

Oracle RAC on 11i EBS

Sweating the Details

February 2008

kurt.forshee@gmail.com

Cell: (214) 334.8582

Prepared by: Kurt Forshee

TABLE OF CONTENTS

Table of Contents	ii
Introduction	1
Understanding the Terminology of RAC	2
The RAC Business Case	2
For 11i EBS, a Better Business Case	3
The RAC Architecture	3
Preparing for RAC.....	4
RAC Technical Considerations	5
Technical Drill-down	6
11i EBS Considerations	7
Software configuration	9
Additional Recommendations.....	13
Short Glossory of Terms.....	15

RAC on 11i EBS: Sweating the Details

A Practical Guide to What You Need to Know

Kurt Forshee

INTRODUCTION

RAC is complex technology affecting levels deeper than setting a database parameter. Decision-makers need to know that network, storage, and server decisions play an important part in the success of RAC architecture. Also, along with the complexity of the architecture comes the increased responsibilities of all the support and administrative staff. Also note the investments required of both time and money to build, test, and support the commitment to RAC.

I want to present some foundational information and best practices about this architectural decision, discussing the foundation of RAC, the operational side of RAC, and the benefits of the decision to use this architecture and technology.

RAC, in its full breadth, will influence architecture, middleware, applications, databases, business processes and create new challenges in most organizations. For that reason it is an important topic to anyone who is involved with Oracle's Enterprise business applications.

UNDERSTANDING THE TERMINOLOGY OF RAC

RAC is the acronym for “Real Application Clusters”. This term normally applies to physical servers joined in a technology configuration that allows all servers in the cluster to share the same resources and failover. To Oracle, this means multiple servers sharing the same file structure so multiple database instances can be opened.

Clusterware normally refers to Oracle’s Clusterware software. This is software installed on RAC nodes that allows Oracle to share and control infrastructure services, including the servers, network services for

ASM is Oracle’s Automatic Storage Management, an oracle instance dedicated to deliver volume manager and file system with raw disk performance.

For a short glossary of other terms, refer to the section at the end of this document.

THE RAC BUSINESS CASE

Oracle’s intent for RAC appears to be the scalability solution in a infrastructure dedicated to exploiting commodity-class hardware, but some sales literature state additional goals. It is my opinion that RAC has always been a scalability option first, with benefits of high-availability and performance contingent on other factors, such as workload and stability.

Truly, Oracle’s RAC vision allows the larger companies committed to Oracle to realize hardware costs savings while still being able to scale their applications as needed. Oracle RAC also allows smaller companies to begin building their infrastructure on a small footprint and scale outward as they grow.

It should be noted that when possible, scaling vertically (larger servers) is mostly the best scalability option and should be considered first in most scenarios. Due to the costs and complexity of configuring clustered servers with Oracle RAC, this option should only be considered after other alternatives have been eliminated, such larger servers, performance tuning, workload management, etc.

Note that Oracle seems aware that many customers are frustrated with the experience of managing, upgrading and patching technology. The continuous cycle of projects to maintain their business applications and the associated environment is a significant cost to the bottom-line.

For those committed to the technology with a strong architectural foundation to support Oracle RAC for running the 11i EBS, it can truly be the difference maker for most organizations.

Here’s a quick excerpt from Gartner regarding RAC: Findings from Oracle OpenWorld 2006: Users Are Investing in Oracle’s DBMS Infrastructure -- 15 December -- Mark A. Beyer

“...organizations unwilling to increase their commitment to Oracle will be initially reluctant to implement the solution. Also, the enterprise database management group subsumes some storage management duties, and many organizations are incapable or unwilling to do this. As the role of the database administrator expands, users must have an implementation strategy that resolves cultural, as well as technical, shifts. No technology solution is a "silver bullet." RAC and ASM require solid planning, justification and implementation. ”

FOR 11i EBS, A BETTER BUSINESS CASE

11i EBS is not able to take advantage of some of the advertised features that RAC provides, such as rolling upgrades and Transparent Application Failover (TAF). Whereas the database can be patched in a rolling upgrade fashion, the applications cannot. Due to the constraints of the 806 ORACLE_HOME tools, TAF is not possible (Note: TAF can be realized with the java based applications).

