Best Practices for Optimizing Availability During Planned Maintenance Using Oracle Clusterware and Oracle Real Application Clusters

Oracle Maximum Availability Architecture White Paper
September 2007

Maximum Availability Architecture

Oracle Best Practices for High Availability

ORACLE

Best Practices for Optimizing Availability During Planned Maintenance Using Oracle Clusterware and Oracle Real Application Clusters

Best Practices for Optimizing Availability During Planned Maintenance Using Oracle Clusterware and Oracle Real Application Clusters

EXECUTIVE SUMMARY

Oracle Clusterware and Oracle Real Application Clusters (RAC) software in Oracle Database 10*g* Release 2 patch set 2 (10.2.0.3) and 11*g* Release 1 (11.1) provide technology that enables high availability during planned maintenance.

This paper covers typical planned maintenance activities in an Oracle Real Application Clusters environment. It details best practices in the areas of hardware maintenance, software maintenance, and Grid provisioning.

By implementing the best practices defined in this paper, you can reduce or eliminate downtime, reduce service level impact, and make planned maintenance activities more manageable. The following table lists categories of planned maintenance activities and the benefit of each:

Planned Maintenance Category	Benefit
Grid provisioning	**Zero database downtime**
Hardware, BIOS, firmware, operating system upgrades and patches	**Zero database downtime**
Oracle Clusterware upgrades and patches	**Zero database downtime**
Online and Rolling RDBMS/ASM patches	**Zero database downtime**
Non-rolling RDBMS/ASM patches	**40 to 60% reduction in database downtime**

PREREQUISITES AND TERMINOLOGY

This document assumes that you are familiar with Oracle Clusterware and Oracle RAC concepts and administration. The *Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* [5] book available with the Oracle Database 10*g* Release 2 documentation set is a good source for this information. You should also be familiar with the following Oracle Maximum Availability Architecture (MAA) white papers and high availability documentation that provide more details about the supporting technologies used in this paper:

- Client Failover Best Practices for Highly Available Oracle Databases: Oracle Database 10*g* Release 2 [6]
- Oracle Database High Availability Overview 10g Release 2 [2]
- SQL Apply Rolling Upgrade Best Practices: Oracle Database 10g Release 2 [7]

The following table defines the terminology used in this paper:

Term	Definition
Patches versus upgrades	The distinction between *patching* and *upgrades* can blur because the terms are often used interchangeably. This document defines the terms "upgrades" and "patches" as follows: • An **upgrade** moves a software component from one major release to another. For example, you may upgrade the Oracle database from release 10.1 to 10.2 or upgrade the operating system from RHEL3 to RHEL4. • A **patch** is any software change that happens between upgrades. For example, you may patch the Oracle Database from release 10.2.0.2 to 10.2.0.3 or patch the operating system from RHEL4 Update 3 to RHEL4 Update 4. Although the term *upgrade* is used loosely in some areas of this document[1], the only *true* upgrades this paper addresses are Oracle Clusterware upgrades and operating system upgrades.
Interim patch	An interim patch is any patch that is applied in between patchsets or major release upgrades. Interim patches are

[1] For example, the phrase "rolling upgrades" may be used when in fact the associated software change is a patch.

	applied with the OPatch utility. Examples of interim patches are simple bug fixes, merges of multiple bug fixes, bundles, and CPU patches.
`$ORACLE_HOME` or `ORACLE_HOME`	The environment variable that specifies the location of the Oracle Database software home. The location of the Oracle Database software.
`CRS home`	The location of the Oracle Clusterware software.
`ASM home`	The location of the ASM software
VIP	A virtual IP Address (VIP) is an alternate public address that client connections use to connect to a hostname through the Domain Name Service (DNS) instead of using the standard public IP address. If a node fails, then the node's VIP fails over to another node on which the VIP can accept connections. The VIP is one part of the Oracle solutions to the TCP timeout problem associated with node and network failures.
Services	Logical entities that represent database workloads. Services are assigned to instances that can perform the work. The use of services eliminates the need for clients to know on which Oracle instance work needs to run. Services are administered using Cluster Ready Services (CRS) on the server side.
EM Grid Control	Oracle Enterprise Manager Grid Control is Oracle's manageability framework.
FAN	Fast Application Notification (FAN) enables fast, out-of-band client notification upon cluster state changes.
DBCA	Database Configuration Assistant (DBCA). Oracle's client tool to administer databases, instances, and services.
OCR	Oracle Cluster Registry (OCR) is persistent data storage for Oracle Clusterware resources and configuration information.
OUI OPatch	Oracle Universal Installer (OUI) is a tool you can use to install Oracle major release and patch sets. Opatch is the Oracle tool used to apply interim patches to Oracle software. OPatch supports the following

operations:

- Applying an interim patch
- Rolling back the application of an interim patch
- Conflict resolution when applying an interim patch after previous interim patches have been applied
- Reporting on installed products and Interim (One-Off) patch

All OPatch commands have a –help option that will display the usage details for that command. The help syntax is:

```
  opatch [ -h[elp] { [ apply | lsinventory |
rollback | version ]}
```

Metalink Note 224346.1 explains how to find the latest version of Opatch.

SCOPE

This paper focuses on database-tier planned maintenance using Oracle Clusterware and Oracle RAC. If higher availability can be achieved using an alternate Oracle technology, it is noted in the paper. For more information on HA best practices, see the following documentation:

- *Oracle Database High Availability Overview 10g Release 2* [2]
- *Oracle Database High Availability Best Practices 10g Release 2* [3]

The planned maintenance activities covered in this paper fall into the following categories:

- Patches and Upgrades including:
  - **Oracle Interim patches:** A patch that is applied in between patchset releases with the Opatch utility. Interim patches are sometimes also referred to as patchset exceptions.
  - **Oracle Merge patches**: A type of interim patch that merges multiple fixes to ensure that one fix doesn't overwrite the other. Merge patches are typically required when a conflict is detected between an existing patch and a new patch.
  - **Oracle Bundle patches:** A type of interim patch that contains fixes to many important bugs, though not as many as a patchset.
  - **Oracle Critical Patch Update (CPU) patches:** A type of interim patch that contains fixes to critical security bugs delivered on a quarterly basis.
  - **Oracle Patchsets:** Software-release mechanisms for delivering tested and integrated product fixes on a regular basis.
  - **Oracle Upgrades:** Moving to a new major software release. Specifically, this paper addresses upgrades of Oracle Clusterware.
  - **Operating system upgrades and patches**: Software maintenance for the operating system.
- Grid Provisioning including add/remove cluster node[2], instances, listeners, services

The testing used for this project was performed using Oracle Database 10 Release 2 patch sets 1 and 2 (10.2.0.2 and 10.2.0.3), and various interim patches (the one exception to this is the online patch testing, which was done using Oracle Database 11*g* Release 1). The test cluster started out with two nodes, was expanded to four nodes and then reduced back down to two nodes. The patching and provisioning

[2] "Add/remove cluster node" is a common phrase used to describe the process of adding and removing Oracle Clusterware and RDBMS software

tests represent a common lifecycle encountered in the database tier of a Grid infrastructure.

AVAILABILITY

The following sections describe how overall availability is assessed in this paper, the components that impact availability during planned maintenance, and the technology that enables high availability.

Assessing Availability

Availability is always assessed from an application perspective. A database is not considered available unless the application is meeting its service level. Application availability falls into the following categories:

- *Application blackout*–Existing application connections cannot process work and new connections cannot be established.
- *Application service-level brownout*–The application is able to connect to the database and process some work, but it is not achieving its application service level
- *Full Application Availability*–The application is up and achieving its service level.

Hardware and Software Components

Planned maintenance can occur at many levels in the technology stack. Table 1 shows the components that require planned maintenance in a typical Oracle database tier environment. Each successive row in Table 1 depends on the row before it, illustrating which components are affected when dependent components are worked on.

