Implementing TimeFinder Multiple Snapshots with Virtual Provisioning for Oracle 11g Databases on EMC Symmetrix VMAX

Applied Technology

Abstract

EMC® Symmetrix VMAX™ storage arrays provide the capability to support up to 128 TimeFinder®/Snap sessions per single source volume. This white paper provides details on the compatibility and usage of this new functionality with Oracle 11g databases.

January 2011
Implementing TimeFinder Multiple Snapshots with Virtual Provisioning for Oracle 11g Databases on EMC Symmetrix VMAX

Applied Technology
Table of Contents

Executive summary ................................................................. 4
Introduction ............................................................................. 4
  Audience ............................................................................. 5
TimeFinder/Snap ...................................................................... 5
  TimeFinder/Snap copy sessions ............................................ 6
  TimeFinder Consistent Split .................................................. 7
  Virtual Provisioning .............................................................. 8
Symmetrix Auto-provisioning Groups ........................................ 11
Oracle ASM ............................................................................ 11
  New features in 11g release 2 for Grid Infrastructure ................ 12
  Oracle ASM and Oracle files .................................................. 12
  ASM job role separation option with SYSASM ....................... 12
  Renaming the ASM diskgroup ............................................... 12
  DBNEWID utility .................................................................. 13
Implementing high-availability solutions with TimeFinder/Snap for Oracle databases ........................................ 14
  TimeFinder/Snap preparation for Oracle Database .................... 15
    Enabling the TimeFinder/Snap multi-virtual snap feature ........ 15
    Creating a TimeFinder/Snap copy session for Oracle Database . 15
    Creating a restartable database image using TimeFinder/Snap ... 16
  TimeFinder/Snap operation on the source host ....................... 16
  Starting Oracle Database on a target host ............................. 16
  Creating multiple rolling point-in-time restartable database images 17
    Creating the first TimeFinder/Snap point-in-time database image 17
    Creating additional TimeFinder/Snap point-in-time database images at regular intervals .... 17
    Re-creating multiple point-in-time database rolling snap images at subsequent time intervals 18
  Mounting and renaming a TimeFinder/Snap restartable database image .................................................. 18
    Renaming the ASM diskgroup replica .................................. 18
    Creating a database hot backup image using TimeFinder/Snap 22
      TimeFinder/Snap operation on the source host .................. 22
      Starting an Oracle database on a target host after snap activation with hot backup 23
      Incremental restore to source database devices ................. 24
      TimeFinder/Snap restoring operation on the source host .......... 24
      Database recovery from a restartable snap ....................... 25
      Terminating TimeFinder/Snap sessions ............................. 27
      Terminating and discarding a TimeFinder/Snap session ........ 27
Conclusion .............................................................................. 27
References .............................................................................. 27
Appendix: Test configuration .................................................... 28
Executive summary

EMC™ Symmetrix VMAX™ is the newest addition to the Symmetrix product family. Built on the strategy of simple, intelligent, modular storage, it incorporates a new scalable Virtual Matrix™ interconnect that connects all shared resources across all VMAX Engines, allowing the storage array to grow seamlessly and cost-effectively from an entry-level configuration into the world’s largest storage system. The Symmetrix® VMAX provides improved performance and scalability for demanding enterprise storage environments while maintaining support for EMC’s broad portfolio of platform software offerings.

Virtual Provisioning™, generally known in the industry as “thin provisioning,” enables organizations to improve ease of use, enhance performance, and increase capacity utilization for certain applications and workloads. Implementing Virtual Provisioning for VMAX storage arrays also improves storage infrastructure utilization, as well as associated operational requirements and efficiencies.

EMC® TimeFinder®/Snap functionality provides customers with a space-efficient mechanism to create pointer-based replicas of production systems. Rather than creating completely independent copies of production volumes, which carry the expense of provisioning a full-size set of target volumes, TimeFinder/Snap utilizes a pointer-based mechanism that only requires storage for changed data. This pointer-based device provides a complete point-in-time replica of the source device while reducing the amount of additional storage required. The storage pool for the changed data is called a save pool. More than one save pool can be created and snap devices will be sharing storage in the save pool assigned to them.

The EMC TimeFinder product line provides a way of creating and restoring database replicas that are fully available in seconds, regardless of the database size. TimeFinder/Snap, in particular, allows enterprise users to make pointer-based, space-saving copies of data simultaneously on multiple target devices from a single source device. The data is available to the target host instantly. Since Symmetrix DMX™ Enginuity™ service release 5772.83.75 and continuing with VMAX, the limit of active TimeFinder/Snap sessions has been increased from 16 active snapshot sessions of a single source to 128 per source. This change is meant to help organizations that need to scale active snap sessions per source beyond the previous limitations. The Symmetrix VMAX Enginuity 5874 service release of Q4 2009 expanded TimeFinder multi-snap support to Virtual Provisioning, allowing up to 128 TimeFinder/Snap sessions per thin device source.

Extending TimeFinder/Snap capacity to 128 simultaneous sessions per source allows a larger number of restore points (snapshots) for Oracle environments because the system can take snaps at regular intervals. Therefore, users can restore the Oracle database(s) to intervals in the past without needing the storage overhead of multiple full copies.

This paper also discusses a use case where TimeFinder snapshots of the Oracle database are taken frequently (perhaps every 3 or 4 hours), using consistency technology without placing the database in hot backup mode. The snapshots can be mounted (or restored back to production volumes) and the database can perform a database media recovery, or roll-forward of the archive logs. This use case allows very short RTOs because only archive logs since the last snapshot need to be applied (in this example the worst case is 3 or 4 hours of archive logs, compared to 24 hours if the last backup was taken the night before).

Introduction

This white paper describes the application of EMC TimeFinder/Snap and its multi-snap functionality for Oracle 10g and 11g databases, with particular focus on using Oracle Automated Storage Management (ASM) and Linux in conjunction with Symmetrix VMAX TimeFinder technology. In a similar way, other operating systems or volume managers can be used. The procedures cover the steps necessary to initiate a configuration utilizing TimeFinder/Snap and its multi-snap features to create space-saving images against a running, active database.

EMC TimeFinder/Snap enables Oracle customers to back up, resynchronize, and restore large amounts of data without jeopardizing data integrity. The topics covered in this paper explain how to implement TimeFinder/Snap with TimeFinder Consistent Split technology to create restartable and recoverable...
database replicas. The business cases of using TimeFinder/Snap and Consistent Split technology for a mission-critical Oracle database solution include the following:

- To create a space-saving image of the database that can be used to create a second instance for testing, development, or reporting
- To create frequent, periodic, point-in-time database restartable images for data protection ("rolling" snaps that allow Oracle database restart)
- To create frequent, periodic, point-in-time database recoverable images for data protection ("rolling" snaps that allow Oracle recovery)
- To create a database hot backup image for backup
- To restore images back to the original databases

The paper describes how to use EMC TimeFinder Consistent Split technology, **consistent activation** (using the -consistent option with the symsnap activate command), to create valid, point-in-time restartable and/or recoverable images of the Oracle database.

As business enterprises experience tremendous growth, they need to meet rapidly changing environments. So organizations’ ability to maintain significantly greater numbers of point-in-time copies of a single source, in conjunction with the space efficiency provided by TimeFinder/Snap, allows them to significantly increase the granularity of these restore or restart points. This provides an effective mechanism to move the source system to a desired state from any given point, as well as the ability to present a particular transactional state of an active Oracle database to an alternative server.

**Audience**

This white paper is intended for Oracle database administrators, systems administrators, and storage administrators responsible for managing Oracle database environments.

