers
- Reduced time to deploy in a standard deployment paradigm
- Create new vServers for DR, test and development projects on a different Oracle Exalogic machine

The VM template image when constructed contains a full bootable OS, all mounted local disks and their contents, and all OS user and groups. The procedures for creating and importing VM templates used in this project can be found at:
http://docs.oracle.com/cd/E18476_01/doc.220/e25258/appendix.htm#sthref118.

After following these steps, you should have an Oracle VM template ready to be imported into a new vServer, configured and deployed.

NOTE: If you created a VM template with Micro Focus COLBOL installed, the COBOL run-time license manager will not function when the template is imported into a new vServer. You must either recreate the mflam license database or re-install Micro Focus COBOL. The license key is generated and keyed for the specific server it was installed on.

10.5.3 Install Micro Focus COBOL

On the vServers, we chose to install the Micro Focus COBOL compliler and license manager in /u02/app/local/mf/SX51_WP4. Each vServer has their own copy of Micro Focus COBOL installed locally using the same directory structure. Follow the accompanying instructions in the README to complete the installation. Make sure that the license manager is restarted after system reboot.

10.6 Configure PeopleSoft for Oracle Active Data Guard

As discussed in the main section of the paper, PeopleSoft PeopleTools version 8.52 and higher supports Oracle Active Data Guard. To enable PeopleTools to support Oracle Active Data Guard, the following is required:

- A physical standby database that has Oracle Active Data Guard enabled – described earlier in Appendix B
- A database service that can be started on the Oracle Active Data Guard database instance – described earlier in Appendix B
- A second database schema – ours is called PSFTADG2

- A secondary Access ID created in PeopleSoft
- A database link that ALWAYS points to the primary database service

The procedures for enabling PeopleTools support for Oracle Active Data Guard are documented at:

http://docs.oracle.com/cd/E38689_01/pt853pbr0/eng/pt/tadm/task_ImplementingOracleActiveDataGuard-3b7d04.html

These procedures should be carried out at both the primary and secondary sites. Note that much of the configuration is performed at the primary site but the application server at the secondary site must still be configured, specifically the application and batch server configuration files.

A few important key items to note:

- The PeopleSoft application server domains at both the primary and secondary sites must be configured to support Oracle Active Data Guard for switchover operations to work properly.
- If the database parameter GLOBAL_NAMES is set to TRUE, then the database link name must match the name of its target database including the DB_DOMAIN if set. The supplied PeopleTools script that creates the database link will fail if GLOBAL_NAMES is set to TRUE.
- Once the PeopleTools application domain server is configured to support Oracle Active Data Guard, both the primary and standby databases and all database services must be up before attempting to start the application server, otherwise, the application server startup will fail.
- Make sure that the database link you create as part of the scripts supplied by PeopleSoft ALWAYS points to the primary database using a service that only runs on the primary. In our case the connect string alias we used is HR91FP3_PRIMARY.
- If the standby database is down for maintenance the PSQUERY service will be down. Start the PSQUERY service on the primary until the standby database is brought back up, at which point you can migrate the PSQUERY service back to the standby.

The TNS connect string alias HR91FP3_PRIMARY on each of the database nodes is as follows:

```
HR91FP3_PRIMARY=
(DESCRIPTION_LIST =
  (FAILOVER=on)
(DESCRIPTION =
   (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
   (ADDRESS_LIST=
     (LOAD_BALANCE=on)
     (ADDRESS=(PROTOCOL=tcp)(HOST=scam08-scan3.us.oracle.com)(PORT=1521))
     (ADDRESS=(PROTOCOL=tcp)(HOST=scam02-scan7.us.oracle.com)(PORT=1521)))
     (CONNECT_DATA=
        (SERVER=DEDICATED)
        (SERVICE_NAME=PAYROLL_ONLINE)))
     )
```

This TNS connect string will only connect to the primary database as that is where the PAYROLL_ONLIN service can start. Do not place this TNS connect string alias onto any of the middle tiers. This should only be on each of the primary and standby database nodes so that the connect string used by the database link can be resolved.