Table 1 - Hardware and Software Components

Component	Example
Hardware	Replacement of a bad CPU or memory card.
BIOS/Firmware	Updating of the system BIOS or hard drive firmware
Operating System software	Operating system patch or upgrade
Oracle Clusterware software	Oracle Clusterware patch or upgrade; Oracle Clusterware software extension to a new node
Database Volume and File system Management software	ASM patch or upgrade; ASM software extension to a new node
Database software	Oracle Database software patch or upgrade; Oracle

	Database software extension to a new node
ASM instance	ASM instance extension to a new node
Database instance	Database instance extension to a new node
Database service	Database service extension to a new node

Oracle has different apply mechanisms, depending on the type of planned maintenance being performed, and the specific components being maintained. Table 2 and Table 3 describe how these apply mechanisms vary depending on the Oracle component:

- Table 2 describes patching components
- Table 3 describes provisioning components

Table 2 - Oracle Patching Components

Oracle Component	**Possible Apply Mechanisms**	**Example**
Oracle Clusterware interim patches	OPatch tool EM Grid Control	Simple bug fix Oracle Clusterware bundle
Oracle Clusterware patch set	OUI EM Grid Control	Patch 10.2.0.2 to 10.2.0.3
ASM interim patches	OPatch tool EM Grid Control	Simple bug fix Merge of multiple bug fixes
ASM patch set	OUI EM Grid Control	Upgrade 10.2.0.2 to 10.2.0.3
RDBMS interim patches	OPatch tool EM Grid Control	Simple bug fix Merge of multiple bug fixes CPU
RDBMS patch set	OUI with subsequent DBCA or scripts	Upgrade 10.2.0.2 to 10.2.0.3

Table 3 - Oracle Provisioning Components

Oracle Component	**Possible Provisioning Mechanisms**	**Examples**

Oracle Clusterware add node	OUI Clone EM Grid Control	Add a node to an exiting cluster
Oracle Clusterware remove node	Manual steps and OUI EM Grid Control	Remove a node from an existing cluster
RDBMS add node	OUI Clone EM Grid Control	Add RDBMS software to a new node in a cluster
RDBMS remove node	Manual steps and OUI EM Grid Control	Remove RDBMS software from an existing node in a cluster
ASM add node	OUI Clone EM Grid Control	Add ASM software to a new node in a cluster
ASM remove node	Manual steps and OUI EM Grid Control	Remove ASM software from an existing node in a cluster
RDBMS add instance	Manual steps DBCA EM Grid Control	Add an RDBMS instance to a node in the cluster
RDBMS remove instance	Manual steps DBCA EM Grid Control	Remove an RDBMS instance from a node in the cluster
ASM add instance	Manual steps DBCA EM Grid Control	Add an ASM instance to a node in the cluster
ASM remove instance	Manual steps DBCA in silent mode EM Grid Control	Remove an ASM instance from a node in the cluster

Listener add	NETCA EM Grid Control	Add a listener to a node in the cluster
Listener removal	NETCA EM Grid Control	Remove a listener from a node in the cluster
Service add	Manual steps DBCA EM Grid Control	Add a service to an instance in the cluster
Service removal	Manual steps DBCA EM Grid Control	Remove a service from an instance in the cluster

Downtime Avoidance

Oracle provides technology that enables *full application availability* for many planned maintenance activities. The following sections describe the technology in more detail.

Provisioning

Oracle Clusterware and Oracle RAC provide the infrastructure that permits Grid provisioning to be performed with *full application availability*. The technology components that enable this capability are as follows:

- The OUI provides software provisioning.
- The Oracle Clusterware Cluster Synchronization Services (CSS) component provides cluster member join and leave operations.
- ASM provides storage cluster member join and leave operations.
- Oracle RAC provides database cluster member join and leave operations.
- Oracle Net permits listeners to be added to, or removed from, new cluster nodes.
- The Oracle Clusterware Cluster Ready Services (CRS) component permits services to be extended to, and retracted from, new cluster nodes.
- FAN and services permit applications to be aware of cluster state changes.

The following list provides examples of Grid provisioning planned maintenance activities that leverage this technology with zero downtime:

- Add/remove cluster node
- Add/remove RDBMS software to/from cluster node

- Add/remove ASM software to/from cluster node
- Add/remove ASM instances to/from storage cluster nodes
- Add/remove RDBMS instances to/from cluster nodes
- Add/remove listeners to/from cluster nodes
- Add/remove services to/from cluster instances

Traditional Patching

Oracle Clusterware and Oracle RAC provide the infrastructure that enables *rolling patches* for qualified interim patches. In most cases, rolling patches are applied iteratively to each node in the cluster so that all but one node in the cluster are available during the patch application process[3]. To qualify as a rolling patch, a patch must be able to coexist with the unpatched code. Oracle determines rolling patch eligibility when the patch is created and that eligibility is included with the patch itself. Patches that are not rolling upgradeable are those that affect shared components such as Interprocess Communication (IPC) or the data dictionary. To determine if a patch is a rolling patch:

- On Linux and UNIX, issue the following command:

```
$opatch query -all <path to patch directory> | grep rolling

Patch is a rolling patch: true
```

- On Windows, issue the following command:

```
C:\> opatch query -all <path to patch directory> | findstr rolling

Patch is a rolling patch: true
```

Online Patching

Note: While the rest of this document is pertinent to Oracle Database 10g Release 2, the sections describing online patching are only relevant to Oracle Database 11g Release 1. Online patching is not supported in Oracle Database 10g Release 2.

As of Oracle Database 11*g* Release 1, Oracle supports *online patching* for some qualified interim patches. Online patching provides the ability to patch the processes in an Oracle instance without bringing the instance down. Each process associated with the instance checks for patched code at a safe execution point, and then copies the new code into its process space. This means that the processes being patched may not necessarily pick up the new code at the exact same time.

A key difference between traditional patching and online patching is that traditional patching is implemented at the software level and online patching is implemented at the software/Oracle instance level. In other words, instances using an `ORACLE_HOME` that receives a traditional patch always use the patched code

[3] It is not a technical requirement that only one node be brought down and patched at a time but it is a best practice because it leaves the most computing resources available during the planned maintenance activity.

whereas instances using an ORACLE_HOME that receives an online patch only receive the patched code if the instance is specified when the patch is applied.

Online patching is most useful for debug patches and interim patches where the scope of the fix is small. The same restrictions that apply to rolling patches apply to online patches, and some further restrictions[4]. Use the following command to determine if a patch is an online patch:

```
$ opatch query <path to patch directory> –is_online_patch
```

Look for the text: "Patch is a online patch: true" in the output to verify that the patch can be applied online.

Applying an online patch increases memory consumption on the system because each Oracle process uses more memory from the Program Global Area (PGA). You need to take your memory requirements into consideration before you begin applying an online patch[5]. Each online patch is unique and the memory requirements are patch specific. As is always the case, the best practice is to apply the patch on your test system first. Doing so also enables you to assess the effect of the online patch on your production system and estimate any additional memory usage. For a typical online debug patch, you can expect a 3-5 percent increase in memory consumption for every process associated with the instance being patched. The memory is released when **both** the patch is rolled back **and** the process that received the patch has exited.

[4] Online patches can be created when the patch does not change shared memory structures in the System Global Area (SGA), or other critical internal code structures.
[5] For example, on Linux, you could use the pmap command or the /proc file system to assess the memory impact in your test environment.

BEST PRACTICES

Planned maintenance best practices are separated into three sections:

- Configuration Best Practices
- Operational Best Practices Common During Planned Maintenance
- Operational Best Practices for Specific Planned Maintenance

Configuration Best Practices

This section details common configuration best practices that can be applied to planned maintenance activities. The sections that follow put these configuration best practices to use in an operational context.

Capacity Planning

Proper capacity planning is a critical success factor for all aspects of Oracle clustering technology, but it is of particular importance for planned maintenance. Customers must ensure that the work a cluster is responsible for can be done when a small part of the cluster (ex: a node) is unavailable. If the cluster cannot keep up after a planned or unplanned outage, the potential for *cascading problems* is higher due to system resource starvation.

When sizing your cluster, ensure that x percentage of the cluster can meet your service levels where x percentage represents the amount of computing resource left over after a typical planned or unplanned outage. For example, if you have a four-node cluster and you want to apply patches in a rolling fashion, meaning one node will be taken down at a time, then three instances should be able to run the work requested by the application.

One other aspect to capacity planning that is important during planned maintenance is ensuring that any work being done as part of the planned maintenance is separated from the application work when possible. For example, if a patch requires a SQL script to be run after all nodes have been patched[6], it is a best-practice to run this script on the last node receiving the patch before allowing the application to start using that node. This technique ensures that the SQL script has full use of the operating system resources on the node and it is less likely that it will affect the application.

Use Local HOME for ASM, RDBMS, and Oracle Clusterware

All rolling patch features require that the software home being patched is local, not shared. This means the software is physically present in a local file system on each node in the cluster and ***it is not*** on a shared cluster file system. The reason for this

[6] An example of this is that `CATCPU.SQL` script that must be run after installing the CPU patch on all nodes

requirement is that if a shared cluster file system is used, patching the software on one node affects all of the nodes[7]. Using a local file system allows software to be patched on one node without affecting the software on any other nodes.

Out of Place Patch Set Installation with Cloning

Traditionally, Oracle RDBMS patch sets have been done *in-place,* which means that the new code was applied directly over the old code. There were a variety of reasons for applying patch sets in-place such as less space consumption and a simpler install. However, many of these reasons are no longer valid in today's IT environment. The downside to an in-place RDBMS patch set upgrade is that the application cannot connect to the database while new code is being copied in[8]. To avoid this availability impact, use a combination of Oracle cloning technology and an *out-of-place* patch set installation. Cloning technology allows the existing software to be copied to a new `ORACLE_HOME` after which a patch set may be applied.

An *out-of-place* patch set installation with cloning has the following advantages:

- Applications remain available while software is upgraded in the new `ORACLE_HOME`
- The configuration inside the `ORACLE_HOME`[9] is retained because the cloning procedure involves physically copying the software.