Despite missing out on some of the High Availability features, RAC benefits 11i EBS users by providing scalability and allowing OLAP functions to be off-loaded to non-OLTP servers. Adding RAC nodes to a busy OLTP system helps if the batch and OLAP processing does not negate the features of RAC by crowding the interconnect by implanting Parallel Concurrent Processing (PCP) and dedicating rac nodes to Discoverer only. Without Parallel Concurrent Processing and offloading Discoverer user processing, RAC would in fact be prohibitive in 11i EBS.

ASM's use of raw disk is possibly the biggest improvement in performance, while global cache (gc) wait events can sometimes seriously degrade performance worse than single node databases.

THE RAC ARCHITECTURE

Today, most Oracle database implementations rely on a physical database layer. There are both benefits and implementation costs in implementing and maintaining this structure, but much of the RAC Architecture will rely on abstraction.

RAC Architecture changes the concepts of physical server control, physical listeners, and storage access and performance. Before, servers and services were started and shutdown by sysadmins; Oracle Clusterware controls much of that. Listeners were defined based on public IP address; now Virtual IP's are started and stopped by Clusterware, and the listeners are defined to use the VIP's rather than public IP's. Storage administrators would configure stripe & mirrored luns and file systems would be broken out into mount points for datafiles, indexes, controlfiles, etc, but ASM eliminates that complete utilization is no longer determined by doing a "df" or a "du" command, but via OEM or asmcmd to check ASM disk groups.

This abstraction puts a lot more administrative power into the hands of the DBA since Clusterware and ASM change the rules. However, there is a different technical foundational level to be made available for this shift to occur successfully.

PREPARING FOR RAC

With all of this information plus that available from Oracle, the hardest step is the understanding how to build an operational foundation to succeed using RAC Technology. Here are some steps that will simplify that decision.

Understand your Business Requirements	Well defined SLA's
Rethink your architecture strategy to maintain continuity for entire enterprise	Introduce new concepts to all architectural components related to all Oracle instances, like using ASM and RMAN even in all non-RAC instances
Ensure a high-availability foundation	Shared resources, clusterware, database HighAvailability configuration must all be understood and tested with the entire team
Upgrade as needed to ensure Supportability	Stay on latest Oracle patchsets of Clusterware and RDBMS to ensure all possible bug fixes and performance improvements are in place. Also keep 11i EBS technology patches up to date to ensure latest admin tools are available
Start building RAC maturity in your organization	Understanding the foundations of success and failure of the underlying architecture will ensure success
Utilize additional tools for monitoring and administration	Good time to start Oracle OEM integration

RAC TECHNICAL CONSIDERATIONS

RAC Architecture is defined mostly by the services that are installed and configured on several physical servers. Hardware, storage and network configuration options must support the Oracle Clusterware requirements, which in turn will support the Oracle RAC instances.

An organization that separates these roles will need to coordinate all the teams to produce the required stable infrastructure before Oracle software install can begin. A technical readiness assessment is probably the best way to get started, reviewing all the current operational standards and SLA's for server, network and storage administration. Once these elements are identified by the technology infrastructure staff, the Oracle RAC project team can begin to ask for their requirements, citing the details of each.

Remember to include the costs associated with each part of the infrastructure. More network and additional servers incur costs, and may mean more staff; additional RAC complexity at the database level may require additional dedicated resources to administer, including servers and services to monitor. Determine the underlying costs of RAC, then add the known costs, and spend time to determine if the benefits realized from the architectural change are worth the costs.

Hardware and storage platforms are extremely important when considering support levels and configuration options. Proprietary server architecture running unix platforms HPUX, AIX, and Solaris present much different options for building a shared configuration to support RAC than does commodity hardware Linux. The same can be said for high-end SAN's being chosen over commodity NAS. If an organization is committed to proprietary hardware and storage, the rules for RAC configuration are not as defined, nor supported, but Oracle, especially in the case where other software is used, such as 3rd party clusterware (ie: HP ServiceGaurd) or file management (Veritas).

The infrastructure decision affects the outcome of the technology deployment, and dictates the need for additional expertise and training. Even in the most stable environments on older, established platforms, the testing scenarios for failures and performance need to be comprehensive to provide the insight into the integration required by RAC.

TECHNICAL DRILL-DOWN

Hardware

Linux is the most documented platform for RAC configuration. It provides the most hardware offerings from different vendors, and Oracle provides the best support for this.