**TimeFinder/Snap**

TimeFinder/Snap software allows users to make multiple pointer-based, space-saving copies of data simultaneously on multiple target devices from a single source device. The resulting point-in-time copy of data is available to the target host for access instantly. TimeFinder/Snap allows data to be copied from a single source device to as many as 128 target devices simultaneously, where the source devices can be either a Symmetrix standard device or a business continuance volume (BCV) device controlled by TimeFinder/Mirror or TimeFinder/Clone. The only exception is a BCV working in TimeFinder/Clone’s clone emulation mode. Starting with the Symmetrix Enginuity Q4 2009 service release, the source can be a Symmetrix thin device (part of Symmetrix Virtual Provisioning as explained later in more detail). The target devices are Symmetrix virtual devices (VDEV) (see Table 1) that consume negligible physical storage through the use of pointers to track changed data.

The VDEV is a host-addressable Symmetrix device with special attributes created at configuration time. But unlike a TimeFinder/Clone target device (standard or BCV), which contains a full volume of data, the VDEV is a logical-image device that offers a space-saving way to create instant point-in-time copies of logical volumes. Any updates to a source device after its activation with a virtual device cause the preupdate image of the changed tracks to be copied to specifically created target volumes, called *save devices*, within the Symmetrix. These save devices are aggregated into save device pools, to which VDEVs are assigned. The virtual device’s indirect pointer is then updated to point to the original track data on the save device, preserving a point-in-time image of the volume. TimeFinder/Snap uses this copy-on-first-write (COFW) technique to conserve disk space, since only changes to tracks on the source cause any incremental storage to be consumed. A summary of the differences between devices used in TimeFinder/Snap operations is listed in the following table.
Table 1. Definitions of Time/Snap devices

<table>
<thead>
<tr>
<th>Device</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual device</td>
<td>A logical-image device that saves disk space through the use of pointers to track data that is immediately accessible. Snapping data to a virtual device uses a COFW technique.</td>
</tr>
<tr>
<td>Save device</td>
<td>A device that is not host-addressable but accessed only through the virtual devices that point to it. Save devices provide a pool of physical space to snap-copy data to which virtual devices point.</td>
</tr>
<tr>
<td>Standard or BCV thick device</td>
<td>A full-size device that can be set as a source to a virtual device.</td>
</tr>
<tr>
<td>Clone</td>
<td>An independent replica of a source device (standard or BCV) that consumes at least the same storage capacity as the source device.</td>
</tr>
<tr>
<td>Thin device</td>
<td>A host-accessible device that has no storage directly associated with it.</td>
</tr>
<tr>
<td>Data devices</td>
<td>Internal devices that when placed in a thin pool provide storage capacity to be used by thin devices.</td>
</tr>
<tr>
<td>Thin pool</td>
<td>A collection of data devices that provide storage capacity for thin devices.</td>
</tr>
<tr>
<td>Standard or BCV thin device</td>
<td>A thin device based on Symmetrix Virtual Provisioning technology that can be set as a source to a virtual device.</td>
</tr>
</tbody>
</table>

As of Enginuity 5772, TimeFinder/Snap source devices no longer incur a penalty associated with COFW technology. The COFW penalty means that if a track is written to the source volume (only the first time) and it has not yet been copied to the Snap save area, it must first be copied to the save area before the write from the host is acknowledged. In later revisions of Enginuity, an Asynchronous Copy on First Write (ACOFW) operation is implemented, such that update operations from the production host occur immediately, and Enginuity manages the transfer of the original preimaged data to the save device in the background. This eliminates the host response time impact that is common with applications leveraging COFW architectures and represents a significant increase in performance. Storage groups copied using TimeFinder/Snap in earlier versions of Enginuity are subject to a COFW penalty only while the snap is activated. Starting with the Q4 2009 service release of Enginuity 5874, TimeFinder/Snap supports the creation of multi-virtual snap sessions from thin devices, which is a strong endorsement for adopting Virtual Provisioning technology (see the “Virtual Provisioning” section on page 8).

**TimeFinder/Snap copy sessions**

TimeFinder/Snap functionality is controlled using copy sessions, which pair the source and target devices. Sessions are maintained on the Symmetrix array and can be queried to verify the current state of the devices. A copy session must first be created, a process that identifies the snap devices in the operation. After the snap session create operation, you must activate it. At this point, the snap target devices become accessible to their host. When the copy session information is no longer required, the copy session can be terminated.

TimeFinder/Snap operations are controlled from the host by using symsnap commands to create, activate, terminate, and restore the source/target snap pairs of copy sessions. The TimeFinder/Snap operations described in this section explain how to manage the devices participating in a copy session through SYMCLI. Alternatively, you can manage the TimeFinder operations from the Symmetrix Management Console (SMC) graphic user interface.

The symsnap command is used to enable TimeFinder/Snap operations. The operation happens in two phases: creation and activation. The creation phase builds bitmaps of the source and target that are later used to manage the changes on the source and target. The execution of a symsnap create command does not actually copy the data from the source volume to the target volume.
Activation of the snap session created in the previous `create` command can be accomplished by executing the `symssnap activate` command. The activation of a snap enables the protection of the source data tracks. When protected tracks are first changed on the source volume, they are copied into the save pool and the VDEV pointers are updated to point to the changed tracks in the save pool. The source track is then marked as unprotected. Subsequent changes to that track will not result in any VDEV or SAVE device activity for the current TimeFinder/Snap sessions. When tracks are changed on the VDEV, the associated source track is unprotected, and the new data is written directly to the save pool and the VDEV pointers are updated in the same way.

When multiple TimeFinder/Snap sessions are activated against a source volume and a protected track is updated, the copy of the track to the save pool is executed once. Next, the snap session pointers for which that track is protected are updated to point to the same location. In this manner, usage of the snap pool space is optimized so that multiple sessions can use the same track allocations. Snap pool tracks will be released when all session pointers for activated snap sessions have been terminated.

A TimeFinder/Snap session can be restored to the source devices. In this case the restore will be incremental and only the differences between source and target devices (as reflected in the save pool) will be copied back to the source devices. Alternatively, a TimeFinder/Snap session can be used to do a full restore of the target data to an unrelated device. Finally, it can also perform an incremental restore to a BCV device that was split earlier from the same source.

When the copy session is terminated, the virtual device image is released; the save device space associated with the session is freed and returned to the save device pool for future use.

Figure 1 illustrates a virtual copy session where the controlling host creates a copy of standard device DEV001 on target device VDEV005.

**Figure 1. TimeFinder/Snap copy of a standard device to a VDEV**

**TimeFinder Consistent Split**

With TimeFinder, you can use the Enginuity Consistency Assist (ECA) feature to perform consistent splits between source and target device pairs across multiple, heterogeneous hosts. Consistent split (which is an implementation of instant split) helps to avoid inconsistencies and restart problems that can occur if you
split database-related devices without first quiescing\(^1\) the database. The difference between a normal instant split and a consistent split is that in a consistent split, the database writes are held at the storage level momentarily while the foreground split occurs, maintaining dependent-write order consistency on the target devices comprising the group. Since the foreground instant split completes in just a few seconds, Oracle needs to be in hot backup mode only for this short time when hot backup is used. When consistent split alone is used to create a restartable replica, interference with business operations is minimal.

After performing a consistent split, TimeFinder target devices are in a state that is equivalent to the state a database would be in after a power failure, or if all database instances were aborted simultaneously. Oracle is familiar with this state and it can recover easily from it by performing a crash recovery the next time the database instance is started.

Note: Consistent activation is used to create a restartable image of the database at a specific point in time. Even when hot backup mode is used, consistent split provides a real advantage, especially when Oracle ASM is used, because it preserves ASM metadata consistency during a split operation, regardless of whether ASM is in the midst of a rebalance or not. For that reason, when the database uses ASM, even when offloading backups with TimeFinder while the database is in hot backup mode, you should always use consistent split should to preserve ASM metadata integrity during the split.

In addition to creating recoverable database replicas using TimeFinder when the Oracle database is in hot backup mode, or creating restartable database replica when consistent split is used without Oracle hot backup mode, Oracle also supports various database recovery scenarios based on dependent write consistent storage replicas (consistent split replica). Oracle support is covered in metalink note ID 604683.1. The purpose of this use case is not to replace backup strategy such as nightly backups based on hot backup mode. Instead, it offers a complementary use case such as when RTO requirements are very strict. It could be a compelling solution to run the database in archivelog mode, and perform periodic snapshots without placing the database in hot backup mode. If recovery is required, the last snapshot is restored, and the limited transactions since that snapshot was taken are rolled forward, creating an extremely fast database recovery solution.