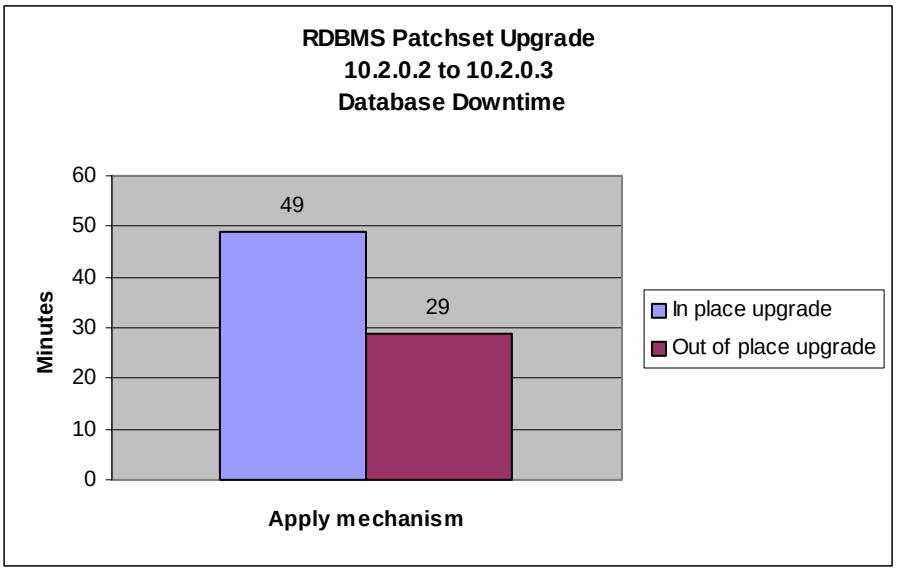
The one disadvantage to an out-of-place patch set installation with cloning is that you must change any `$ORACLE_HOME` environment variable hard coded in application code and Oracle specific scripts. See Appendix B for a list of Oracle-specific manual reconfiguration steps you must run to rectify the `$ORACLE_HOME` environment variable after cloning

If application availability is more important to you than changing customizations, consider performing an out-of-place patch set installation with cloning. The following chart shows increase in availability when upgrading from release 10.2.0.2 to 10.2.0.3 using out-of-place upgrades with cloning versus a traditional in-place upgrade.

> **Note:** Oracle offers other solutions, such as SQL Apply Rolling Upgrade and Streams, to reduce downtime to seconds during upgrades. This example illustrates the benefit that can be obtained without the extra steps, and potential limitations, associated with using these features[10].

---

[7] This would require all components using the software to be shut down on all nodes.
[8] A typical Oracle RDBMS patch set installation can take at least 20 minutes
[9] Examples of this are files such as listener.ora, tnsnames,.ora, initSID.ora, etc.
[10] For example, SQL Apply Rolling Upgrade and Streams have datatype restrictions that may prevent their use.



Client Configuration and Migration

The ability to migrate client connections to and from nodes on which you are working is a critical aspect of planned maintenance. Migrating client connections should always be the first step in any planned maintenance activity requiring software shutdown (for example, when performing a rolling upgrade). The potential for problems increases if there are still active database connections when the software shutdown commences.

Oracle provides *services*, *FAN, FAN-integrated clients, client side load balancing, fast connection failover, and run time connection load balancing* to achieve this objective. Detailed information about these features can be found in the MAA white paper: Client Failover Best Practices for Highly Available Oracle Databases: Oracle Database 10*g* Release 2 [6].

An example of a best-practice process for client redirection during planned maintenance is as follows (**Note:** the following example is specific to FAN ONS[11]):

1. FAN-ONS-integrated clients properly configured with runtime connection load balancing and fast connection failover.
2. Oracle Clusterware stops services on the instance to be brought down or relocates services to an alternate instance.
3. Oracle Clusterware returns a Service-Member-Down event.
4. FAN-ONS-integrated client receives the event and moves connections to other instances offering the service.

[11] FAN OCI does not respond to service events in release 10.2.0.3. In this case, you can use the `DBMS_SERVICE` package to remove sessions from instances that are being worked on.

Separate Home Directory Locations for ASM and RDBMS

While it is technically feasible to run ASM and RDBMS instances out of the same `ORACLE_HOME`, it is not a preferred configuration. You should create an ORACLE_HOME for the RDBMS and an ASM home for ASM instances to enable more flexibility during patches and upgrades. For example, you want to avoid having to stop your volume manager (ASM) to apply a patch that fixes code exclusively used by the RDBMS. Doing so would require that you shut down all of the databases, including the ones that are not using the patched code.

Run Listener Out of Most Recent Oracle Home or ASM Home

If you have many Oracle homes, then managing the listener or listeners in use can be a confusing and error-prone task. You should run the listener with the latest version when multiple versions are available[12]. If you typically update your ASM software before your RDBMS software, then running the listener out of the ASM home simplifies the manageability of your network configuration and proactively avoids potential bugs in older listener code.

Highly Available Services

For cases where a service only has one preferred instance, ensure that the service is started immediately on an available instance after it is brought down on its preferred instance. This ensures that affected clients can instantaneously reconnect and continue work. Oracle Clusterware handles this responsibility and it is of utmost importance during unplanned outages.

Even though you can rely on Oracle Clusterware for this during planned maintenance as well, it is safer to ensure that the service is available on an alternate instance by manually starting an alternate preferred instance ahead of time. This eliminates the single point of failure with a single preferred instance and you have the luxury to do this because it is a planned activity. To accomplish this, add at least a second preferred instance to the service definition and start the service before the planned maintenance. You can then stop the service on the instance where maintenance is being performed with the assurance that another service member is already available. Adding one or more preferred instances does not have to be a permanent change. You can revert it back to the original service definition after performing the planned maintenance.

Manually relocating a service rather than changing the service profile is advantageous in cases such as the following:

- If Oracle XA is being used, then use manual service relocation because running a service on more than one instance is not supported.

[12] The Oracle Clusterware home, referred to in this document as the CRS home, does not qualify for this because Oracle Clusterware cannot run the listener in Oracle 10*g* Release 2

- If an application is not designed to work properly with multiple service members, then application errors or performance issues can arise.

As with all configuration changes, you should test the effect of a service with multiple members to assess its viability and impact in a test environment before implementing the change in your production environment.

Operational Best Practices Common During Planned Maintenance

The following sections describe common operational best practices. They are referenced in the context of specific planned maintenance activities in the next section.

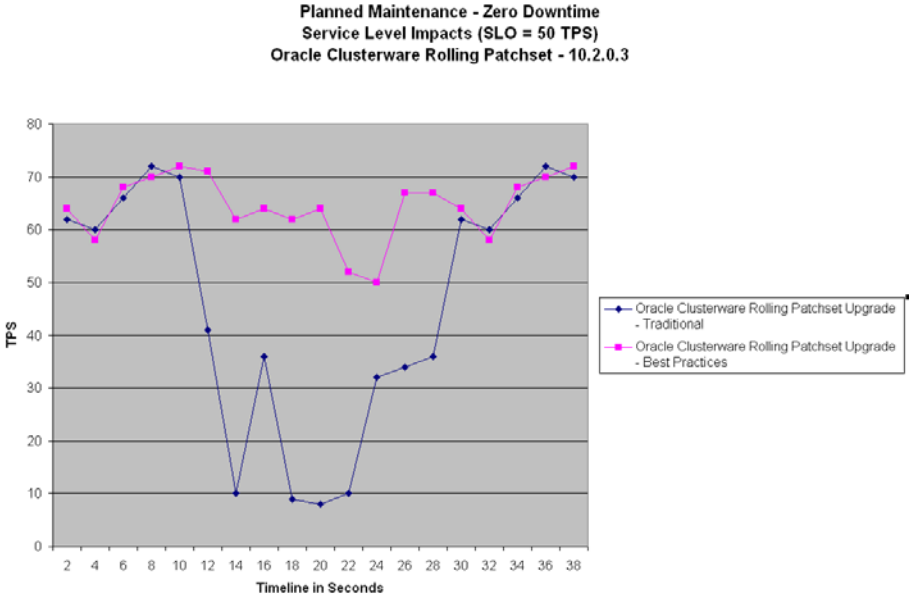
Planned Maintenance Executed in a Rolling Manner

When possible, perform planned maintenance in a rolling manner. This means performing planned maintenance on one node at a time, thereby leaving n-1 nodes available to the application. Detailed examples of this can be found in the Operational Best Practices for Specific Planned Maintenance section.

Explicit Shutdown Immediate During Rolling Patches

Rolling patches require you to shut down the RDBMS instance on the current node being worked on. The type of shutdown differs depending on the technique used. For example, in release 10.2.0.3, a shutdown of the Oracle Clusterware stack forces a `SHUTDOWN ABORT` whereas the default action for SRVCTL initiated shutdown commands is a `SHUTDOWN IMMEDIATE`.