However, other vendors provide better support for their platforms, giving them the edge on server stability, even though Oracle support is lacking. Hence, the first question that needs asked when on a non-Linux platform is “why RAC?”. Scaling a non-Linux server usually means buying a bigger or newer box with bigger or faster processors, while fail-over is achieved well by using Oracle’s Standby database features.

In any case, RAC instances normally require additional memory on the database server for buffer cache than non-RAC, so all existing server memory will need analysis before making configuration changes. In RAC, space is allocated for the Global Cache Service (GCS) in every block buffer cache. The amount of memory required depends on how the application accesses the data – i.e. if the same block is cached in more than one instance. Switching to Automatic Shared Memory Management (ASMM) will let Oracle determine the amount to increase the buffer cache, but the server will need to have the memory available.

Storage

Storage Platform is of main concern here for cost reasons. Most storage arrays offer many of the same features for stripe and mirrors, and many have an extensive disk cache ability, but shared connectivity is our main concern for RAC. NAS and iSCSI are best suited for sharing, and receive my highest recommendation for RAC architecture because of the technology fits together seamlessly. SAN is often expensive and even cost-prohibitive in sites dedicated to commodity hardware, and does not have the options available for sharing without additional components, such as fibre-channel.

In all platforms, the Stripe-and-mirror-everything (SAME) recommendation applies. Although ASM will stripe the data also, stripe-on-stripe is not a bad thing here.

Raw is fastest access with Oracle, making it preferable over Clustered File System (CFS). ASM particularly wants to be on shared raw/block volumes, or logical raw volumes if using 3rd party clusterware, and the speed of the physical read-writes via ASM makes this a very desirable configuration. Of course, clusterware’s registry (OCR) and voting disk need to be on sharable raw volumes as well.

Network

1Gb Ethernet is the current standard for network connectivity; 10Gb and Infiniband are quickly gaining market share to make network latency non-existent. With Oracle RAC, we want to ensure that the interconnect between RAC instances does not exceed 70% utilization, something that will rarely be achieved in 1Gb NIC cards.

Redundant network components, such as NIC cards and switches, ensure reliability. There are basic configuration requirements at the Unix/Linux levels to enable the redundancy and sharing across multiple cards.

Three IP addresses need to be assigned to each server: Public IP, Private IP (interconnect), and Virtual IP (VIP). These IP’s are assigned to the NIC ports, except for the VIP, which is started and controlled by Oracle Clusterware so it can be moved in the event of a server failure.

11i EBS CONSIDERATIONS

Patching to the latest ATG RUP Levels provide the best toolsets for administering the 11i tech stack.

11i Topologies usually consist of separate RDBMS and 11i EBS tiers. The 11i EBS tiers can be segmented into Web, Forms, Admin, and concurrent processing (ccm) node, but there is absolutely no technical or functional benefit to separate these tiers except for capacity. In the past, Oracle recommended sharing the admin/ccm nodes; again, there was real benefit to this configuration, but for RAC, you should no longer share admin and db tiers. To enable Parallel Concurrent Processing (PCP), you would want to have at least two apps tiers running concurrent managers and the FNDDFS Listener.

Mixed use 11i EBS instances supporting both OLTP end users and OLAP business reporting should have a separate node for Discoverer (10gAS) so that a separate TNS entry can be used.

It's also wise to consider the customizations that have been done prior to RAC conversions. Certain database utilities, such as utl_file, are not support by 11i EBS, and the use of these create large issues that need resolved before moving forward. In the case of utl_file, are you able to control and administer delivery of the output by controlling which Oracle RAC node will be running the code and generating the output? Also, poor performing custom code may become awfully performing custom code in RAC.

RAC node requirements

The Oracle RAC instance's steady state is maintained by Oracle Clusterware. In order for Oracle Clusterware to maintain a healthy status, it must be part of a quorum; meaning that the Oracle RAC instance cannot be healthy if it cannot heartbeat another RAC instance in the cluster. In a 2-node cluster, if heartbeat fails, Oracle RAC instance is not sure why, and checks the quorum partition to find there is a timestamp from other nodes, Oracle has no way of knowing which node has the failed heartbeat (or interconnect), and both nodes will fail. In a 3+ node cluster, if heartbeat fails to one node, the surviving nodes able to heartbeat each other will remain alive, and only the node having the issue will be ejected from the cluster. Hence, a quorum requires at least 3-nodes for failover.

Since I've stated before that Oracle RAC is not primarily a failover option as much as it is a scalability option, 2-node RAC configurations are acceptable for separating work load as mentioned before, but can not server as failover solution.