TimeFinder/Snap along with Consistent Split technology enables Oracle customers to back up, resynchronize, and restore large amounts of data without jeopardizing data integrity. This paper describes how you can use a consistent activation (using the –consistent option with the symsnap activate command) to create valid, point-in-time, restartable and/or recoverable images of the Oracle database.

In summary, TimeFinder/Snap provides the following advanced technology solution for Oracle 10g and 11g databases:

- TimeFinder/Snap backup and recovery capabilities
- TimeFinder/Snap restart capabilities using consistent activation
- TimeFinder/Snap full and incremental restore capabilities

**Virtual Provisioning**

Symmetrix thin devices are logical devices that you can use in many of the same ways that Symmetrix devices have traditionally been used. Unlike traditional Symmetrix devices, thin devices do not need to have physical storage preallocated at the time the device is created and presented to a host (although in some cases customers interested only in the thin pool wide striping and ease of management choose to fully preallocate the thin devices). You cannot use a thin device until it has been bound to a thin pool, also called a shared storage pool. Multiple thin devices may be bound to any given thin pool. The thin pool is comprised of devices called data devices that provide the actual physical storage to support the thin device allocations.

---

\(^1\) Database quiesce here refers to shutting down the database, or placing it in hot backup mode.
When a write is performed to a part of any thin device for which physical storage has not yet been allocated, the Symmetrix allocates physical storage from the thin pool for that portion of the thin device only. The Symmetrix operating environment, Enginuity, satisfies the requirement by providing a block of storage from the thin pool called a thin device extent. This approach reduces the amount of storage that is actually consumed.

The minimum amount of physical storage you can reserve at a time for the dedicated use of a thin device is referred to as a data device extent. The data device extent is allocated from any one of the data devices in the associated thin pool. Allocations across the data devices are balanced to ensure that an even distribution of allocations occurs from all available data devices in the thin pool.

For Symmetrix, the thin device extent size is the same as the data device extent size, which is 12 Symmetrix tracks or 768 KB in size. There is no waste associated with this approach because applications typically allocate large portions of data at one time. Even if the data is striped across multiple LUNs, it will be stored in an adjacent location every time the striping mechanism will wrap over the same LUN, essentially filling up the thin pool extents. For example, Oracle data files are usually created with a size range of a single gigabyte to tens of gigabytes. Even when a host logical volume manager (LVM) is used to stripe the data, such as Oracle ASM, each LUN that is used (or ASM member) will create the next Allocation Unit adjacent to the previous one. In accordance, the thin devices will fill up data device extents completely before moving to the next one in the thin pool. That way the data is striped in the thin pool evenly and without wastage.

Note: There is no reason to match the LVM stripe depth with the thin device extent size. Oracle accesses data either by random single block read/write operations (usually 8 KB in size) or sequentially by reading large portions of data. In either case there is no advantage or disadvantage to match the LVM stripe depth to the thin device extent size as single block read/write operate on a data portion that is smaller than the LVM stripe depth. Sequential operations will simply continue to read data on each LUN (every time the sequential read wraps to that same LUN) regardless of the stripe depth.

When you perform a read on a thin device, the data being read is retrieved from the appropriate data device in the thin pool to which the thin device is associated. If for some reason a read is performed against an unallocated portion of the thin device, zeroes are returned to the reading process.

When more physical data storage is required to service existing or future thin devices, such as when a thin pool is approaching full storage allocations, you can add data devices to existing thin pools dynamically without needing a system outage. New thin devices can also be created and associated with existing thin pools.

When data devices are added to a thin pool they can be in an enabled or disabled state. In order for the data device to be used for thin extent allocation it needs to be in the enabled state. To remove it from the thin pool, you need to disable it. Symmetrix automatically initiates a drain operation on a disabled data device without any disruption to the application. Once all the allocated extents are drained to other data devices, you will be able to remove the data device from the thin pool.

The following figure depicts the relationships between thin devices and their associated thin pools. Thin Pool A contains nine data devices and thin Pool B contains three data devices. There are nine thin devices associated with thin Pool A and three thin devices associated with thin pool B. The data extents for thin devices are distributed on various data devices as shown in Figure 2.
Figure 2. Thin devices and thin pools containing data devices

Due to the way thin extents are allocated across the data devices, a form of thin pool striping occurs. The more data devices are in the thin pool, the wider the striping will be, therefore increasing the number of devices that can participate in application I/O. Currently, the thin extent size for data devices is 12 tracks or 768 KB in size. Future versions of the Enginuity operating environment may change the thin extent size.

The maximum size of a thin device in a Symmetrix VMAX is 240 GB. If you need a larger size, you can create a metavolume comprised of thin devices. EMC recommends concatenating the metavolume rather than striping since the thin pool is already striped using data device extents. Concatenated metavolumes also support fast expansion capabilities, as new metavolume members can be easily appended to the existing concatenated metavolume. This functionality may be applicable when the provisioned thin device has become fully allocated at the host level, and it is required to further increase the thin device to gain additional space. Note that it is not recommended to provision applications with a low number of very large LUNs. The reason is that each LUN provides the host with an additional I/O queue to which the host operating system can stream I/O requests and parallelize the workload. Host software and HBA drivers tend to limit the amount of I/Os per LUN that you can queue at a time. To avoid host queuing bottlenecks under heavy workloads, it is better to provide the application with multiple smaller LUNs rather than a few large LUNs.

Since Virtual Provisioning supports striped metavolumes, you may want to take advantage of this condition because some workloads will benefit from multiple levels of striping (for example, for Oracle redo logs when SRDF/S is used, and host striping is not available).

You can protect data devices in the thin pool with any EMC RAID offering: RAID 1, RAID 5 or RAID 6. Both RAID 1 and RAID 5 protect from a single drive failure, and RAID 6 protects from two drive failures. A RAID 1 group resides on two physical disk drives; a RAID 5 (3+1) group resides on four disk physical drives. When a thin pool of disk is created, it is always made out of similarly configured RAID groups and not individual disks. For example, if eight RAID 5 (3+1) thin devices are created, and added to one pool, the pool has eight RAID 5 devices of four physical disks each. If one of the disks in this pool fails, it means that a single disk from one of the eight RAID 5 groups failed, and that RAID group will continue to service read and write requests, in degraded mode, without data loss. Also, as with any RAID group with a failed disk, Enginuity will immediately invoke a hot sparing operation to restore the RAID group to its normal state. While this RAID group is rebuilding, any of the other RAID groups in the pool can have a disk failure without any loss of data. In this example, with eight RAID groups in the pool there
can be one failed disk in each RAID group in the pool without data loss. In this manner, data stored in the thin pool is no more vulnerable to data loss than any other data stored on similarly configured RAID devices. Therefore, a protection of RAID 1 or RAID 5 for thin pools is acceptable for most applications and RAID 6 is only required if you warrant additional parity protection to protect your system from a failure of two disks in any single RAID group.

**Symmetrix Auto-provisioning Groups**

Enginuity versions 5874 and later provide Auto-provisioning Groups for storage and system administrators who are looking to use a simplified model for storage provisioning. Auto-provisioning Groups are implemented using the `symaccess` CLI command with Solutions Enabler 7.0 and later or by using the Symmetrix Management Console (SMC).

The fundamental concept of Auto-provisioning Groups is the logical grouping of related objects and the creation of a view that associates the related groups together. The following logical groupings are used:

- **Initiator groups**: An initiator group is a logical grouping of up to 32 Fibre Channel initiators (HBA ports) identified by World Wide Names (WWNs), eight iSCSI names, or a combination of both. An initiator group may also contain the name of another initiator group to allow the groups to be cascaded to a depth of one.