Performing a `SHUTDOWN IMMEDIATE` is the best practice. The `SHUTDOWN IMMEDIATE` action distributes the Global Resource Directory (GRD) resources more gracefully to remaining nodes and does not require a surviving instance to perform instance recovery. If a `SHUTDOWN ABORT` is used, a surviving instance must perform instance recovery, which can temporarily restrict access to GRD resources and cause a brief *application service level brownout*. Here is an example showing service level impact with (best practice approach) versus `SHUTDOWN ABORT`(traditional approach) during a CRS patch set installation:



If you must use a `SHUTDOWN ABORT`, then be sure to follow the instance recovery best practices defined in the MAA white paper, Oracle Database 10*g* Release 2 Best Practices: Optimizing Availability During Unplanned Outages Using Oracle Clusterware and RAC [8].

Minimize_Downtime Option of OPatch for Database Patches

In the event that a database interim patch does not qualify for rolling installation, OPatch offers a `MINIMIZE_DOWNTIME` option that can be used to achieve higher availability than the traditional patching mechanism. As is the case with rolling patches, this feature requires that software be installed locally on each node. This feature does not apply to Oracle Clusterware interim patches because they are always rolling upgradeable.

Traditionally, non-rolling database interim patches have been applied using the following process:

1. Shut down the affected software on all nodes.
2. Apply the interim patch to the software on all nodes.
3. Start the patched software.

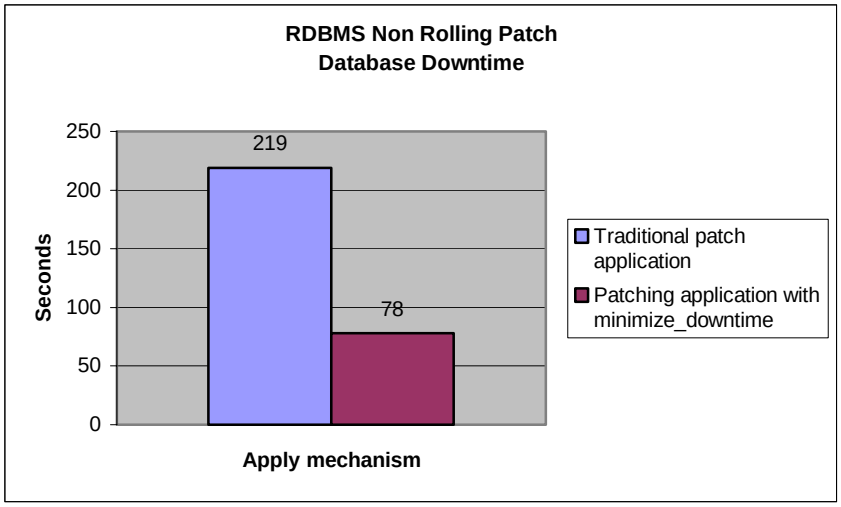
With minimize_downtime, the following steps are used by the OPatch utility:

1. Shut down a subset of instances in the cluster (subset 1)
2. Apply the patch to instances that are shutdown (subset 1)
3. Shut down the unpatched cluster instances (subset 2)
4. Start the patched cluster instances (subset 1)
5. Apply the patch on the second set of instances (subset 2)
6. Start the second set of patched cluster instances (subset 2)

This process ensures that patched and unpatched code are not active at the same time, and the only downtime is the time it takes to shut down one set of instances and start the other (steps 3 and 4)..

Starting with OPatch version 10.2, the `MINIMIZE_DOWNTIME` option can be used for rollback. In prior versions, this was not supported. If you are using a pre-10.2 version of Opatch, and would like to minimize downtime during rollback, a similar technique can be done manually by using the `-LOCAL` option of OPatch on the nodes in each subset.

Here is an example showing the availability increase by using `MINIMIZE_DOWNTIME`:



Always Relink When Patching/Upgrading the Operating System in a Rolling Manner

It is not always clear if a relink of Oracle software is required when performing an operating system patch or upgrade. Because the application is available during rolling operating system patches and upgrades, you should always relink just in case it is needed. Using this best practice removes guesswork and reduces the potential for error.

For the RDBMS and ASM, use the following command to relink:

```
$ORACLE_HOME/bin/relink all
```

For Oracle Clusterware, the only code that can be relinked is the client-shared libraries because the Oracle Clusterware code itself cannot be relinked. To relink the client-shared libraries, issue the following command:

```
<CRS_HOME>/bin/genclntsh
```

SRVCTL Commands from Latest Release During patch Set Installation

When performing patch set upgrades, SRVCTL commands will be used to manage the Oracle RAC resources through the patch set installation process. If available, the SRVCTL command from the latest patch set release for the particular database version should be used. This will avoid any potential bugs and ensure the best functionality is available for the upgrade. For example, if performing an upgrade from 10.2.0.2 to 10.2.0.3, then use the 10.2.0.3 SRVCTL commands during the upgrade process.

Note that SRVCTL from the latest patch set release will always be available when performing out-of-place patch set upgrades.

Guaranteed Restore Points

For any planned maintenance that changes the database, a guaranteed restore point should be created before the maintenance. If fallback is required, then the database

can be flashed back to the guaranteed restore point in addition to the restoration of the related software. The amount of time it takes to flash back to the restore points correlates to the amount of changes since the restore point. Generally, with planned maintenance, this will be a significantly lower amount of time compared to a database restore.

Note that guaranteed restore points can be used even if flashback database is not enabled however the very first guaranteed restore point must be created when the database is in mount mode.

Use Proper Order of Instances at Service Creation Time

When creating services, the order that the instances are specified in is the order that will be used when picking a failover target for the service. If you prefer that a service failover to particular instances, then ensure the instance order reflects this at service creation time. If you have multiple services running on a given instance, consider creating those services with different ordering of available instances so that they will not all failover to the same target. This will spread out the work during both unplanned outages and planned maintenance.

Add ASM Instance, Listener, and RDBMS Instance Together

After adding Oracle Clusterware and RDBMS software to a new node, you must add the ASM instance, listener, and RDBMS. DBCA can add these three resources in a single step and ensure that the resource configuration is properly stored in the OCR. DBCA does this implicitly when invoking the 'Add Instance' operation. Using this automation is a best practice to ease manageability.

Migrate Work Off Node Being Maintained

Before starting the shutdown process on a cluster node/instance, ensure that the application has safely moved its work off of that node/instance to other nodes/instances in the cluster. With no application work running, the shutdown is more reliable and timely because it has less to do.

The preferred method for moving client connections off of a node/instance is to use the recommendations described in the Client Configuration and Migration section.

In the event that all work cannot be moved from the node/instance being maintained, at a minimum, long running transactions should be moved or terminated. This ensures that the overhead of transaction recovery does not interfere with the planned maintenance operation.

Use Disable/Enable When Stopping/Starting Resources During Planned Maintenance

When stopping Oracle Clusterware resources during planned maintenance, those resources should also be disabled after they are stopped. This ensures that, if an

unplanned outage occurs during the maintenance, Oracle Clusterware will not try to start the resources. Once the planned maintenance is complete, the resources should be enabled and started.

Similarly, Oracle Clusterware itself should be disabled when the planned maintenance requires that it be down throughput the entire operation.

Patching Fallback

It is imperative to have a fallback plan when performing planned maintenance in case something goes wrong. When planning for fallback, it is important that both software and data storage restoration is considered when the patch affects both. For example, an RDBMS patch set includes software changes as well as changes to the database.

The best practice for fallback during RDBMS patching is as follows:

- For interim patches, use OPatch's rollback feature to restore the old software. If the patch also includes a SQL script, use a guaranteed restore point to restore the database to before the SQL script was executed.
- For patch sets, use out-of-place patch sets so that the old software still exists and can be used by modifying the $ORACLE_HOME in the OCR. To restore the database, use a guaranteed restore point to restore the database to before the post patch set SQL script was run.

The best practice for fallback during Oracle Clusterware patching is as follows:

- For interim patches, use OPatch's rollback feature to restore the old software. If the patch also includes an OCR change, see Appendix C for guidelines.
- Reinstall Oracle Clusterware and logically recreate your application specific OCR entries. Examples of logically recreating OCR entries with SRVCTL can be found in the *Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* [5]. Another option for logically recreating OCR entries is to take an export of the OCR before the planned maintenance using the ocrconfig utility. This export must be taken during a time when the OCR is quiesced to ensure the export is consistent. The exported entries can be placed back in the OCR using the import option of ocrconfig.