The database link creation statement is:

```
CREATE DATABASE LINK PRIMARY CONNECT TO HR91FP3 IDENTIFIED BY HR91FP3 USING
'HR91FP3_PRIMARY';
```

PeopleTools requires a second TNS connect alias to the physical standby service. The PSQUERY service was defined earlier in this appendix. This connect string must be placed into all TNS names.ora accessible by the application domain server, and the alias name must be used for the StandbyDBName in the PSADMIN utility. The following is the TNS connect string alias called PSFTADG that is used in this project.

```
PSFTADG=
(DESCRIPTION_LIST =
  (FAILOVER=on)
  (DESCRIPTION =
    (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS=(PROTOCOL=tcp)(HOST=scam02-scan7.us.oracle.com)(PORT=1521))
      (ADDRESS=(PROTOCOL=tcp)(HOST=scam08-scan3.us.oracle.com)(PORT=1521)))
      (CONNECT_DATA=
         (SERVER=DEDICATED)
         (SERVICE_NAME=PSQUERY))
     )
```

On the primary database (scam08db03) a new row must be inserted into the PS.PSDBOWNER table so that the application server can authenticate with the Oracle Active Data Guard standby database. Do the following on the primary database using SQL*Plus:

```
sqlplus / as sysdba

SQL> INSERT INTO PS.DBOWNER VALUES ('PSFTADG', 'HR91FP3');

COMMIT;
```

The above INSERT statement is replicated to the standby database via redo transport. The application server configuration is now complete.

10.7 F5 BIG-IP Load Balancer

See Appendix A-9.5 for a discussion on implementing the F5 load balancer. The same was implemented at the DR site along with the health monitor.

11 Appendix C: Standby Site Test

This section describes how to validate that the standby site is ready to assume the pimary role in the event of a disaster. It is important to test the PeopleSoft application at the DR site to validate that all components are working properly. Because the Oracle database supports snapshot standby it is not necessary to shutdown the primary. While the physical standby is in the snapshot standby role, redo from the primary is still being received at the standby providing data protection. The outstanding redo is applied when the standby resumes the physical standby role. For the application and PIA components we simply need to create a sapshot of the ZFS replicated report repository and mount it. This allows for any site-specific configuration changes to be made without disrupting the production site.

Before following the procecures below, ensure that the application server domains and the PIA web servers are down at the standby site. The database should be in the physical standby role with Oracle Active Data Guard enabled.

CAUTION: Before you proceed with the standby site test, make sure that no production user or batch process can access the PeopleSoft applicatoin at the standby site. Ensure that the URLs (and DNS) do not accidentally route traffic to the standby site or those transactions will be lost when the database is reverted back to a physical standby.

11.1 Physical Standby to Snapshot Standby

Because we have Data Guard Broker configured, a single command will do the job of converting the physical standby to a snapshot standby. As discussed earlier, the snapshot standby allows the database to be opened read-write for testing. A guaranteed restore point is created so that when the testing is complete and the database is reverted back to a physical standby, all changes made during the testing are discarded. Once the database is open as a snapshot standby, the application can start as it normally would.

Before we convert the database, the following `srvctl` command shows the status of the services on the standby (payroll_s2):

```
[oracle_psft@scam02db07 FSFO]$ srvctl status service -d payroll_s2
Service PAYROLL_BATCH is not running.
Service PAYROLL_ONLINE is not running.
Service PSQUERY is running on instance(s) PAYROLL1,PAYROLL2
```

11.1.1 Convert the Physical Standby to Snapshot Standby

1. Log on to Data Guard Broker on either the primary or standby database.

```
[oracle_psft@scam02db07 FSFO]$ dgmgrl
DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production
Copyright (c) 2000, 2009, Oracle. All rights reserved.
Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/<password>
Connected.
```

```
DGMGRL> show configuration
Configuration - payroll_dgb
  Protection Mode: MaxPerformance
  Databases:
    PAYROLL_S1 - Primary database
    PAYROLL_S2 - Physical standby database
Fast-Start Failover: DISABLED
Configuration Status:
SUCCESS
```

2. Convert to snapshot standby.

```
DGMGRL> convert database 'PAYROLL_S2' to SNAPSHOT STANDBY;
Converting database "PAYROLL_S2" to a Snapshot Standby database, please
wait...
Database "PAYROLL_S2" converted successfully
DGMGRL>
```

3. Check the database services.

```
[oracle_psft@scam02db07 FSFO]$ srvctl status service -d payroll_s2
Service PAYROLL_BATCH is not running.
Service PAYROLL_ONLINE is not running.
Service PSQUERY is not running.
```