- **Port groups**: A port group is a logical grouping of Fibre Channel and/or iSCSI Symmetrix front-end director ports.

- **Storage groups**: A storage group is a logical grouping of up to 4,096 Symmetrix devices. You can use both thick and thin LUNs.

- **Masking views**: A masking view defines an association between one initiator group, one port group, and one storage group. When a masking view is created, the devices in the storage group are mapped to the ports in the port group and masked to the initiators in the initiator group. Depending on the server and application requirements, each server or group of servers may have one or more masking views that associate a set of Symmetrix devices to an application, server, or cluster of servers.

With Auto-provisioning Groups, related initiators (HBAs) are grouped into an initiator group, related front-end ports are grouped into a port group, and related devices are grouped into a storage group. A masking view associates the groups together. At the time the masking view is created, all the required mapping and masking are performed automatically in a single operation. This approach dramatically reduces complexity and simplifies storage allocation.

Oracle databases benefit from Auto-provisioning Groups as the feature makes storage device provisioning, monitoring, and management very easy. In the Oracle Grid model, you can use Auto-provisioning Groups to simplify and manage tasks including changing access to storage devices, host HBA ports, or storage connectivity.

**Oracle ASM**

Automatic Storage Management (ASM) is a no-cost feature of Oracle Database 10g and 11g that provides the database administrator with a simple database storage management interface that is consistent across all server and storage platforms. ASM provides integrated cluster file system and volume management capabilities, and is the foundation of a storage grid for Oracle. In addition, ASM eliminates the need for third-party volume managers and file systems for managing Oracle databases.

As you can see in Figure 3, when ASM is used, storage devices are grouped into ASM diskgroups and the database files inside each diskgroup are striped. While customers can continue to use filenames and directory structures inside ASM diskgroups, these will be visible only by using Oracle tools such as SQLPLUS, Oracle Enterprise Manager, RMAN, or ASMCMD.
New features in 11g release 2 for Grid Infrastructure
The new Oracle 11g release 2 Grid Infrastructure comprises of Oracle Clusterware and Oracle ASM. In this new Grid Infrastructure release, ASM and Oracle Clusterware are installed into a single home directory, which is referred to as the Grid Infrastructure home. The following are the important new features in the grid infrastructure software that are relevant in implementing the test environment for this technology white paper.

Oracle ASM and Oracle files
With this release, Oracle Cluster Registry (OCR) and voting disks can be placed on ASM. This feature enables ASM to provide a unified storage solution, storing all the data for the clusterware and the database, without the need for third-party volume managers or cluster file systems. For new installations, OCR and voting disk files can be placed either on ASM, or on a cluster file system or NFS system. Installing Oracle Clusterware files on raw or block devices is no longer supported, unless an existing system is being upgraded.

ASM job role separation option with SYSASM
The SYSASM privilege that was introduced in Oracle ASM 11g release 1 (11.1) is now fully separated from the SYSDBA privilege. The SYSASM privilege also can be granted using password authentication on the Oracle ASM instance. Providing system privileges for the storage tier using the SYSASM privilege instead of the SYSDBA privilege provides a clearer division of responsibility between ASM administration and database administration, and helps to prevent different databases using the same storage from accidentally overwriting each other’s files.

Renaming the ASM diskgroup
11g release 2 introduced the `renamedg` utility to rename diskgroups. The `renamedg` tool enables you to change the name of a cloned diskgroup. The prerequisite for executing this `renamedg` command is to unmount the diskgroup to be renamed on all nodes in the cluster before renaming it. The `renamedg` utility renames a diskgroup using a two-step process and is documented in the Oracle storage administrator’s guide.

Examples of the use of `renamedg`:

```
$ renamedg -dname MYOLDDG -newdname MYNEWDG
```

Figure 3. The operational stack and ASM

The Operational Stack

![The Operational Stack Diagram]

The Operational Stack
Automatic Storage Management
Disks
Logical Vol File System
Tablespace
Tables
File Names
Tablespace
Tables
LVM+FS ASM
Storage
Note: After renaming the ASM diskgroup, the location of a data file that resides in the renamed diskgroup has to be updated before you can open the database.

**DBNEWID utility**

DBNEWID is a database utility that can change the internal database identifier (DBID) and the database name (DBNAME) for an operational database. Changing the DBID of a database is a serious procedure. If you change the DBID of a database, all previous backups and archived logs of the database become unusable. After you change the DBID, you must open the database with the `RESETLOGS` option, which re-creates the online redo logs and resets their sequence to 1. Once you have changed the DBID, you should make a backup of the whole database immediately.

In contrast, changing the DBNAME without changing the DBID does not require you to open with the `RESETLOGS` option, so database backups and archived logs are not invalidated. However, changing the DBNAME does have consequences. You must change the `DB_NAME` initialization parameter after a database name change so that the parameter reflects the new name. Also, you may have to re-create the Oracle password file. If you restore an old backup of the control file (before the name change), then you should use the initialization parameter file and password file that you used before the database name change.

This paper presents use cases and solutions to mount multiple snapshot copies of the product database at different time intervals to a secondary host. In the test scenario, EMC changed the dbname for each point-in-time snapshot copy image of the database in order to mount them on the same target host.

**Changing only the database name**

The following steps describe how to change the database name without changing the DBID.

1. Ensure that you have a recoverable whole database backup. Verify that the target database is mounted but not open, and that it was shut down cleanly prior to mounting.

2. Invoke the utility on the command line, specifying a valid user with the `SYSDBA` privilege. You must specify both the `DBNAME` and `SETNAME` parameters. This example changes the name to `test_db2`:

   ```
   % nid TARGET=SYS/oracle@test_db DBNAME=test_db2 SETNAME=YES
   ```

   DBNEWID performs validations in the headers of the control files (not the data files) before attempting I/O to the files. If validation is successful, then DBNEWID prompts for confirmation, changes the database name in the control files, and exits. After DBNEWID completes successfully, the database is left mounted, but you cannot use it yet. If validation is not successful, then DBNEWID terminates and leaves the target database intact. You can open the database, fix the error, and then either resume the DBNEWID operation or continue to use the database without changing the database name.

3. Shut down the database. Set the `DB_NAME` initialization parameter in the initialization parameter file to the new database name. Create a new password file. Start up the database and resume normal use.
Implementing high-availability solutions with TimeFinder/Snap for Oracle databases

Table 2 summarizes the operations for TimeFinder/Snap business use cases for Oracle databases.

Table 2. Summary of TimeFinder/Snap for Oracle Database high availability

<table>
<thead>
<tr>
<th>Usage</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snap preparation</td>
<td>Enable the multi-virtual snap feature and create copy sessions.</td>
<td>Enable the multi-virtual snap feature. Create a copy session by associating source and target devices. Up to 128 targets can be associated with source devices.</td>
</tr>
<tr>
<td>Hot backup-based recoverable database image</td>
<td>1. Activate a copy session while the database in hot backup mode. 2. Start the database replica using media recovery operations.</td>
<td>Create a backup image by activating the copy session while the Oracle database is open and in hot backup mode.</td>
</tr>
<tr>
<td>Restartable database image</td>
<td>1. Activate a copy session with consistent activation to create a restartable database point-in-time image. 2. Start the database replica using the startup command alone.</td>
<td>Activate the copy session with the consistent activation option without first putting the database in hot backup mode to create a restartable point-in-time image of the database (or multiple databases). Start the database replica on a target host simply using the startup SQL command, or on production after the TimeFinder/Snap session is restored.</td>
</tr>
<tr>
<td>Restartable rolling database images</td>
<td>Create multiple point-in-time restartable database rolling snap images at periodic time intervals.</td>
<td>When implementing rolling TimeFinder/Snap sessions, restartable database(s) images are created frequently. After predefined sessions are active, terminate the oldest session prior to activating the next session. By using consistent activation, no hot backup overhead is introduced to the database, and you can activate multiple related databases together creating a consistent, point-in-time image across all of them.</td>
</tr>
<tr>
<td>Recoverable rolling database images</td>
<td>Create multiple point-in-time recoverable database rolling snap images at periodic time intervals.</td>
<td>When implementing rolling TimeFinder/Snap sessions, recoverable database images are created frequently. After predefined sessions are active, terminate the oldest session prior to activating the next session. By using recoverable images, archive logs can be applied to these images or alternatively to production once the session is restored. This solution reduces RTO drastically since you only need to restore the archive logs created during the time between the previous and current snapshot activations.</td>
</tr>
<tr>
<td>Incremental restore</td>
<td>Restore incrementally to source database devices.</td>
<td>An incremental synchronization (only changed tracks) of source devices from the target devices. As soon as you start the TimeFinder/Snap restore operation, you have access to the production host devices and the restored data, even if the data is being copied in the background.</td>
</tr>
<tr>
<td>Full restore</td>
<td>Perform a full restore from target virtual devices to unrelated Symmetrix devices.</td>
<td>A full synchronization from the target devices to new, unrelated devices. As soon as you start the TimeFinder/Snap restore operation, you have access to the production host devices and the restored data, even if the data is being copied in the background.</td>
</tr>
</tbody>
</table>
**TimeFinder/Snap preparation for Oracle Database**