Apply Online Patches With the –LOCAL Option

Note: *While the rest of this document is pertinent to Oracle 10g Release 2, the sections describing online patching are only relevant to Oracle 11g Release 1. Online patching is not supported in Oracle 10g Release 2.*

When applying online patches, use the `-LOCAL` option of OPatch to apply the patch to each node individually. This provides the following benefits:

- After the first node is patched, you can assess the effectiveness of the online patch before rolling it out to other nodes. This is an inherently safer way to apply a patch.
- It simplifies patch installation by enabling operating system authentication to be used during installation. Authentication is required for online patches because the patch application will log into the instance to install and enable the patch. You can use operating system authentication because the patch is applied locally. In addition, it simplifies the instance parameter configuration required at install time.
- It spreads out the impact of the online patch so as not to impact the application service level. In other words, rather than having all Oracle processes accept new code across the cluster at the same time, which can cause a brief *application service level brownout*, it spreads the work over a period of time.

Always install and remove online patches with the same instance configuration

When applying an online patch, the OPatch utility invokes internal commands that enable the patch. Those commands are specific to a particular instance, which is why instance parameters must be passed in when applying the patch. To ensure that the patched instances are tracked properly, you should always use the exact same instance parameters when applying and rolling back a patch. For example, if you have an `ORACLE_HOME` that has two instances using it and you only need the online patch applied on one of the instances, then specify only that instance when applying and rolling back the patch. Issue the following command to determine the instances that have the patch applied:

```
OPatch lsinv –detail
```

Operational Best Practices For Specific Planned Maintenance

This section contains operational best practices that make use of the best practices described in the previous sections. Operational best practices are organized by process so components for which the process is the same are grouped together. Processes are explained at a high level with some sample code excerpts where appropriate[13]. The best practice fallback for each process is also included.

Note the processes below are defined at a high level to give guidance for a particular planned maintenance activity. They are not a substitute for associated documentation such as the patch README file.

Important points to consider:

- **It is always recommended to do the following:**
  - **Test any planned maintenance activity on a test system that mimics your production system**
  - **After performing the planned maintenance, test it with a real world production load to ensure you can maintain your performance and HA service levels**
  - **Test your fallback plans and contingency plans**
  - **Create a backup prior to maintenance**
- **All processes in this section assume that the previously defined best practices are followed. Specifically, the best practice of using local software homes is imperative to the success of the rolling patch procedures outlined below. Do not attempt to use the rolling patch procedures below if you are using shared software homes.**

Hardware, Firmware, BIOS Patching, Operating System Patching

Examples of operations in this category are as follows:

- BIOS update from hardware vendor
- Firmware patch for internal hard drive from hardware vendor
- RHEL4 Update 4 patched to RHEL4 Update 5

The best practice process for this category is as follows:

1. Ensure that the services used by the application are available on more than one instance and that the application has been tested with multiple service members.
2. Stop and disable the services for instances running on the current node being worked on. This generates a FAN event, which causes the

[13] Oracle USA Inc. does not support any of the sample code excerpts in this white paper

application to stop sending work to the instances. For example:

```
srvctl stop service -d gen -s oltp -i gen1
srvctl disable service -d gen -s oltp -i gen1
```

3. Use SRVCTL to perform a `SHUTDOWN IMMEDIATE` on the instances on the current node being worked on. For example:
   ```
   srvctl stop instance -d gen -i gen1 -o immediate
   ```
4. Shut down the Oracle Clusterware stack as owner root on the current node being worked on. For example:
   ```
   <CRS_HOME>/bin/crsctl stop crs
   ```
5. Disable the Oracle Clusterware stack as owner root on the current node being worked on. For example:
   ```
   <CRS_HOME>/bin/crsctl disable crs
   ```
6. Perform planned maintenance including any required backups.
7. If the maintenance is an operating system patch, relink the Oracle RDBMS code and Oracle Clusterware client code.
8. Enable the Oracle Clusterware stack as owner `root` on the current node being worked on. For example:
   ```
   <CRS_HOME>/bin/crsctl enable crs
   ```
9. Startup the Oracle Clusterware stack as owner root on the current node being worked on. Note this will implicitly start the instances that were shutdown. For example:
   ```
   <CRS_HOME>/bin/crsctl start crs
   ```
10. Assess the health of the system with a subset of the application workload to ensure the changes have not caused a problem.
11. Enable/start the services on the instances that were shutdown. This will generate a FAN event that will, along with Run Time Connection Load Balancing, cause the application to start sending work to the instances again. For example:
    ```
    srvctl enable service -d gen -s oltp -i gen1
    srvctl start service -d gen -s oltp -i gen1
    ```
12. Repeat on next node until all nodes are patched.
13. If the service configuration was changed, restore it to its initial state.

If something goes wrong during the patching, the fallback plan is as follows:

- Consult with the vendor providing the patch for fallback best practices

Operating System Upgrades

- Upgrading a Redhat operating system from RHEL3 to RHEL4 is an example of an operating system upgrade.

The best practice process for this category is as follows:

1. Ensure that the services used by the application are available on more than one instance and that the application has been tested with multiple service members.
2. Stop and disable the services for instances running on the current node being worked on. This will generate a FAN event, which will cause the application to stop sending work to the instances. For example:
   ```
   srvctl stop service -d gen -s oltp -i gen1
   srvctl disable service -d gen -s oltp -i gen1
   ```
3. Use SRVCTL to `SHUTDOWN IMMEDIATE` the instances on the current node being worked on. For example:
   ```
   srvctl stop instance -d gen -i gen1 -o immediate
   ```
4. Shut down the Oracle Clusterware stack as owner `root` on the current node being worked on. For example:
   ```
   <CRS_HOME>/bin/crsctl stop crs
   ```
5. Disable the Oracle Clusterware stack as owner root on the current node being worked on. For example:
   ```
   <CRS_HOME>/bin/crsctl disable crs
   ```
6. Perform the operating system upgrade including any required backups. ***Note: Once the upgrade is complete, all Oracle software should exist in the same location where it was before the upgrade.***
7. Relink the Oracle RDBMS code and Oracle Clusterware client code.
8. Enable the Oracle Clusterware stack as owner root on the current node being worked on. For example:
   ```
   <CRS_HOME>/bin/crsctl enable crs
   ```
9. Startup the Oracle Clusterware stack as owner `root` on the current node being worked on. **Note:** this implicitly starts the instances that were shutdown. For example:
   ```
   <CRS_HOME>/bin/crsctl start crs
   ```
10. Assess the health of the system with a subset of the application workload to ensure the changes have not caused a problem.
11. Enable/start the services on the instances that were shutdown. This will generate a FAN event that will, along with Run Time Connection Load Balancing, cause the application to start sending work to the instances again. For example:
    ```
    srvctl enable service -d gen -s oltp -i gen1
    srvctl start service -d gen -s oltp -i gen1
    ```
12. Repeat on next node until all nodes are patched.
13. If the service configuration was changed, restore it to its initial state.

- If something goes wrong during the operating system upgrade, the fallback plan is to consult with the operating system vendor for fallback best practices.

Oracle Clusterware Interim Patches

Examples of planned maintenance that fall into this category include:

- Oracle Clusterware Bundle Patch 2 for 10.2.0.2
- Oracle Clusterware merge interim patch
- Oracle Clusterware one-off interim patch

Oracle Clusterware Bundle patches are the preferred method for obtaining Oracle Clusterware fixes.

All Oracle Clusterware patches can be applied in a rolling fashion. Is it very rare for an Oracle Clusterware interim patch to change the OCR so this process assumes that and does not include backup/restoration of the OCR. In the unlikely event that the OCR is changed, see Appendix C for guidelines.

The best practice process for this category is as follows:

1. Ensure that the services used by the application are available on more than one instance and that the application has been tested with multiple service members.
2. Stop and disable the services on instances running on the current node being worked on. This will generate a FAN event, which will cause the application to stop sending work to the instances. For example:
   ```
   srvctl stop service -d gen -s oltp -i gen1
   srvctl disable service -d gen -s oltp -i gen1
   ```
3. Use SRVCTL to `SHUTDOWN IMMEDIATE` the instances on the current node being worked on. For example:
   ```
   srvctl stop instance -d gen -i gen1 -o immediate
   ```
4. Shut down the Oracle Clusterware stack as user `root` on the current node being worked on. For example:
   ```
   <CRS_HOME>/bin/crsctl stop crs
   ```
5. Apply the Oracle Clusterware interim patch in rolling mode with the OPatch utility. Instructions to perform this are detailed in the patch README file. The OPatch utility implicitly backups up the old version of the software so it is not required to do this as a separate step
6. If the Oracle Clusterware stack is not implicitly started by the patch procedures, startup the Oracle Clusterware stack on the current node being worked on. Note this will implicitly start the resources that were shutdown. For example:
   ```
   <CRS_HOME>/bin/crsctl start crs
   ```

7. Assess the health of the system with a subset of the application workload to ensure the changes have not caused a problem.
8. Enable/start the services on the instances that were shutdown. This will generate a FAN event that will, along with Run Time Connection Load Balancing, cause the application to start sending work to the instances again. For example:
   ```
   srvctl enable service -d gen -s oltp -i gen1
   srvctl start service -d gen -s oltp -i gen1
   ```
9. Repeat on next node until all nodes are patched.
10. If the service configuration was changed, restore it to its initial state.