I always recommend a minimum of 3-node RAC so that, in the case of a failed server, both production instances of the database would be unusable.

11i Mixed Use database

11i implementations often use the Database for both end-user OLTP and OLAP business reporting. These separate functions tend to work against each other to produce a very poor-performing database. Although there are business reasons to not separate the database into two physical instances, RAC can allow the two types of processing to co-exist against the same database, but using different instances. This allows OLAP processing instance to be tuned specifically for a separate workload than the on-line users. This is, in my opinion, the best use of RAC in 11i EBS.

Normally, long running processes that consume database resources are controlled via the concurrent manager; Discoverer processes, ad-hoc queries, and other OLAP functions are not under concurrent manager control. RAC allows for the segregation of resources for these functions.

When SLA's require the OLAP instance be available for 24x7, then I recommend additional nodes to the RAC cluster to support this instead of allowing OLAP to begin consuming OLTP server resources in the event of failure.

Instance Strategy

For RAC, a fully functioning "model" of the production clustered environment needs to be available for all testing and troubleshooting. All changes to the PROD system should go through this model office environment and be fully tested before being introduced to PROD.

However, all other instances in the organization do not require RAC configuration since application functionality is not often affected by RAC. Although I recommend ASM as an enterprise standard, RAC is only necessary in PROD and near-PROD as an integration testing step before moving changes into production environment.

Total Commitment to testing

Once RAC is adopted, integration testing all changes before they are moved to PROD is essential. Load and Failover tests are often necessary after PROD is in RAC if changes are introduced that change any configuration or database utilization pattern. It is not a good idea to enable a new module or responsibility to the 11i EBS in RAC configuration without doing some thorough testing to see if the changes will affect other user processes negatively.

SOFTWARE CONFIGURATION

O/S

Oracle gives all the detailed instructions for O/S patchset levels and kernel parameters for each hardware platform. It is essential to be at or above each parameter listed. In some cases, patching or setting configuration parameters beyond the recommended values is a good idea.

Clusterware

Third-party clusterware is the biggest responsibility at this level since it must integrate properly with Oracle Clusterware to succeed. I recommend 3rd party clusterware because it handles NIC card failover, I/O Fencing, and storage for presenting Logical Raw Volumes, but I would insist the organization has on-site staff experts to administer and configure this. Without on-site staff, I would stay with Oracle Clusterware only and not allow any additional software to interfere with the operations of the software.

If the staff is available, I usually defer the DBA tasks of installing and configuring Oracle Clusterware to the sysadmins responsible for the 3rd party clusterware. This eliminates the need for the DBA to be responsible for root owned processes that need to startup/shutdown software in the correct order to be sure shared services are available before Oracle Clusterware starts.

If the choice of 3rd party clusterware is certified with Oracle Clusterware and is configured correctly, it is a great benefit to the organization since it performs its roles better than Oracle's Clusterware. The biggest advantage is that 3rd party clusterware runs in kernel mode, while Oracle Clusterware runs in user mode; in the event of a server overload (processes freeze), 3rd party clusterware can still do its job of I/O Fencing, but Oracle Clusterware is part of the user kernel processes sitting frozen. It's a great help when done well.

Oracle Clusterware install belongs in a separate ORACLE_HOME. Once installed, most commands to start and stop the services will need to be run as root user, so sysadmins will need to grant sudo access for these commands.

The Cluster Verify Utility (cluvfy) is provided by Oracle to verify the setup for Clusterware. CLUVFY is not going to tell you if Clusterware will work, it will only tell you that the setup is ready for Clusterware

Oracle clusterware installer and upgrade to 10.2.0.3 will push the Oracle Software stack out to all nodes in the cluster, usually identified by the Oracle Universal Installer (OUI).

Data

The Datafiles for the database need to be accessible by all nodes of the cluster. Although a Clustered File System (CFS) can be used via 3rd party clusterware or Oracle's Cluster File System (OCFS) on Linux, this is not my recommendation for many reasons. CFS is slower with A LOT of overhead than providing physical raw devices for ASM instance.

Automatic Storage Management

This is the best practice for Oracle RAC. This layer of abstraction to the raw file partitions provides a fastest Direct I/O file environment for sharing files between Oracle RAC instances. ASM is an Oracle instance running on each node of the cluster, preferably installed in a separate ORACLE_HOME. Since 11i EBS autoconfig will create a

listener for the RDBMS, it is recommended to create a separate ASM listener (ASMLISTENER_<servername>) for the ASM instance.