Enabling the TimeFinder/Snap multi-virtual snap feature

The default behavior of TimeFinder/Snap is to support a total of 16 snapshot sessions per source device. Support for 128 snapshot sessions per device needs to be specifically enabled to take advantage of the functionality. This functionality is enabled by setting a host environment variable in a command shell session:

```bash
SET SYMCLI_MULTI_VIRTUAL_SNAP=ENABLED
```

Creating a TimeFinder/Snap copy session for Oracle Database

You must create a TimeFinder/Snap copy session before you can take a copy image of the database for any business use case. You must activate the session in order for the target virtual devices to be made accessible to their host.

- Create device groups for the TimeFinder/Snap session
  - From the source host, create Symmetrix device groups that include all source and target devices of matching sizes and emulations. There may be more target devices than source devices if you plan to create more than one session. Since ASM is used for the Oracle database, all the database files, including data files, control files, and online redo logs, are inside ASM diskgroups.
  - When creating an image for the hot backup use case, create two Symmetrix device groups — one for data files, and the other for archive log files, because the snap sessions will be activated at different times.

Note: Using device groups with TimeFinder is only one option. Alternatively a composite group or device files can be used when source and target devices span more than one Symmetrix frame.

- Create TimeFinder/Snap copy sessions using the previously created device groups
  - For creating a hot backup database image, create two TimeFinder/Snap copy sessions — one for data files, and the other for archive log files.

In the following examples, EMC used Oracle 11g release 2 Standalone Grid Infrastructure (for ASM and Oracle Clusterware), and a 11g release 2 database on Linux with TimeFinder/Snap operations. With Oracle 11g R2, the Oracle Cluster Registry (OCR) and voting disks can be placed inside ASM in diskgroups. Alternatively, other volume managers, file systems, and operating systems can be used in a similar way.

Typically, for an Oracle database using ASM, host devices are grouped into ASM diskgroups and the database files are striped inside each diskgroup. Since each diskgroup is considered a unit, the Symmetrix device groups should always include the full diskgroup.

EMC recommends that you keep an up-to-date copy of the Oracle database parameters file, ASM instance parameter file, any network configuration files, and any password files at the target host or as part of the backup set. This will save time if the source host becomes unavailable and the backup has to be used, or if the database is started at the remote site.

Note: When the quorum and OCR are stored in ASM (allowed in Oracle database 11g R2 and later), it is an EMC best practice to not include that ASM diskgroup in any storage replication operation since it contains information that is relevant to the production host and network. Therefore EMC recommends creating an additional ASM diskgroup, called ‘+GRID’. For example, this would apply for three small devices configured in ASM high redundancy. The reason for high redundancy is simply to cause Oracle to create three quorum devices. Normal redundancy will create two quorum and external redundancy will create one quorum (the decision is left to the DBA). In that way, all other ASM diskgroups can be created as before, separate from the ‘+GRID’ ASM diskgroup. If the TimeFinder target devices are to be mounted to a backup host, a newly configured ‘+GRID’ ASM diskgroup can be set up to include the Oracle clusterware and mount the rest of the replicated diskgroup into it.
Creating a restartable database image using TimeFinder/Snap

Creating a restartable database image is often used for testing, developing, or reporting applications. In this case, hot backup mode is not used and archive logs are not required. All Oracle data files, online redo logs, and control files are included in the source devices.

TimeFinder/Snap operation on the source host

All the following actions in the example below are performed on the source host.

Note: You can perform the same operations from a management or backup host with minor syntax changes to the sample scripts below.

1. Create a TimeFinder/Snap copy session that includes all the Oracle data files, redo logs, and control files.

```
symdg create ASMDB
symld -g ASMDB addall -range 00:0A
symld -g ASMDB addall dev -range 4E:57 -vdev
symsnap -g ASMDB create
```

2. Activate the copy session with the consistent activation option to create a point-in-time restartable image of the Oracle database source devices.

```
symsnap -g ASMDB activate -consistent
```

Starting Oracle Database on a target host

When Oracle Database (10g or 11g) is started at a target host from an image taken using the consistent activation option, you can restart the image of the database in the same manner as you would start it after a power failure or shutdown abort.

Note: You should not use manual (user-managed) media recovery.

All the following actions are performed on the target host.

1. **System preparations:** Verify that the target devices are accessible by the target host (in READY state). For example, if the -not_ready option was used during the session activate, change the device states to Ready. In some cases, a rescanning of the SCSI bus is required for the operating system to recognize new devices. If you use a logical volume manager (LVM), import volume groups. If you use file systems, check (fsck) the file systems, and mount them to the target host. If the database is on a file system, EMC recommends mounting the target devices in an identical directory structure to the source host and Oracle software is located at the same ORACLE_HOME for ease of use. In this example, the database resides in Oracle ASM diskgroups.

2. **Oracle preparations:** Review and update (if necessary) any Oracle parameter files, including the parameter file for the ASM instance, and Oracle network files with the correct information for the target host environment. If you made changes to database file locations, you must rename those files to point to their new location. In this example, no changes are necessary as the target host is similar to the source host.

3. Start the ASM instance as ASM system administrator (sysasm) at the target host.

```
SQL> startup;
```
4. Start the database at the target host after the ASM instance has been started.

```
SQL> startup;
```

**Creating multiple rolling point-in-time restartable database images**

TimeFinder/Snap multiple snap allows a larger number of restore points for Oracle environments by taking snaps at regular intervals. Users can restore to intervals in the past without requiring the storage consumption of multiple full copies.

The process for creating multiple point-in-time database rolling snap images is the same as creating single point-in-time snap images. In this example, we describe a use case of establishing and maintaining three point-in-time snap images of a source database. Each snap session occurs in one-hour intervals.

**Creating the first TimeFinder/Snap point-in-time database image**

All the following actions are performed on the source host.

1. Create a copy session that includes all the Oracle data files, redo logs, and control files.

```
symdg create Snap1
symld -g Snap1 addall -range 00:0A
symld -g Snap1 addall dev -range 4E:57 -vdev
symsnap -g Snap1 create
```

2. Activate the copy session with the consistent activation option to create a point-in-time restartable image of the Oracle database source devices.

```
symsnap -g Snap1 activate -consistent
```

If the database is on a file system, EMC recommends that you flush the file system just before the TimeFinder/Snap copy activation (use UNIX: `sync`, `sync`, OCFS Flush, and so on).

**Creating additional TimeFinder/Snap point-in-time database images at regular intervals**

Create and activate a second snap session a certain amount of time after you have performed the first snap session (for example, one hour). Perform all the following actions are performed on the source host.