In order for the application servers to start, we need to start all services including PSQUERY on the snapshot standby.

```
[oracle_psft@scam02db07 FSFO]$ srvctl start service -d payroll_s2 -s
PAYROLL_BATCH
[oracle_psft@scam02db07 FSFO]$ srvctl start service -d payroll_s2 -s
PAYROLL_ONLINE
[oracle_psft@scam02db07 FSFO]$ srvctl start service -d payroll_s2 -s
PSQUERY
[oracle_psft@scam02db07 FSFO]$ srvctl status service -d payroll_s2
Service PAYROLL_BATCH is running on instance(s) PAYROLL1,PAYROLL2
Service PAYROLL_ONLINE is running on instance(s) PAYROLL1,PAYROLL2
Service PSQUERY is running on instance(s) PAYROLL1,PAYROLL2
```

11.1.2 Create a ZFS Snapshot of the Report Repository and Mount It

There are two replicas on scan03sn ZFS:

- peoplesoft_software_u01 exported as /export/peoplesoft/peoplesoft_software_u01
- peoplesoft_report_repository_scan04 exported as /export/peoplesoft/report_repository_test_scan03

We need the report repository to be read-write, and at present it is a replica that is amounted as a read-only file system. In order to have the process scheduler write to the report repository, we need to create a clone of the replica that is read-write. To do so follow these steps:

1. Log onto the ZFS BUI and naviate to the **Projects** on the left side and click **Replica**.
2. Select the **peoplesoft_report_repository_scan04** replica.
3. On the right side of the page there is a plus sign (+) with a tool tip that reads "Clone most recently received project snapshot". Click the plus sign.
4. Enter the project name peoplesoft_report_repository_test_scan03.
5. Check the override mount point and enter the mount point /export/peoplesoft/report_repository_test_scan03
6. Click **Continue.**
7. On each of the application servers, add the following into /etc/fstab:

```
172.17.0.9:/export/peoplesoft/report_repository_test_scan03
/peoplesoft/report_repository_test nfs4
rw,rsize=131072,wsize=131072,bg,hard,timeo=600
```

8. Mount the file system on all servers.

```
[root@scan03vm0031-eobi1 ~]$ mount /peoplesoft/report_repository_test
```

9. Move the log_output symbolic link to point to the snapshot file system.

```
[oracle_psft@scan03vm0031-eoib1 prcs]$ cd
$PS_CFG_HOME/appserv/prcs/HR91FP3
[oracle_psft@scan03vm0031-eoib1 HR91FP3]$ rm log_output
[oracle_psft@scan03vm0031-eoib1 HR91FP3]$ mkdir
/peoplesoft/report_repository_test/log_output
[oracle_psft@scan03vm0031-eoib1 HR91FP3]$ ln –s
/peoplesoft/report_repository_test log_output
```

11.2 Start the Application Domain Server and Process Scheduler on scan03vm0031-eoib1 and scan03vm0032-eoib1

We use the startAPP.sh and startWS.sh scripts described earlier.

```
[oracle_psft@scan03vm0031-eoib1 FSFO]$ ./startAPPS.sh
Tue Aug 20 21:53:32 PDT 2013
------ Starting Apps Server for domain: HR91FP3 ----
Attempting to boot bulletin board...
tmadmin - Copyright (c) 2007-2008 Oracle.
Portions * Copyright 1986-1997 RSA Data Security, Inc.
All Rights Reserved.
Distributed under license by Oracle.
Tuxedo is a registered trademark.
```

```
No bulletin board exists. Entering boot mode.
> INFO: Oracle Tuxedo, Version 10.3.0.0, 64-bit, Patch Level 043
Booting admin processes ...
exec BBL -A :
        process id=6232 ... Started.
1 process started.
Attaching to active bulletin board.
> Attempting to boot ...
INFO: Oracle Tuxedo, Version 10.3.0.0, 64-bit, Patch Level 043
Booting server processes ...
exec PSWATCHSRV -o ./LOGS/stdout -e ./LOGS/stderr -A -- -ID 257397 -D HR91FP3
-S PSWATCHSRV :
        process id=6236 ... Started.
exec PSAPPSRV -o ./LOGS/stdout -e ./LOGS/stderr -p 1,600:1,1 -s@psappsrv.lst
-- -D HR91FP3 -S PSAPPSRV :
        process id=6237 ... Started.
exec PSAPPSRV -o ./LOGS/stdout -e ./LOGS/stderr -p 1,600:1,1 -s@psappsrv.lst
-- -D HR91FP3 -S PSAPPSRV :
        process id=6245 ... Started.
exec PSAPPSRV -o ./LOGS/stdout -e ./LOGS/stderr -p 1,600:1,1 -s@psappsrv.lst
-- -D HR91FP3 -S PSAPPSRV :
        process id=6253 ... Started.
exec PSAPPSRV -o ./LOGS/stdout -e ./LOGS/stderr -p 1,600:1,1 -s@psappsrv.lst
-- -D HR91FP3 -S PSAPPSRV :
        process id=6261 ... Started.
exec PSAPPSRV -o ./LOGS/stdout -e ./LOGS/stderr -p 1,600:1,1 -s@psappsrv.lst
-- -D HR91FP3 -S PSAPPSRV :
        process id=6269 ... Started.
…
exec JREPSVR -o ./LOGS/stdout -e ./LOGS/stderr -A -- -W -P
/u02/app/local/oracle_psft/peoplesoft/ps_config/appserv/HR91FP3/jrepository :
        process id=6504 ... Started.
rocesses started.
```