If something goes wrong during the patch application, the fallback plan is as follows:

1. OPatch cannot rollback patches in a rolling fashion the same way that it applies patches in a rolling fashion. The best practice therefore, assuming the patch is not causing a serious problem that needs to be eliminated immediately, is to rollback the patch on each node individually using the `-LOCAL` option.

Oracle Clusterware Patch Sets

- Oracle Clusterware Patch Set 10.2.0.3 is an example of planned maintenance.

All Oracle Clusterware patches can be applied in a rolling fashion. The best practice process for these planned maintenance activities is as follows:

1. Ensure that the services used by the application are available on more than one instance and that the application has been tested with multiple service members.
2. Stop and disable the services on instances running on the current node being worked on. This will generate a FAN event, which will cause the application to stop sending work to the instances. For example:
   ```
   srvctl stop service -d gen -s oltp -i gen1
   srvctl disable service -d gen -s oltp -i gen1
   ```
3. Apply the Oracle Clusterware patch set in rolling mode with the OUI. The patch set code will be staged into a temporary location inside of *<CRS_HOME>* and activated when the post install root scripts are run in a later step. In other words, do not execute the root scripts yet even if the OUI prompts you to. These scripts will be run in steps 5 and 6.
4. Use SRVCTL to `SHUTDOWN IMMEDIATE` the instances on the current node being worked on. For example:
   ```
   srvctl stop instance -d gen -i gen1 -o immediate
   ```

5. Shut down the CRS stack on the current node being worked on. For example:
   ```
   <CRS_HOME>/bin/crsctl stop crs
   ```
6. Execute the post-install root scripts as instructed by the OUI. This will implicitly start the Oracle Clusterware stack. For example:
   ```
   <CRS_HOME>/install/root102.sh
   ```
7. Assess the health of the system with a subset of the application workload to ensure the changes have not caused a problem.
8. Enable/start the services on the instances that were shutdown. This will generate a FAN event that will, along with Run Time Connection Load Balancing, cause the application to start sending work to the instances again. For example:
   ```
   srvctl enable service -d gen -s oltp -i gen1
   srvctl start service -d gen -s oltp -i gen1
   ```
9. Repeat steps 1 and 2 and 4 through 8 on the next node until all nodes are patched.
10. If the service configuration was changed, restore it to its initial state.

If something goes wrong during the patch set installation, the fallback plan is as follows:

- Reinstall Oracle Clusterware and logically recreate your application specific OCR entries. Examples of logically recreating OCR entries with SRVCTL can be found in the *Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* [5]. Another option for logically re-creating OCR entries is to take an export of the OCR before the planned maintenance using the OCRCONFIG utility. This export must be taken during a time when the OCR is quiesced to ensure the export is consistent. The exported entries can be placed back in the OCR using the import option of the OCRCONFIG command.

Oracle Clusterware Upgrades

- Upgrading Oracle Clusterware from 10gR1 to 10gR2 is an example of planned maintenance.

All Oracle Clusterware upgrades can be done in a rolling fashion. The best practice process for these planned maintenance activities is as follows:

1. Ensure that the services used by the application are available on more than one instance and that the application has been tested with multiple service members.
2. Stop and disable the services on instances running on the current node being worked on. This will generate a FAN event, which will cause the application to stop sending work to the instances. For example:

```
srvctl stop service -d gen -s oltp -i gen1
srvctl disable service -d gen -s oltp -i gen1
```

3. Unlock the <*CRS_HOME*> by executing the pre-update script located on the media of the new version: (*mountpoint/clusterware/upgrade/preupdate.sh)*
4. Apply the Oracle Clusterware upgrade in rolling upgrade fashion with the OUI. The upgrade code will be staged into a temporary location inside of <*CRS_HOME*> and activated when the post install root scripts are run in a later step. In other words, do not execute the root scripts yet even if the OUI prompts you to. These scripts will be run in steps 6 and 7.
5. Use SRVCTL to `SHUTDOWN IMMEDIATE` the instances on the current node being worked on. For example:
   ```
   srvctl stop instance -d gen -i gen1 -o immediate
   ```
6. Shut down the Oracle Clusterware stack on the current node being worked on. For example:
   ```
   <CRS_Home>/bin/crsctl stop crs
   ```
7. Execute the post-install root scripts as instructed by the OUI. For example:
   ```
   <CRS_HOME>/install/rootupgrade.sh
   ```
   This implicitly starts the Oracle Clusterware stack.
8. Assess the health of the system with a subset of the application workload to ensure the changes have not caused a problem.
9. Enable/start the services on the instances that were shutdown. This generates a FAN event that will, along with Run Time Connection Load Balancing, cause the application to start sending work to the instances again. For example:
   ```
   srvctl enable service -d gen -s oltp -i gen1
   srvctl start service -d gen -s oltp -i gen1
   ```
10. Repeat steps 1-3 and 5-8 on next node and continue until all nodes are patched.
11. If the service configuration was changed, restore it to its initial state.

If something goes wrong during the upgrade, the fallback plan is as follows:

- Reinstall Oracle Clusterware and logically recreate your application specific OCR entries. Examples of logically recreating OCR entries with SRVCTL can be found in the *Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* [5]. Another option for logically recreating OCR entries is to take an export of the OCR before the planned maintenance using the ocrconfig utility. This export must be taken during a time when the OCR is quiesced to ensure the export is consistent. The exported entries can be placed back in the OCR using the import option of ocrconfig.

RDBMS and ASM Rolling Interim Patches

- An Oracle one-off rolling patch is an example of this type of planned maintenance.

Most Oracle RDBMS and ASM rolling interim patches do not change the database so this process assumes that and does not include backup/restoration of the database[14]. In the event that the database is changed with a rolling interim patch, such as is the case with Oracle CPU patches, see Appendix C for guidelines.

The best practice process for this category as follows:

1. Verify that the patch to be applied is a rolling patch. For example:
```
$ opatch query -all | grep rolling
  Patch is a rolling patch: true
```
2. Ensure that the services used by the application are available on more than one instance and that the application has been tested with multiple service members.
3. Stop and disable the services on the instances that are using the `ORACLE_HOME` to be patched. For example:
```
srvctl stop service -d gen -s oltp -i gen1
srvctl disable service -d gen -s oltp -i gen1
```
4. Use SRVCTL to `SHUTDOWN IMMEDIATE` and disable the instances (ASM or RDBMS) and listeners using the `ORACLE_HOME` to be patched. For example:
```
srvctl stop instance -d gen -i gen1 -o immediate
srvctl disable instance -d gen -i gen1
srvctl stop asm –n node1
srvctl disable asm –n node1
srvctl stop listener –n node1
srvctl disable listener –n node1
```
5. Apply the Oracle rolling patch against the affected `ORACLE_HOME` and/or `ASM home` using the OPatch utility. Instructions to perform this are detailed in the patch Readme file
6. Use SRVCTL to enable and startup the instances (ASM or RDBMS) and listeners using the `ORACLE_HOME` to be patched. For example:
```
srvctl enable instance -d gen -i gen1
srvctl start instance -d gen -i gen1
srvctl enable asm –n node1
srvctl start asm –n node1
```

[14] In general, a database change would come in the form of a SQL script packaged with the patch and referenced in the Readme file. For example, the Oracle July 2007 CPU Bundle patch for release 10.2.0.3 is a rolling interim patch and it includes the execution of the `CATCPU.SQL` script after the patch is applied

```
srvctl enable listener –n node1
srvctl start listener –n node1
```

7. Assess the health of the system with a subset of the application workload to ensure the changes have not caused a problem.
8. Enable and start the services on instances that were shut down. This generates a FAN event that, along with Run Time Connection Load Balancing, causes the application to send work to the instances again. For example:
   ```
   srvctl enable service -d gen -s oltp -i gen1
   srvctl start service -d gen -s oltp -i gen1
   ```
9. Repeat steps 2-8 on next node until all nodes are patched.
10. If the service configuration was changed, restore it to its initial state.

If something goes wrong during the patch installation, OPatch cannot rollback patches in a rolling fashion the same way that it applies patches in a rolling fashion. The best practice therefore, assuming the patch is not causing a serious problem that needs to be eliminated immediately, is to rollback the patch on each node individually using the `-LOCAL` option.

RDBMS and ASM Non-Rolling Interim Patches

Oracle interim patch 4381035 for 10.2.0.3 is an example of planned maintenance that falls into this category.

Most Oracle RDBMS and ASM non-rolling interim patches do not change the database so this process assumes that and does not include backup/restoration of the database. In the event that a non-rolling interim patch changes the database, see Appendix C for guidelines.

Recall for this activity we break up the cluster into two sets of nodes and patch in a way to maximize availability. The two sets of nodes will be referred to as the "first set" and "second set." When deciding how to break up your cluster into two sets, ensure that the smallest set of nodes can handle the application workload.