When creating ASM diskgroup, it is usually best practice to use external redundancy when using a storage lun that is mirrored. If using storage that is not mirrored, ASM allows for creating disk groups and mirrors internally.

The ASM instance is specifically configured for Oracle datafiles, logfiles (online and archive), and controlfiles, and is recognized and available by any Oracle RDBMS instance without any additional configuration requirement.

Like Clusterware, ASM installation will push the Oracle binaries out to all nodes in the cluster.

ASM Operations

ASM files are not physical files, but are aliases to metadata. RMAN can copy and backup the data, restoring the data to another ASM instance or to a filesystem files. There are filesystem utilities to view the datafiles: `asmlib` for linux, and `asmcmd` on other platforms.

When datafiles are created in ASM, they use Oracle Managed File defaults, which name the files under specified directories for the instance. For instance, the datafile for GLD would be `+ASM/prod/datafile/gld.1.9999`. Creating the datafile is a simple command: “create tablespace `XXKWF` datafile `+DATA`” or “alter tablespace add datafile `+DATA`”, and ASM defaults the file to `autoextend unlimited`. Since datafiles may be restored to filesystems with datafile size limits, creating files in ASM may require you to use file sizes and `autoextend maxsize`.

The issues with ASM arise in watching the database for growth. Normally, you could view file sizes and percentage used; with ASM, you need to query the `v$asm_disk` table. Also, you can use OEM to watch growth.

Using separate diskgroups for Data and Flash Recover Area (FRA) are helpful in separating data from online redo and archivelogs. Since monitoring size and growth is a challenge, be sure to use RMAN to backup the archivelog files several times daily with the “delete input” option to make sure archivelogs do not fill up the diskgroup.

ASM constantly “rebalances” the data by moving heavily accessed blocks to faster access storage. When adding disks to ASM diskgroups, be sure they are in the same increments as the previously added disks, since Rebalance process will move data around when disks are added.

ASM and the ASMListener should be under CRS control, started and stopped by CRS. This ensures that ASM is up before starting the database.

ASM Caveats and alerts

Only one ASM instance per server; multiple databases need to share the same ASM instance. Also, physical standby running on a file system instead of ASM may have issues when new files are added to the primary instance, or when `utl_file` related operations are ASM based.

RAC Database configuration

Normal Oracle RAC configuration issues arise with 11i EBS, except that the parameter file will be overwritten by 11i’s autoconfig process, so be sure to keep any customizations in the ifile.

ASM Data Migration

Best practice is to use RMAN to copy data into ASM from the file system, but if your 11i EBS datafiles have not yet been moved to Oracle Application Tablespace Model (OATM), now is the time. Using the OATM utility to migrate all data is easy to do, and on a good storage platform, very fast.

The RMAN commands to copy the data from file system into ASM are simple:

```
SQL> alter system set db_create_file_dest = '+DATA'; (or set this in init.ora)

SQL> alter system set db_create_online_log_dest_1 = '+DATA';

SQL> alter system set db_create_online_log_dest_2 = '+FRA';

RMAN> startup nomount;

RMAN> restore controlfile from '/u01/prod/data/ctrl01.dbf';

RMAN> Backup as copy database format '+DATA';

RMAN> Switch database to copy;

RMAN> alter database open;

SQL> alter tablespace TEMP add tempfile size 1000M autoextend off; (don't set TEMP to autoextend)

SQL> alter database add logfile group 99 size 100M;

SQL> alter system switch logfile;

SQL> alter database drop logfile group 1;
```

RAC configuration

Use adcfgclone to recreate controlfile and build environment with RAC specific settings, such as pfile and listener. The rconfig option is documented, but is not as intuitive for the experienced apps DBA familiar with autoconfig.

Adding RAC instances in 11i EBS after the first database instance is configured, running adpreclone and copying the \$ORACLE_HOME/appsutil directory from node1 to all other RAC nodes will allow adcfgclone.pl to be used to configure the instances. After setting the init.ora parameter "cluster = TRUE" in node1, creating an additional logfile thread, an UNDO tablespace for the new thread, and enabling the new thread for each new RAC instances will lay the foundation to create additional RAC instances via adcfgclone.