11.3 Start the PIA Web Server on scan03vm0033-eoib1 and scan03vm0049-eoib1

```
[oracle_psft@scan03vm0033-eoib1 FSFO]$ ./startWS.sh
Attempting to start WebLogic Server PIA
No activity will be logged to this window.
Server activity will be logged to
/u02/app/local/oracle_psft/peoplesoft/ps_config/webserv/peoplesoft/servers/PI
A/logs/PIA_*
PID for WebLogic Server PIA is: 7390
```

11.4 Application Testing

At this point you can test the PeopleSoft application. Perform workload testing and run actual batch jobs to ensure everything is working properly. Make sure that all of the configurations are correct and all of the application components function properly.

11.5 Shutdown Application Servers and PIA Web Servers

When testing is complete, shut down the PIA web server and the application domain servers. The stopAPP.sh, stopPS.sh, and stopWS.sh scripts will bring down all components.

After the application and web services are down, reset the symbolic link to point back to the replicated report repository.

```
[oracle_psft@scan03vm0031-eoib1 prcs]$ cd $PS_CFG_HOME/appserv/prcs/HR91FP3
[oracle_psft@scan03vm0031-eoib1 HR91FP3]$ rm log_output
[oracle_psft@scan03vm0031-eoib1 HR91FP3]$ ln –s /peoplesoft/report_repository
log_output
```

11.6 Shut Down the Database Services.

```
[oracle_psft@scam02db07 FSFO]$ srvctl stop service -d payroll_s2 -s
PAYROLL_BATCH
[oracle_psft@scam02db07 FSFO]$ srvctl stop service -d payroll_s2 -s
PAYROLL_ONLINE
[oracle_psft@scam02db07 FSFO]$ srvctl stop service -d payroll_s2 -s PSQUERY
[oracle_psft@scam02db07 FSFO]$ srvctl status service -d payroll_s2
Service PAYROLL_BATCH is not running.
Service PAYROLL_ONLINE is not running.
Service PSQUERY is not running.
```

11.7 Convert the Snapthot Standby Back to Physical Standby

Use Data Guard Broker to convert the snapshot standby back to a physical standby database. The physical standby will rezume its role as an Oracle Active Data Guard standby.

```
[oracle_psft@scam02db07 FSFO]$ dgmgrl
DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production
Copyright (c) 2000, 2009, Oracle. All rights reserved.
Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/<password>
Connected.
DGMGRL> show configuration
Configuration - payroll_dgb
  Protection Mode: MaxPerformance
  Databases:
    PAYROLL_S1 - Primary database
    PAYROLL_S2 - Snapshot standby database
Fast-Start Failover: DISABLED
Configuration Status:
SUCCESS
DGMGRL> convert database 'PAYROLL_S2' to PHYSICAL STANDBY;
Converting database "PAYROLL_S2" to a Physical Standby database, please
wait...
Operation requires shutdown of instance "PAYROLL1" on database "PAYROLL_S2"
```

```
Shutting down instance "PAYROLL1"...
Database closed.
Database dismounted.
ORACLE instance shut down.
Operation requires startup of instance "PAYROLL1" on database "PAYROLL_S2"
Starting instance "PAYROLL1"...
ORACLE instance started.
Database mounted.
Continuing to convert database "PAYROLL_S2" ...
Operation requires shutdown of instance "PAYROLL1" on database "PAYROLL_S2"
Shutting down instance "PAYROLL1"...
ORA-01109: database not open
Database dismounted.
ORACLE instance shut down.
Operation requires startup of instance "PAYROLL1" on database "PAYROLL_S2"
Starting instance "PAYROLL1"...
ORACLE instance started.
Database mounted.
Database "PAYROLL_S2" converted successfully
DGMGRL> show configuration
Configuration - payroll_dgb
  Protection Mode: MaxPerformance
  Databases:
    PAYROLL_S1 - Primary database
    PAYROLL_S2 - Physical standby database
Fast-Start Failover: DISABLED
Configuration Status:
SUCCESS
```