1. Create a copy session that includes all the Oracle data files, redo logs, and control files.

```
symdg create Snap2
symld -g Snap1 addall -range 00:0A
symld -g Snap2 addall dev -range 58:58 -vdev
symsnap -g Snap2 create
```

2. Activate the copy session with the consistent activation option to create a point-in-time restartable image of the Oracle database source devices.

```
symsnap -g Snap2 activate -consistent
```
3. Create and activate a third snap session one hour after the second snap session. All the following actions are performed on the source host.

4. Create a copy session that includes all the Oracle data files, redo logs, and control files.

   ```
   symdg create Snap3
   symld -g Snap1 addall -range 00:0A
   symld -g Snap3 addall dev -range 58:58 -vdev
   symsnap -g Snap3 create
   ```

5. Activate the copy session with the consistent activation option to create a point-in-time restartable image of the Oracle database source devices.

   ```
   symsnap -g Snap3 activate -consistent
   ```

Re-creating multiple point-in-time database rolling snap images at subsequent time intervals

1. Terminate the previously created Snap1 snap session, then re-create and reactivate the copy session with the consistent activation option to create a new point-in-time restartable image of the Oracle database source devices.

   ```
   symsnap -g Snap1 terminate
   symsnap -g Snap1 create
   symsnap -g Snap1 activate -consistent
   ```

2. Terminate the previously created Snap2 snap session, then re-create and reactivate the copy session with the consistent activation option to create a new point-in-time restartable image of the Oracle database source devices as described for the Snap1 session.

   ```
   symsnap -g Snap2 terminate
   symsnap -g Snap2 create
   symsnap -g Snap2 activate -consistent
   ```

3. Repeat this terminating, re-creating, and reactivating multiple point-in-time rolling snap image process as needed for maximum protection of the production database.

   With an increase in sessions per source device for rolling snap images, and the likelihood of multiple source LUNs for a given database, using the device group methodology may become unwieldy. Alternatively, users can implement TimeFinder/Snap operations utilizing device files, which allow administrators to create discrete files that retain the mapping of source LUNs and target TimeFinder/Snap targets.

   **Mounting and renaming a TimeFinder/Snap restartable database image**

Renaming the ASM diskgroup replica

In Database 11g R2, Oracle introduced the renamedg feature to change the name of an ASM diskgroup. This feature correlates with multiple point-in-time snap copy images of the database so that you can mount them on the same server. In this use case, the snap copy images, DATA1, DATA2, and REDO ASM diskgroups, will be mounted to a secondary host installed with Oracle 11g R2 Standalone Grid.

Implementing TimeFinder Multiple Snapshots with Virtual Provisioning for Oracle 11g Databases on EMC Symmetrix VMAX

Applied Technology
Infrastructure and Database software. Subsequently, the ASM diskgroups will be renamed, the database file locations will be updated, and the database will use new ID and instance names. This procedure enables users to replicate multiple snapshot images of the source database to the same target host.

**Note:** When you use the `renamedg` command, ensure that the ASM diskgroup is dismounted.

### Initial information gathering

The status of the Oracle resources on the server will tell you whether or not a database is online, and whether ASM diskgroups are dismounted. To check the status of Oracle resources on the server, use the following `crsctl` command from the Grid Infrastructure home:

```
$ crsctl status resource
```

After executing this command, you will see a list of the status and configuration information for Oracle resources. If the ASM diskgroups are still online, dismount the diskgroups that you need to rename using the following `asmcmd` command:

```
$ asmcmd umount diskgroup_name
```

The following script confirms that all the diskgroups are dismounted:

```
$ asmcmd lsdg
```

### Rename ASM diskgroups

Use the ASM diskgroup renaming utility, `renamedg`, to rename a target ASM diskgroup:

```
renamedg dgname=<diskgroup> newdgname=<newdiskgroup>
asm_diskstring=<discoverystring>
```

```
$ renamedg dgname=data1 newdgname=dg1data1
asm_diskstring='/dev/emcpower*1' verbose=true
NOTE: No asm libraries found in the system
Parsing parameters...
Parameters in effect:
    Old DG name       : DATA1
    New DG name       : DG1DATA1
    Phases            :
        Phase 1
        Phase 2
    Discovery str     : /dev/emcpower*1
    Clean             : TRUE
    Raw only          : TRUE
renamedg operation: dgname=data1 newdgname=dg1data1 asm_diskstring=/dev/emcpower*1
verbose=true
Executing phase 1
Discovering the group
Performing discovery with string:/dev/emcpower*1
Identified disk UFS:/dev/emcpowerao1 with disk number:1 and timestamp (329266671861822464)
Identified disk UFS:/dev/emcpoweraq1 with disk number:0 and timestamp (329266671861822464)
Checking if the diskgroup is mounted
Checki...
```

Implementing TimeFinder Multiple Snapshots with Virtual Provisioning for Oracle 11g Databases on EMC Symmetrix VMAX

Applied Technology
Checking disk number: 0
Checking if diskgroup is used by CSS
Generating configuration file..
Completed phase 1
Executing phase 2
Looking for /dev/emcpowerao1
Modifying the header
Looking for /dev/emcpoweraq1
Modifying the header
Completed phase 2
Terminating kgfd context 0x2b61624ca0a0

If you need to, you can rename other ASM diskgroups in the same way. In EMC’s test environment, in addition to DATA1, diskgroups DATA2 and REDO are also part of the database ASM diskgroups, and are subject to renaming as the following:

$ renamedg dgname=data2 newdgname=dg1data2
asm_diskstring=/dev/emcpower*1' verbose=true
$ renamedg dgname=redo newdgname=dg1redo
asm_diskstring=/dev/emcpower*1' verbose=true

Mount renamed ASM diskgroups
After renaming an ASM diskgroup, you can mount the renamed diskgroups using `asmcmd` command as the following:

$ asmcmd mount diskgroup_name

In this example, we mounted the renamed DG1DATA1, DG1DATA2, and DG1REDO diskgroups

$asmcmd mount dg1data1
$asmcmd mount dg1data2
$asmcmd mount dg1redo

The following script confirms that you have mounted the diskgroups:

$ asmcmd lsdg

Alternatively, you can mount the diskgroup by connecting to ASM instance.

sqlplus "/ as sysasm"
sql> alter diskgroup DG1DATA1 mount;

Update database file locations
If you choose to rename the ASM diskgroups, you also need to update the location of database control_files and other *_file_dest prior to opening the database. In this instance, we need to modify the parameter file, and change all occurrences of ‘+DATA1’ to ‘+DG1DATA1’, ‘+DATA2’ to ‘+DG1DATA2’, and ‘+REDO’ to ‘+DG1REDO’.

First, mount the database and rename database files and redo logs locations as necessary.

SQL> startup nomount pfile='/oracle/ora11/dbs/inittpcc.ora'
ORACLE instance started.
Total System Global Area 343154688 bytes
Fixed Size 1336428 bytes
Variable Size 218106772 bytes
Database Buffers 117440512 bytes

Implementing TimeFinder Multiple Snapshots with Virtual Provisioning for Oracle 11g Databases on EMC Symmetrix VMAX
Applied Technology
Implementing TimeFinder Multiple Snapshots with Virtual Provisioning for Oracle 11g Databases on EMC Symmetrix VMAX

Redo Buffers 6270976 bytes

SQL>alter database mount;

Next, use the commands below to generate a sql script to rename diskgroup information for **datafiles, redo logs, and tempfiles**; you have to modify the queries to reflect your particular database environment.

```sql
SQL> select 'alter database rename file '''||name||''' to '''||substr(name,instr(name,'D',1,1))||'''' from V$DATAFILE;

SQL> select 'alter database rename file '''||member||''' to '''||substr(member,instr(member,'R',1,1))||'''' from V$logfile;

SQL> select 'alter database rename file '''||name||''' to '''||substr(name,instr(name,'D',1,1))||'''' from V$TEMPFILE;
```

Run the respective sql rename scripts, as generated above, to rename the datafile, redo logfile, and tempfile locations. Use the VSRECOVER_FILE view to check for any availability issues with datafiles. You should also verify the alert log to confirm that the renaming is successful. If the renaming process is successful, you can open the database.