The best practice process for this category is as follows:

1. Stop and disable the services on the first set of instances that are using the `ORACLE_HOME` to be patched. For example:
   ```
   srvctl stop service -d gen -i gen1 -s oltp
   srvctl disable service -d gen -i gen1 -s oltp
   srvctl stop service -d gen -i gen2 -s oltp
   srvctl disable service -d gen -i gen2 -s oltp
   ```
2. Perform a `SHUTDOWN IMMEDIATE` and disable the first set of instances (RDBMS and/or ASM) that are using the `ORACLE_HOME` to be patched. Example (listeners not included here but they must also be considered if using the `ORACLE_HOME` being patched):

```
srvctl stop instance -d gen -i gen1 -o immediate
srvctl disable instance -d gen -i gen1
srvctl stop instance -d gen -i gen2 -o immediate
srvctl disable instance -d gen -i gen2
srvctl stop asm -n node1
srvctl disable asm -n node1
srvctl stop asm -n node2
srvctl disable asm -n node2
```

3. Apply the interim patch using the -`MINIMIZE_DOWNTIME` option of OPatch. OPatch asks you some questions about the sets of nodes.

4. When OPatch has completed patching the first set of nodes and says they are ready to be started, perform a `SHUTDOWN IMMEDIATE` and disable the second set of (RDBMS and/or ASM) instances that are using the `ORACLE_HOME` to be patched **and then** enable and startup the first set of instances (RDBMS and/or ASM) using the patched code. For example:

```
srvctl stop instance -d gen -i gen3 -o immediate
srvctl disable instance -d gen -i gen3
srvctl stop asm -n node3
srvctl disable asm -n node3
srvctl enable asm -n node1
srvctl start asm -n node1
srvctl enable asm -n node2
srvctl start asm -n node2
srvctl enable instance -d gen -i gen1
srvctl start instance -d gen -i gen1
srvctl enable instance -d gen -i gen2
srvctl start instance -d gen -i gen2
```

5. Enable/start the services on the first set of instances that are using the `ORACLE_HOME` that has been patched. This generates a FAN event that, along with Run Time Connection Load Balancing, causes the application to start sending work to the instances again. For example:

```
srvctl enable service -d gen -i gen1 -s oltp
srvctl start service -d gen -i gen1 -s oltp
srvctl enable service -d gen -i gen2 -s oltp
srvctl start service -d gen -i gen2 -s oltp
```

6. OPatch is now ready to apply the patch to the second set of instances. Once it is done with that, enable and startup the second set of instances (RDBMS and/or ASM) and services:

```
srvctl enable asm -n node3
srvctl start asm -n node3
srvctl enable instance -d gen -i gen3
srvctl start instance -d gen -i gen3
```

```
srvctl enable service -d gen -i gen3 -s oltp
srvctl start service -d gen -i gen3 -s oltp
```

Using this approach, the only *application blackout* is the time taken to shut down the first set of nodes, start the second set of nodes, and start the service on the first set of nodes. This is illustrated in steps 4 and 5.

If something goes wrong during the patch installation, the fallback plan depends on the version of Opatch you are using. If you are using:

- OPatch 10.2 (or higher)— The best practice is to roll back the patch with the `MINIMIZE_DOWNTIME` option.
- A pre-10.2 version of Opatch—`MINIMIZE_DOWNTIME` is not supported so the best practice is to simulate the `MINIMIZE_DOWNTIME` feature by rolling back the patch on each subset of nodes using the `-LOCAL` option.

Both of these options assume the patch is not causing a serious problem that needs to be eliminated immediately in which case the database would be shut down and the patch rolled back on all nodes at once.

RDBMS and ASM Non-Rolling Patch sets

An example of planned maintenance for RDBMS and ASM non-rolling patch set is Oracle RDBMS/ASM patch set 10.2.0.3

Recall that higher levels of availability can be achieved for patch set upgrades using SQL Apply Rolling Upgrade if your database meets the minimum requirements. See the SQL Apply Rolling Upgrade Best Practices: Oracle Database 10g Release 2 [7] white paper for details.

If you want the highest level of availability possible using just Oracle Clusterware and Oracle RAC, apply a patch set using an out-of-place upgrade. Before attempting this process, ensure you have the fix for bug 5888210.

The best practice process for this category as follows:

1. Clone existing software tree into a new `ORACLE_HOME` using the documented Oracle cloning techniques. Every node in the cluster must do this independently. It is important to perform a verification check after copying the software so that any changed files are acceptable. For example:

```
# This should be run as root to ensure copy permissions/ownership
echo "Pre-clone copy started at `date`" | tee -a clone.log
cp -pr /u01/app/oracle/product/10.2.0 /u01/app/oracle/product/10.2.0.3
echo "Pre-clone copy ended at `date`" | tee -a clone.log
echo "Verification check started at `date`" | tee -a clone.log
diff -q -r /u01/app/oracle/product/10.2.0 /u01/app/oracle/product/10.2.0.3 |
tee -a clone_diff.out
echo "Verification check ended at `date`" | tee -a clone.log
echo "Clone started at `date`" | tee -a clone.log
su - oraha -c "perl /u01/app/oracle/product/10.2.0.3/db/clone/bin/clone.pl
ORACLE_HOME=/u01/app/oracle/product/10.2.0.3/db ORACLE_HOME_NAME=10gR2P2 '-
O"CLUSTER_NODES={node1,node2,node3}"' '-O"LOCAL_NODE=node1"'"
/u01/app/oracle/product/10.2.0.3/db/root.sh << eof
/usr/local/bin
```

```
eof
echo "Clone ended at `date`" | tee -a clone.log
```

2. Apply the RDBMS patch set to the cloned ORACLE_HOME. This is a typical patch set installation performed using the OUI. The patch set Readme instructions should be followed.

3. Create a guaranteed restore point that you can use to back out database changes if required[15]. For example:

```
create restore point patchset1 guarantee flashback database;
```

4. Stop and disable the services affected by the upgrade. For example:

```
srvctl stop service -d gen -s oltp
srvctl disable service -d gen -s oltp
```

5. Stop and disable the databases, listeners, and/or ASM instances using the affected Oracle home across the entire cluster. For example:

```
srvctl stop database -d gen
srvctl disable database -d gen
srvctl stop asm -n node1
srvctl disable asm -n node1
srvctl stop asm -n node2
srvctl disable asm -n node2
srvctl stop listener -n node1
srvctl disable listener -n node1
srvctl stop listener -n node2
srvctl disable listener -n node2
```

6. Change the Oracle homes for the databases, listeners, and ASM instances affected. **Note:** To successfully complete all of these commands, you must use a release 10.2.0.3 or higher version of SRVCTL. For example:

```
srvctl modify asm -n node1 -i +ASM1 -o
/u01/app/oracle/product/10.2.0.3/asm
srvctl modify listener -n node1 -o
/u01/app/oracle/product/10.2.0.3/asm
srvctl modify asm -n node2 -i +ASM2 -o
/u01/app/oracle/product/10.2.0.3/asm
srvctl modify listener -n node2 -o
/u01/app/oracle/product/10.2.0.3/asm
srvctl modify database -d gen -o
/u01/app/oracle/product/10.2.0.3/db
```

7. Enable and start the listeners and/or ASM instances with the patched code. For example:

```
srvctl enable asm -n node1
srvctl start asm -n node1
```

[15] This guaranteed restore point can be dropped once the application has signed off that the patch set installation was a success

```
srvctl enable asm -n node2
srvctl start asm -n node2
srvctl enable listener -n node1
srvctl start listener -n node1
srvctl enable listener -n node2
srvctl start listener -n node2
```

8. Execute the normal post-patch set installation commands as described in the patch set Readme document. For Oracle RAC, this includes things like setting `CLUSTER_DATABASE` to `FALSE`, starting up in `UPGRADE` mode, running the `CATUPGRD.SQL` and `UTLRP.SQL` scripts, setting `CLUSTER_DATABASE` back to `TRUE`, and so on.
9. If the database is not already running, enable and start the database with the patched code. For example:
```
srvctl enable database -d gen
srvctl start database -d gen
```
10. Enable and start the affected services. For example:
```
srvctl enable service -d gen -s oltp
srvctl start service -d gen -s oltp
```

Using this approach, the only *application blackout* time is the time it takes to:

1. Shut down affected components (usually < 10 minutes)
2. Change the Oracle home in the OCR (seconds)
3. Run the post patch set scripts (dependent on scripts that need to be run, which is generally between 15 and 60 minutes)
4. Start affected components (usually less than 5 minutes)

If something goes wrong during the patch set installation, the fallback plan is as follows:

1. Stop and disable the services affected by the upgrade. For example:
```
srvctl stop service -d gen -s oltp
srvctl disable service -d gen -s oltp
```
2. Stop the database, stop/disable listeners, and/or ASM instances using the affected Oracle home across the whole cluster. For example:
```
srvctl stop database -d gen
srvctl disable instance -d gen -i gen1
srvctl disable instance -d gen -i gen2
srvctl stop asm -n node1
srvctl disable asm -n node1
srvctl stop asm -n node2
srvctl disable asm -n node2
srvctl stop listener -n node1
srvctl disable listener -n node1
```

```
srvctl stop listener -n node2
srvctl disable listener -n node2
```