11.8 Unmount the ZFS Snapshot on All Servers

On all vServers, unmount the ZFS snapthot of the report repository as follows:

```
[root@scan03vm0031-eobi1  FSFO]$ umount /peoplesoft/report_repository_test
```

It is optional to discard the clone however, it is likely that you will want to create a new clone for future testing.

12 Appendix D: Site Switchover

Site switchover is when the primary and standby change roles. This capability is very valuable for allowing business to continue at the secondary site while maintenance is performed at the primary site. The high-level steps for performing a switchover (or switchback) are:

1. Drain the process scheduler queues or place some jobs on hold.

   It is important that no jobs are running when attempting to swithover to the secondary DR site.

2. Shut down the PeopleSoft application domain and PIA web servers.

   Use stopAPPS.sh, stopPS.sh, and stopWS.sh scripts described earlier to shut down the application.

3. Perform a database switchover with Data Guard Broker.

```
[oracle_psft@scam08db03 FSFO]$ dgmgrl
DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production
Copyright (c) 2000, 2009, Oracle. All rights reserved.
Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/<password>
Connected.
DGMGRL> show configuration
Configuration - payroll_dgb
  Protection Mode: MaxPerformance
  Databases:
    PAYROLL_S1 - Primary database
    PAYROLL_S2 - Physical standby database

Fast-Start Failover: DISABLED
Configuration Status:
SUCCESS
DGMGRL> switchover to 'PAYROLL_S2';
Performing switchover NOW, please wait...
New primary database "PAYROLL_S2" is opening...
Operation requires shutdown of instance "PAYROLL1" on database
"PAYROLL_S1"
Shutting down instance "PAYROLL1"...
ORACLE instance shut down.
Operation requires startup of instance "PAYROLL1" on database "PAYROLL_S1"
Starting instance "PAYROLL1"...
ORACLE instance started.
Database mounted.
Database opened.
Switchover succeeded, new primary is "PAYROLL_S2"
```

4. Perform a role reversal of the report repository ZFS share where scan03sn is the primary and scan04sn is the standby replica. Follow the ZFS documentation for performing a role reversal of the PeopleSoft_repository_scan04 and the PeopleSoft_software_u01 shares.

   a. Make sure that the replication is stopped at the source.

b. On the target, find the replica and select the replica then click **Replication**.

c. Click the reverse direction icon.

d. Enter a new local project name if you are prompted to do so.

5. Start the PeopleSoft application and PIA web servers on the new primary (scan03 site) using the startAPP.sh, startPS.sh, and startWS.sh scripts.

   NOTE: You do not need to remount the file systems for the replicated shares following a role revresal. NFS is able to determine that the file system state on the new primary is now read-write, and on the new standby the file system is now read-only.

6. Perform a DNS push to propogate name resolution to the new primary. If you have an F5 BIG-IP load balancer this may be handled for you wth Global Traffic Manager.

   To switch back where scan04 resumes its original primary role and scan03 is the standby, follow the above steps but for switching the database over to payroll_s1 and the replication of the shares from scan04sn to scan03sn.

13 Appendix E: Site Failover

For a site failover scenario we assume that the primary site is unavailable and completely inaccessible. In this scenario the standby assumes the primary role. To do so, the following steps are performed.

1. Push new DNS entries for the new primary for client user access. If an F5 BIG-IP load balancer is used, this may be achieved with Global Traffic Manager inside an enterprise network cloud. For customer-facing services and B2B interfaces this may require the network administrators to push routing rules to the various point of presence (POP) to affect the change globally.