You can then re-create spfile on the ASM diskgroup using the following steps

```bash
$cd $ORACLE_HOME/dbs
$cat inittpcc.ora
SPFILE='+DG1DATA1/tpcc/spfiletpcc.ora'
$mv inittpcc.ora inittpcc.ora.new
$ cp /tmp/param.txt $ORACLE_HOME/dbs/inittpcc.ora
$sqlplus '/ as sysdba'
SQL>create spfile='+DG1DATA1/tpcc/spfiletpcc.ora' from pfile;
exit
$mv inittpcc.ora initpcc.old

Modify inittpcc.ora.new and change diskgroup to +DG1DATA1
$cat inittpcc.ora.new
SPFILE='+DG1DATA1/tpcc/spfiletpcc.ora'
$mv inittpcc.ora.new inittpcc.ora
```

At this time the database can be started. Repeat the above steps for additional snap images of the ASM database that you would like to rename and mount on the same server.

**Change the database name**

To mount multi-snap point-in-time database images on the same server with a distinct instance and database name, use Oracle’s DBNEWID utility to change the database name for individual database images after you have renamed the ASM diskgroups and database files. The following are the steps you should follow to change the database name without changing the DBID.

First, ensure that you have a recoverable whole database backup. Ensure that the target database is mounted, but not open, and that it was shut down consistently prior to mounting.

```sql
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP MOUNT
```

Invoke the utility on the command line, specifying a valid user with the SYSDBA privilege. You must specify both the **DBNAME** and **SETNAME** parameters. This example changes the name from tpc to tpc1:

```bash
Implementing TimeFinder Multiple Snapshots with Virtual Provisioning for Oracle 11g Databases on EMC Symmetrix VMAX
Applied Technology
```
% nid TARGET=SYS/oracle@tpcc DBNAME=tpcc1 SETNAME=YES

DBNEWID performs validations in the headers of the control files (not the datafiles) before attempting I/O to the files. If validation is successful, then DBNEWID prompts for confirmation, changes the database name in the control files, and exits. After DBNEWID completes successfully, the database is left mounted but is not yet usable.

DBNEWID: Release 11.2.0.1.0
(c) Copyright 2009 Oracle Corporation. All rights reserved.

Connected to database TPCC (DBID=3942196782)

Control Files in database:
+DG1DATA1/control_001
+DG2DATA1/control_002

Change database name of database TPCC to TPCC1? (Y/[N]) => Y

Proceeding with operation

Database name changed from TPCC to TPCC1 - database needs to be shutdown.
Modify parameter file and generate a new password file before restarting.

DBNEWID - Successfully changed database name

If validation is not successful, then DBNEWID terminates and leaves the target database intact. You can open the database, fix the error, and then either resume the DBNEWID operation or continue using the database without changing the database name.

Finally, shut down the database.

SQL> SHUTDOWN IMMEDIATE

Set the DB_NAME initialization parameter in the initialization parameter file to the new database name.

Create a new password file.

Start up the database and resume normal use.

Repeat the steps in this section as necessary to rename a database name of additional point-in-time database snap consistent copy images. Now, you can mount multiple snap copy images of the database on the same host with distinct database names.

Creating a database hot backup image using TimeFinder/Snap

The process of activating a snap session while Oracle is open for read/write operations and is in hot backup mode is performed when the snap target is required as a valid backup image. Since backup operations are mostly read, the backup application itself does not add to the save device storage requirement. Optionally, the target device image can be restored to other Symmetrix devices for further use.

TimeFinder/Snap operation on the source host

Perform the following actions on the source host.
1. Create a snap session that includes all the Oracle data files. In this case, a second snap session is created that includes the archive redo logs and backup control file that you need for the recovery process. Optionally, copy the backup control file to the target host over the network.

```
symdg create ASMDB
symld -g ASMDB addall -range 04:0A
symld -g ASMDB addall dev -range 51:57 -vdev
symdg create Archives
symld -g Archives add dev 0B
symld -g Archives add dev 58 -vdev
symsnap -g ASMDB create
symsnap -g Archives create
```

2. Verify that Oracle archive log mode is enabled and if not, follow Oracle procedures to enable it before proceeding.

```
SQL> archive log list; or
SQL> select LOG_MODE from V$DATABASE;
```

The database is placed in hot backup mode.

```
SQL> alter database begin backup;
```

3. Activate the ASMDB copy session to create a point-in-time image of the Oracle data files’ source devices.

```
symsnap -g ASMDB activate -consistent
```

The database is taken out of hot backup mode.

```
SQL> alter database end backup;
```

4. Archive the current online redo log.

```
SQL> alter system archive log current;
```

5. Save a backup control file.

```
SQL> alter database backup controlfile to
   '/ora11/dbs/snap/arch/bckcontrol.ctl';
```

In this example, the backup control file is saved under the archive log file directory. Therefore, it is available with the archives to the target host. Alternative methods can be used to send the backup control file to the target host.

6. Activate the archive logs copy session to make a point-in-time image of the archive redo logs and to make the backup control file available. (The backup control file can be sent to the target host by means of ftp; archive logs can use an additional archive log destination.)

```
symsnap -g Archives activate -consistent
```

Starting an Oracle database on a target host after snap activation with hot backup

An Oracle database image that was created while Oracle was in hot backup mode requires media recovery before it can be used. Note that once the target database is opened for read/write, it becomes no longer valid as a backup of the production database. In addition, a recovery process always requires careful planning and preparation according to the circumstances and host environment where the recovery takes place. In this example, the environment on the target host is assumed to be identical to the source host. For more
information on Oracle media recovery, refer to the Oracle document *User-Managed Backup and Recovery Guide*.

All the following actions are performed on the target host.

1. **System preparations**: Verify that the target devices are accessible by the target host. For example, if the `-not_ready` option was used during the session activate, then change device states to Ready. In some cases, you will need to rescan the SCSI bus for the operating system to recognize new devices. If your system uses LVM, import volume groups. However, if you use file systems, check *(fsck)* the file systems, and mount them to the target host. If the database is on the file system, EMC recommends mounting the target devices in an identical directory structure to the source host and locating the Oracle software at the same `ORACLE_HOME` for ease of use. In this example, the database resides in Oracle ASM diskgroups.

2. **Oracle preparations**: Review and update (if necessary) any Oracle parameter files, including the parameter file for the ASM instance, and Oracle network files with the correct information for the target host environment. If you made changes to database file locations, you must rename those files to their new location. In this example, no changes are necessary as the target host is similar to the source host. Point the database to use the backup control file.

3. Start the ASM instance at the target host.
   
   ```sql
   SQL> startup;
   ```

4. Recover the database on the target host using the archive logs. In this case, the recovery is performed using the AUTOMATIC UNTIL CANCEL option. Refer to Oracle documentation for other types of recovery.
   
   ```sql
   SQL> startup mount;
   SQL> recover automatic database until cancel using backup controlfile;  
   CANCEL
   ```

5. Open the database. In this case, the database is open for read/write with the resetlogs option. (Opening the database in Read-Only mode will not invalidate the database as a backup source to the production/source database.)

   ```sql
   SQL> alter database open resetlogs;
   ```

### Incremental restore to source database devices

When you need to update the source devices with information from the target virtual devices, an incremental restore (only changed data tracks are copied) will take place.

### TimeFinder/Snap restoring operation on the source host

To perform a restore operation on the source host, follow these steps:

1. If any application on the target host is accessing the target devices, stop the application. If file systems are mounted on the target devices, unmount them. If LVM is used, deport the volume groups such that the target devices are not used anymore and are ready to be restored to the source devices.

2. Shut down the Oracle database and ASM instance on the source host.

3. Restore the copy session.

   ```shell
   symssnap -g ASMDB restore
   ```
**Database recovery from a restartable snap**

This use case illustrates fast database recovery by using the most recent consistent (restartable) replica and by applying logs to it. The database was not placed in hot backup mode before the snap activation.