3. If ASM and the listener were affected, change the associated OCR Oracle home, and enable/start the components. For example:

```
srvctl modify asm -n node1 -i +ASM1 -o /u01/app/oracle/product/10.2.0/asm
srvctl modify listener -n node1 -o /u01/app/oracle/product/10.2.0/asm
srvctl modify asm -n node2 -i +ASM2 -o /u01/app/oracle/product/10.2.0/asm
srvctl modify listener -n node2 -o /u01/app/oracle/product/10.2.0/asm
srvctl enable listener -n node1
srvctl start listener -n node1
srvctl enable listener -n node2
srvctl start listener -n node2
srvctl enable asm -n node1
srvctl start asm -n node1
srvctl enable asm -n node2
srvctl start asm -n node2
```

4. Flashback the database to the guaranteed restore point. This includes enabling and starting one instance with the patched code, executing the `FLASHBACK` command, shutting down and disabling the instance with abort, changing the Oracle home in the OCR, starting up the instance with the old code, and issuing the `OPEN RESETLOGS`. For example:

```
srvctl enable instance -d gen -i gen1
srvctl start instance -d gen -i gen1 -o mount
flashback database to restore point patchset1;
srvctl stop instance -d gen -i gen1 -o abort
srvctl disable instance -d gen -i gen1
srvctl modify database -d gen -o /u01/app/oracle/product/10.2.0/db
srvctl enable instance -d gen -i gen1
srvctl start instance -d gen -i gen1 -o mount
alter database open resetlogs;
```

   This further illustrates why it is a best-practice to keep a separate ASM home for ASM and to run the listener out of the most recent Oracle home. By following those two best practices, you have less dependency and impact both during patching and fallback.

5. Enable the instances and start the database if necessary. For example:

```
srvctl enable instance -d gen -i gen1
srvctl enable instance -d gen -i gen2
srvctl start database -d gen
```

6. Enable/startup the affected services. For example:

```
srvctl enable service -d gen -s oltp
srvctl start service -d gen -s oltp
```

RDBMS and ASM Online Interim Patches

Note: *While the rest of this document is pertinent to Oracle 10g Release 2, the sections describing online patching are only relevant to Oracle 11g Release 1. Online patching is not supported in Oracle 10g Release 2.*

Examples of planned maintenance that fall into the online interim patches category include:

- Diagnostic patches
- Simple bug fixes whose patch has been created to be applied online

The best practice process for this category as follows:

1. Verify that the patch to be applied is an online patch. For example:
   ```
   $ opatch query –is_online_patch
     Patch is a online patch: true
   ```
2. Ensure the patch has been tested in a test environment so that memory requirements are fully understood
3. Apply the Oracle online patch against the affected `ORACLE_HOME` and instance using the `-LOCAL` option of the OPatch utility. Instructions to perform this are detailed in the patch Readme file. An example of applying an online patch is as follows:
   ```
   $ opatch apply -local -connectString inst1::::
   ```
4. Assess the health of the system with a subset of the application workload to ensure the changes have not caused a problem
5. Repeat on next node until all nodes are patched

If something goes wrong during the patch installation, the fallback plan is to roll back the patch using OPatch using the `-LOCAL` option on every instance. The following command shows an example for one instance:

```
$ opatch rollback -id 123456 -local -connectString inst1::::
```

Oracle Clusterware, RDBMS, and ASM Add and Remove Node

Examples of planned maintenance that fall into the add/remove node category include:

- Adding/Removing a node to a cluster from an Oracle Clusterware perspective
- Adding/Removing a node to a cluster from an Oracle RDBMS / ASM perspective

The best practice process for this category is as follows:

1. Backup the Oracle inventory and any affected software
2. For Oracle Clusterware operations, backup the OCR. For example:
   ```
   # ocrconfig -export ocr.exp
   ```

Note this operation should be done when no cluster configuration changes are taking place so that the export taken is consistent.

3. Execute the add/remove node procedures defined in *Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* (Part B14197) [5]

Note that you can also use the Oracle cloning technology to accomplish the provisioning detailed here. This technique is advantageous for mass deployment cases. See the Oracle Universal Installer and OPatch Users Guide for details.

If something goes wrong during the add/remove node procedure, consider using the following guidelines for fallback:

1. If the software home is registered in the inventory, remove the software using the OUI.
2. For Oracle Clusterware add/remove node, if the error occurred during or after the OCR was updated, restore the OCR using the logical backup.
3. For fallback during add node, physically remove the software from the software home to prevent OUI warning messages when you retry the operation.
4. Retry the operation.

RDBMS Instance, ASM Instance, Listener, and Service Addition and Removal

Examples of planned maintenance that fall into this category include:

- Adding/removing Oracle RAC instances
- Adding/removing ASM instances
- Adding/removing listeners
- Adding/removing database services

The best practice process for this category is as follows:

1. For instance addition and removal, backup the associated spfile or pfile.
2. Use the DBCA utility for Oracle RAC instance, ASM instance, listener, and services addition/removal. When adding a brand new node, all can be done together in one step. Just proceed through the `ADD INSTANCE` screens and the DBCA will recommend ASM and listener extension.
3. If not done in step 2, use the NETCA utility to add/remove listeners.

If something goes wrong during the addition/removal, the following guidelines should be considered for fallback:

1. Use DBCA to undo any configurations that occurred prior to the failure
2. Restore the backup spfile or pfile if necessary.

3. Retry the operation

APPENDIX A: ENTERPRISE MANAGER GRID CONTROL

You can automate many of the operational best practices defined in this document through the Enterprise Manager Grid Control Deployment Procedures. Oracle will be providing best practice white papers for these in the near future.

APPENDIX B: ORACLE SOFTWARE RECONFIGURATION AFTER CLONING

If you are using Enterprise Manager Database Control, then you need to edit the following configuration files in the cloned `ORACLE_HOME` directories. In the `ORACLE_HOME/sysman` directory, change the `$ORACLE_HOME` environment variable to specify the new `ORACLE_HOME` location:

- config/emca.properties
- config/emd.properties
- config/emd.properties.emca
- config/emagentlogging.properties
- config/emomslogging.properties
- lib/env_emdb.mk
- lib/env_sysman.mk

Note: If the source `$ORACLE_HOME` contains any hard links or symbolic links, then these will also be cloned. If any of these links in the cloned `ORACLE_HOME` reference the source `ORACLE_HOME`, then you must manually change them to specify the cloned `ORACLE_HOME` path.

APPENDIX C: GUIDELINES WHEN AN INTERIM PATCH CHANGES THE PERSISTENT DATA STORAGE

If an Oracle Clusterware interim patch changes the OCR, you must backup the OCR prior to applying the patch so you can restore the OCR if there is a problem. The recommended method is to take a logical backup using the `-EXPORT` option of the `OCRCONFIG` command. It is imperative that you do this at a time when the OCR is not being changed using other SRVCTL commands to ensure a consistent backup is made. If a restore is needed, use the `-IMPORT` option of the `OCRCONFIG` command.

If an Oracle RDBMS interim patch changes the database, create a guaranteed restore point prior to applying the patch. If a problem occurs during the patch application, then flash back the database to the guaranteed restore point. This includes stopping the database, starting one instance in mount mode, executing the `FLASHBACK` command, and issuing the `OPEN RESETLOGS` command. For example:

```
srvctl stop database -d gen
srvctl start instance -d gen -i gen1 -o mount
flashback database to restore point patchset1;
alter database open resetlogs;
```

REFERENCES

1. Oracle Maximum Availability Architecture
2. *Oracle Database High Availability Overview 10g Release 2 (10.2)* (Part #B14210)
3. *Oracle Database High Availability Best Practices 10g Release 2 (10.2)* (Part B25159)
4. *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide 10g Release 2 (10.2) for Linux* (Part B14203)
5. *Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide 10g Release 2 (10.2)* (Part B14197)
6. Client Failover Best Practices for Highly Available Oracle Databases: Oracle Database 10*g* Release 2 at http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_ClientFailoverBestPractices.pdf
7. SQL Apply Rolling Upgrade Best Practices: Oracle Database 10g Release 2
8. Oracle Database 10*g* Release 2 Best Practices: Optimizing Availability During Unplanned Outages Using Oracle Clusterware and RAC at http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_FastRecoveryOracleClusterwareandRAC.pdf
9. Oracle® Universal Installer and OPatch User's Guide11g Release 1 (11.1) (Part B31207) for Windows and UNIX
10. Real Application Clusters Installation Guide for Linux and UNIX (11.1) (Part 28264)
11. Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide 10g Release 2 (11.1) (Part B28254)

Best Practices for Optimizing Availability During Planned Maintenance Using Oracle Clusterware and Oracle Real Application Clusters
September 2007
Author: Michael D. Nowak
Contributors: Roland Knapp

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