2. In Data Guard Broker issue a failover command to cause the standby database to become the new primary. In this case payroll_s2 will become the new primary. Because the Data Guard protection mode is Maximum Performance there may be a small amount of data loss. The failover will take seconds to just a few minutes. Log into Data Guard Broker on the standby. The `show configuration` command will show errors. Perform the failover command as shown below.

```
DGMGRL> show configuration
Configuration - payroll_dgb
  Protection Mode: MaxPerformance
  Databases:
    PAYROLL_S1 - Primary database
    PAYROLL_S2 - Physical standby database
Fast-Start Failover: DISABLED
Configuration Status:
ORA-16625: cannot reach database "PAYROLL_S1"
DGM-17017: unable to determine configuration status

DGMGRL> failover to 'PAYROLL_S2';
Performing failover NOW, please wait...
Failover succeeded, new primary is "PAYROLL_S2"
```

   The standby atabase is now the primary.

3. On ZFS, any project that has continuous replication to the replica at the standby may have a "failed" status due to lack of connectivity to the primary. You can still use the Reverse Direction action and cause the replica to become a local project and export for read-write. There is no need to remount the file system because NFS detects the state change of the file system. The steps for doing this are the same as for site switchover in appendix D.

4. On the standby vServers (now, the new primary) start the application using startAPP.sh, startPS.sh, and startWS.sh scripts.

   NOTE: Be sure that the PSQUERY database service is started on the new primary. In this project the PSQUERY service was defined for both PRIMARY and PHYSICAL_STANDBY database roles. Once the failover completed, the PSQUERY service was started automatically. If this service is not started, the PeopleSoft application domain servers will fail to start.

14 Appendix F: Site Reinstantiation

Recovery of the primary site after a failure depends on the nature of the failure. For a complete catastrophic lost or severe damage to the primary data center facility, this requires rebuilding and/or repairing that data center or relocating to new facilities. New systems must be provisioned and configured. Oracle Exadata Database Machine comes pre-configured to reduce a significant amount of the setup time.

If the primary site was lost due to a prolonged network outage or power outage, but the facility remained intact, once the infrastructure services have been restored it is then just a matter of synchronizing the old primay site and bringing it online as a standby. This is the scenario for this case study.

ZFS replication needs to be resumed using the ZFS BUI as described earlier.

14.1 Reinstate the PeopleSoft Database

To reinstantiate the old primary database the following steps were performed.

1. Bring up the Grid Infrastructure and databases.

   Once the power and/or network have been restored, bring up the cluster if it is not already up. Ensure that the ASM instances are up. If the systems are powered up the database nodes will start Cluster Ready Services (CRS) then join and form the cluster. In turn, the cluster will start up ASM, the listeners, VIPs, and databases.

   Because our database is under Data Guard control, , once it starts, Data Guard notices that it is up and then takes control and keeps it from opening as a primary. At this point we can use Data Guard Broker to reinstantiate the database as the new standby. If flashback was not enabled, then a full restore from a backup (or from the primary using RMAN duplicate) would be necessary to reinstantiate the standby database.

   If the database on the old primary (scam08db03) has not yet been started, start it with the following command:

   ```
   srvctl start database –d payroll_1
   ```

   Let it start. You will see errors:

   ```
   Data Guard: version check completed
   Data Guard determines a failover has occurred - this is no longer a
   primary database
   ORA-16649 signalled during: ALTER DATABASE OPEN
   ```

2. On one of the new primary Exadata database nodes (scam02db07), run DGMGRL and display the configation:

   ```
   DGMGRL> show configuration
   Configuration - payroll_dgb
     Protection Mode: MaxPerformance
   ```

```
  Databases:
    PAYROLL_S2 - Primary database
    PAYROLL_S1 - Physical standby database (disabled)
      ORA-16661: the standby database needs to be reinstated
Fast-Start Failover: DISABLED
Configuration Status:
SUCCESS
```