Oracle supports various database recovery scenarios based on dependent write consistent storage replicas created using SRDF and/or TimeFinder. Oracle support is covered in metalink note ID 604683.1. The purpose of this use case is not to replace normal backup strategies, such as nightly backups based on hot backup mode. Instead, it offers a complementary use case, such as when RTO requirements are very strict. It could be a compelling solution to run the database in archivelog mode, and perform periodic snapshots without placing the database in hot backup mode. If recovery is required, the last snapshot is restored and in parallel the limited transactions since that snapshot was taken are restored, creating a fast database recovery solution.

Consider this scenario. The database is in archive log mode and periodic TimeFinder consistent snaps are created that include only the data. At some point a database recovery is required based on the last point-in-time snap copy image with consistent activation. In this use case, snap copy session ASMDATA is designated for taking periodic consistent snaps that include only the data, which resides in ASM diskgroups DATA1 and DATA2. The ASMDATA snap copy session consists of devices that include ASM diskgroups DATA1 and DATA2.

The following are the high-level steps to create a recoverable database image using a restartable snap:

1. Shut down the production database and ASM instances.
2. Restore the most recent TimeFinder/Snap copy image ASMDATA (split afterwards) that includes all the Oracle data files, which reside in ASM diskgroups DATA1 and DATA2.
3. Start the ASM instance.
4. Mount the database.
5. Only in the case of incomplete recovery is it necessary to perform a scan to update data file headers.
6. Perform database full or incomplete recovery (possibly while the TimeFinder background restore is still taking place).

Ensure that you use only DATA1 and DATA2 device groups.

Detailed operations are as follows:

1. Shut down any production database and ASM instances (if still running).

   ```
   $ export ORACLE_SID=TPCC
   $ sqlplus "/ as sysdba"
   SQL> shutdown abort
   
   $ export ORACLE_SID=+ASM
   $ sqlplus "/ as sysdba"
   SQL> shutdown abort
   ```

2. Restore the most recent ASMDATA TimeFinder snap copy image.

   ```
   symsnap -g ASMDATA restore
   ```

3. Start the ASM instance.

   ```
   $ export ORACLE_SID=+ASM
   $ sqlplus "/ as sysdba"
   SQL> startup
   ```

4. Mount the database.

   ```
   $ export ORACLE_SID=CLONE_DB
   $ sqlplus "/ as sysdba"
   ```

---

Implementing TimeFinder Multiple Snapshots with Virtual Provisioning for Oracle 11g Databases on EMC Symmetrix VMAX

Applied Technology
5. If incomplete recovery is required, perform `dbms_backup_restore.scandatafile()` as shown in the example below (data files can be scanned in parallel to reduce the overall time). Full recovery (when redo logs are available) doesn’t require the scan operation.

6. Perform database recovery based on one of the following options.

**Full (complete) database recovery**
When all online redo logs and archive logs are available, it is possible to perform a full media recovery of the Oracle database to achieve a no data loss of committed transactions.

```
SQL> recover automatic database;
SQL> alter database open;
```

Note: It might be necessary to point the location of the online redo logs or archive logs if the recovery process did not locate them automatically (common in RAC implementations with multiple online or archive logs locations). The goal is to apply any necessary archive logs as well as the online logs fully.

---

**Point-in-time database recovery**
When a full media recovery is not desirable, or when some archives or online logs are missing, an incomplete recovery can be performed. When performing incomplete recovery enough logs need to be applied to pass the maximum point of data file fuzziness so they are all consistent. After passing that point, additional archive can potentially be applied. The following is a sample script (based on the Oracle metalink note mentioned previously) that can help identify the minimum SCN required to open the database and update the data file headers accordingly.

Scan the datafile script:

```sql
spool scandatafile.out
set serveroutput on
declare
    scn number(12) := 0;
    scnm max number(12) := 0;
begin
    for f in (select * from v$datafile) loop
        scn := dbms_backup_restore.scandatafile(f.file#);
        dbms_output.put_line('File ' || f.file# || ' absolute fuzzy scn = ' || scn);
        if scn > scnm max then scnm ax := scn; end if;
    end loop;
    dbms_output.put_line('Minimum PITR SCN = ' || scnm ax);
end;
```

Sample output generated by the scan data script:

```
SQL> @./scandata.sql
File 1 absolute fuzzy scn = 27088040
File 2 absolute fuzzy scn = 27144475
File 3 absolute fuzzy scn = 27164171
... File 22 absolute fuzzy scn = 0
Minimum PITR SCN = 27164171
```
Perform incomplete database recovery (sample commands):

```
SQL> alter database recover database until change 27164171;
SQL> alter database open resetlogs;
```

**Terminating TimeFinder/Snap sessions**

When you restore a TimeFinder/Snap session, that session remains in a restored state. It is necessary to terminate any restored session prior to discarding the TimeFinder/Snap session itself. It is also necessary to terminate a restored session prior to restoring the state from a different session. In this latter case, it is possible to then revert to any other valid session from the 128 sessions in effect with TimeFinder/Snap multi-virtual snap. You will also be able to revert back to the originally restored session, if changes were subsequently made to that restored state, and it was necessary to undo those changes.

Terminating a restored session does not discard the TimeFinder/Snap session, it simply disconnects the snapshot restore session from the standard devices. The following example command can be used to terminate a restored session:

```
symsnap -g ASMDB terminate -restored
```

**Terminating and discarding a TimeFinder/Snap session**

When a TimeFinder/Snap session is fully terminated, or when all device pointers represented in the VDEVs are deleted, those tracks allocated in the save pool that are no longer required by any TimeFinder/Snap session are freed for subsequent use. To terminate a TimeFinder/Snap session, execute the following command syntax:

```
symsnap -g ASMDB terminate
```

**Conclusion**

The new TimeFinder/Snap multi-virtual snapshot functionality delivers support for creating up to 128 TimeFinder/Snap sessions against a single source, and is fully compatible with Oracle 10g and 11g environments. This new functionality starting with the Enginuity 5772.83.75 service release provides significant improvements and enhances operational aspects of Oracle database implementations.

Providing full compatibility with Oracle backup and restore functionality, the increased TimeFinder/Snap enhancements may easily be integrated with existing environments, delivering robust customer solutions. Organizations are able to use the enhanced functionality to easily revert to earlier sessions to fulfill business needs. Additionally, these TimeFinder/Snap replicas may be surfaced to secondary servers, as required, to provide access to the point-in-time restartable or recoverable database states.

**References**

- White Paper: EMC Symmetrix VMAX Using EMC SRDF/TimeFinder and Oracle Database 10g/11g - Applied Technology
- White Paper: Implementing Virtual Provisioning on EMC Symmetrix VMAX with Oracle Database 10g and 11g - Applied Technology
- White Paper: Reducing Backup Window and Recovery Time with Oracle 11g RMAN and EMC TimeFinder/Clone - Applied Technology
- EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide
- Oracle documentation library at http://www.oracle.com
Appendix: Test configuration
Details of the test configuration are follows:

Primary server
- Red Hat Linux running 2.6.18-128.el5
- EMC Solutions Enabler 7.1
- EMC PowerPath® 5.3 SP 1

Backup or target server
- Red Hat Linux running 2.6.18-128.el5
- EMC Solutions Enabler 7.1
- EMC PowerPath 5.3 SP 1

EMC Symmetrix VMAX system
- Enginuity 5874

Oracle Database
- Version 11.2.0.1.0

Oracle Database layout

<table>
<thead>
<tr>
<th>Type</th>
<th>Symmetrix device number</th>
<th>Mount point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Files</td>
<td>004-00A</td>
<td>+DATA</td>
</tr>
<tr>
<td>Redo Logs</td>
<td>000-003</td>
<td>+REDO</td>
</tr>
<tr>
<td>Archive Logs</td>
<td>000B</td>
<td>/orall/dbs/snap/arch/</td>
</tr>
<tr>
<td>Control Files</td>
<td></td>
<td>+DATA/ctl1.ctl</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+DATA/ctl2.ctl</td>
</tr>
</tbody>
</table>