11i EBS Apps Tier configuration

With RAC nodes configured and entries in fnd_nodes and fnd_database_instances, running autoconfig on the apps tiers will create the correct tnsentries for failover and load balance. The cp_two_task entries in the context file should not be set to the 806_balance entry if Parallel Concurrent Processing is being used. On one apps tier, setting cp_two_task to DB1, and on 2nd apps tier, setting cp_two_task to DB2 will allow concurrent manager queues to be defined on specific database nodes, and will also failover to 2nd apps tier if the queues are defined properly.

Operational Advice

Monitoring the services running on the system for RAC is essential. Normally, just watching database performance can tell you if something is acting up with the RAC config, especially if you begin to see a lot of global cache (gc) wait events or locks. Be sure to continually monitor the database performance.

Backups / Recovery / Clone

RMAN is the main tool to use with ASM for backups and recovery, and should be used whether ASM is in place or not.

The OCR and Voting disks should be backed up regularly in case of failure. By default, Oracle will backup the OCR daily, but you can manually back up the voting using a “dd” command.

Cutover Options

Every project task timeline is constrained by downtime limitations. Reducing the downtime requirements for cutover to RAC architecture is sometimes difficult.

If the business allows for a longer downtime period, the “Big Bang” go live is possible in a 48-72 hour window. During the Big Bang, you can do the CRS and ASM install, 10gR2 upgrade, ASM Data migration, and RAC configuration in a single outage window.

If smaller windows are necessary, the incremental approach can be used to upgrade to 10gR2, install CRS and ASM, migrate to ASM, and then do RAC configuration all in separate outage windows.

If there is no downtime available, all the configuration steps can be done in a separate production worthy environment and the database created as a Physical Standby, which can be rolled forward until switchover, which would result in minimal downtime.

Helpful options

Instance Recovery nows plays the main roll in failover rather than database recover, so setting the following parameter is more needful:

`_fast_start_instance_recover_target`

TCP Timeout settings – set these lower to get a faster failover response from the server, and hence a faster failover response from CRS and Oracle RAC.

ADDITIONAL RECOMMENDATIONS

- Define SLA's for performance and availability for each service or application
- Use Grid Control to manage CRS, ASM, Database and Application.
- All changes to the production environment must be previously tested on a separate environment
- Apply changes to one system element at a time, first on test then on production.
- Keep a detailed change log
- Implement services where possible to manage workload for OLAP and customizations
- Configure OSWatcher to have handy information about the OS layer in case of need, see on Metalink Note:301137.1, OS Watcher User Guide
- Configure RDA to have handy information to Oracle Support in case of need, see on Metalink Note:314422.1, Remote Diagnostic Agent Getting Started. RDA 4.5+ includes RAC data collection capability. It can be used in place of RAC Diagnostics tool RACDDT.
- Establish support mechanisms and escalation procedures.
- Make sure DBA's have well tested procedures about how to deal with problems and collect required diagnostics.
- Use Racdiag.sql to check database during normal behavior and be able to compare results, see on Metalink Note:135714.1

SHORT GLOSSORY OF TERMS

ASM is .. this is not a specific requirement for RAC, but certainly provides a better storage mechanism for Oracle RAC than CFS

RBAL - Oracle background process. In an ASM instance coordinated rebalancing operations. In a DB instance, opens and mount diskgroups from the local ASM instance.

ARBx - Oracle background processes. In an ASM instance, a slave for rebalancing operations

PSPx - Oracle background processes. In an ASM instance, Process Spawners

GMON - Oracle background processes. In an ASM instance, diskgroup monitor.

ASMB - Oracle background process. In a DB instance, keeps a (bequeath) persistent DB connection to the local ASM instance. Provides heartbeat and ASM statistics.

During a diskgroup rebalancing operation ASM communicates to the DB AU changes via this connection.

O00x - Oracle background processes. Slaves used to connected from the DB to the ASM instance for 'short operations'.

Autoconfig – 11i EBS administrative tool for defining configuration options of the applications and database tiers

Adcfgclone – 11i EBS administrative tool for cloning applications (appsTier) and database (dbTier) tiers

VIP – Virtual IP defined and controlled by Oracle Clusterware for access connection from application to database

OLTP – On-Line Transaction Processing: end user and transaction interfaces for entering data

OLAP – On-Line Analytical Processing: Reporting and Analytical functions

SLA – Service Level Agreement: the agreed upon uptime requirements and down-time response windows for an enterprise application.