3. From the Broker reinstate payroll_s1 as the new standby:

```
DGMGRL> reinstate database 'PAYROLL_S1';
Reinstating database "PAYROLL_S1", please wait...
Operation requires shutdown of instance "PAYROLL1" on database
"PAYROLL_S1"
Shutting down instance "PAYROLL1"...
ORA-01109: database not open
Database dismounted.
ORACLE instance shut down.
Operation requires startup of instance "PAYROLL1" on database "PAYROLL_S1"
Starting instance "PAYROLL1"...
ORACLE instance started.
Database mounted.
Continuing to reinstate database "PAYROLL_S1" ...
Reinstatement of database "PAYROLL_S1" succeeded

DGMGRL> show configuration
Configuration - payroll_dgb
  Protection Mode: MaxPerformance
  Databases:
    PAYROLL_S2 - Primary database
    PAYROLL_S1 - Physical standby database
Fast-Start Failover: DISABLED
Configuration Status:
SUCCESS
DGMGRL> show database 'PAYROLL_S1';
Database - PAYROLL_S1
  Role:            PHYSICAL STANDBY
  Intended State:  APPLY-ON
  Transport Lag:   0 seconds
  Apply Lag:       0 seconds
  Real Time Query: ON
  Instance(s):
    PAYROLL1 (apply instance)
    PAYROLL2
Database Status:
SUCCESS
```

4. Migrate the PSQUERY database service to the standby.

The PSQUERY database service is started when the database is in the role of a physical standby. On the compute nodes, scam08db03 and scam08db04, the PSQUERY should be up. If not, start the PSQUERY service:

```
srvctl start service –d payroll_s1 –s PSQUERY
```

Once it is started, you can shut down the PSQUERY on the primary and users running reports and queries against that service will migrate back to the Oracle Active Data Guard standby database.

```
srvctl stop service –d payroll_s2 –s PSQUERY
```

14.2 Reinstate ZFS Replicated Shares

Using the BUI, restart or resume continuous replication of Peoplesoft_Repository_scan04 back to the standy site scan04sn. You should also peform a manual update of the peoplesoft_software_u01 project/share at this time.

Ensure that both shares are exported on the ZFS and ensure that the middle tier servers can mount the file systems. They should be in a read-only state.

Check the directory owner and groups to ensure they are correct. If not, it is possible that the NIS server is not up yet and needs to be restarted.

During a sheculed maintenance window perfom the Site Test described in Appendix C to validate that the application starts and functions correctly.

15 Appendix G: PeopleSoft Upgrades and Patching

Both Oracle Exadata Database Machine and Oracle Exalogic hardware infrastructure have provisions for reducing downtime for a variety of maintenance activities such as patching and upgrades. For Oracle Exadata Database Machine, the following are activities that can be performed with no down time for any running application:

- Rolling storage cell patching or upgrades
- Cell disk replacement or entire cell repair[4]
- Rolling IB and Ethernet switch maintenance and firmware upgrades

The following maintenance activities can be performed with no down time for PeopleSoft:

- Rolling database node OS kernel patching or upgrades
- Rolling grid infrastructure patching or upgrades
- Oracle RAC rolling database bundle patch application
- Out-of-Place patching of PS_HOME or PS_APP_HOME[5]

15.1 Case Study Example: Applying Database Bundle Patch

The following is a specific example of applying Exadata bundle patch 20 for Oracle Database 11.2.0.3 while PeopleSoft was running a workload of 1,000 online HR users doing various tasks. No users encountered any errors during the entire process.

The bundle patch application workflow was performed as follows:

- Apply bundle patch 20 to the standby database first (Standby-First)
- Apply bundle patch 20 to the primary database using Oracle RAC rolling upgrade

In the following example the patch set required was pre-staged and all patch prerequisites have been met. The PeopleSoft application was up and running.

[4] Subject to the available amount of required mirror free to restore redundancy of ASM disk groups. Please see My Oracle Support note 1551288.1

[5] Out-of-place patching depends on the nature of the patch. If the patch being applied to the PeopleSoft stack is compatible with the existing code, the patch can be applied to a cloned PS_HOME and servers can be shut down and restarted in a rolling manner on the new PS_HOME.

15.2 Standby First Bundle Patch Apply

Because PeopleSoft is configured wth Oracle Active Data Guard we need to apply the bundle patch in an Oracle RAC rolling manner. The database unique name for the standby is payroll_s2.

The steps to apply the patch are:

1. Stop and disable the PSQUERY on the PAYROLL1 RAC instance of payroll_s2.

```
srvctl stop service -d payroll_s2 -s PSQUERY -i PAYROLL1

srvctl disable service -d payroll_s2 -s PSQUERY -i PAYROLL1
```

2. Stop and restart instance PAYROLL1.

```
srvctl stop instance -d payroll_s2 -i PAYROLL1 -o immediate

srvctl start instance -d payroll_s2 -i PAYROLL1
```

3. Check that the connections went to the second instance. We know that the server name connections are from the scan04 Oracle Exadata Database Machine.

```
sqlplus / as sysdba
  SQL> select inst_id,count(*)
  from gv$session
  where program like '%scan04%'
  group by inst_id;
   INST_ID   COUNT(*)
---------- ----------
         1          0
         2          15
```

   Instance 1 has no connections.

4. Apply the patch:

```
 opatch auto
/u01/app/oracle_psft/product/11.2.0.3/dbhome_psft/patches/11203_BP20/16869
210 -oh /u01/app/oracle_psft/product/11.2.0.3/dbhome_psft
```

5. Check for any errors in the patch log files and correct any issues as necessary.

6. Re-enable the PSQUERY service on the PAYROLL1 instance and start the service. Note that the instance is already up.

```
srvctl enable service -d payroll_s2 -s PSQUERY -i PAYROLL1

srvctl start service -d payroll_s2 -s PSQUERY -i PAYROLL1
```

7. Repeat the above steps for the second instance, PAYROLL2, replacing PAYROLL1 with PAYROLL2.

8. Test and validate the patch by running the standby for a period of time (up to 7 days if necessary). Some of the PeopleSoft Application Engine processes, Query Viewer, XML Plublisher, and Tree Viewer will use the Oracle Active Data Guard database with this bundle patch in place.

   NOTE: The catbundle.sql script has not yet been run. This will be done later.

15.3 Primary RAC Rolling Bundle Patch Apply

On the primary database stop the PAYROLL_ONLINE and PAYROLL_BATCH services and disable them, stop the instance for each node we patch, then re-enable and restart them, one node at a time, one node after another.

1. Stop and disable the PAYROLL_ONLINE and PAYROLL_BATCH services.

```
srvctl stop service -d payroll_s1 -s PAYROLL_ONLINE -i PAYROLL1

srvctl disable service -d payroll_s1 -s PAYROLL_ONLINE -i PAYROLL1

srvctl stop service -d payroll_s1 -s PAYROLL_BATCH -i PAYROLL1

srvctl disable service -d payroll_s1 -s PAYROLL_BATCH -i PAYROLL1
```

2. Shut down and restart the PAYROLL1 instance.

```
srvctl stop instance -d payroll_s1 -i PAYROLL1 -o immediate

srvctl start instance -d payroll_s1 -i PAYROLL1
```

3. Check that the connections went to the second instance. We know that the server nameconnections are from the scan04 Oracle Exadata Database Machine.

```
sqlplus / as sysdba
  SQL> select inst_id,count(*)
  from gv$session
  where program like '%scan04%'
  group by inst_id;
   INST_ID   COUNT(*)
---------- ----------
         1          0
         2          67
```

   Instance 1 has no connections.

4. Apply the patch:

```
opatch auto
/u01/app/oracle_psft/product/11.2.0.3/dbhome_psft/patches/11203_BP20/16869
210 -oh /u01/app/oracle_psft/product/11.2.0.3/dbhome_psft
```

   The instance will be restarted by opatch.

5. Check for errors in the patch log files and correct any issues as necessary.
6. Re-enable and restart the PAYROLL_ONLINE and PAYROLL_BATCH services.

```
srvctl enable service -d payroll_s1 -s PAYROLL_ONLINE -i PAYROLL1

srvctl start service -d payroll_s1 -s PAYROLL_ONLINE -i PAYROLL1

srvctl enable service -d payroll_s1 -s PAYROLL_BATCH -i PAYROLL1

srvctl start service -d payroll_s1 -s PAYROLL_BATCH -i PAYROLL1
```

7. Repeat the above steps for the second instance, PAYROLL2, replacing PAYROLL1 with PAYROLL2.
8. Once the bundle patch has been applied for the PAYROLL2 instance, run the catbundle.sql script.

```
cd $ORACLE_HOME

sqlplus / sysdba

@rdbms/admin/catbundle.sql exa apply
```

   The bundle patch process is complete.

Oracle White Paper Title
February 2014
Author: Darryl Presley
Contributing Authors: Lyn Pratt, Richard Exley, Ray Dutcher

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

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

Hardware and Software, Engineered to Work Together