Oracle Maximum
Availability Architecture

An Oracle White Paper
Dec 2011

Database Cloning using Oracle Sun ZFS Storage Appliance and Oracle Data Guard

ORACLE

Introduction

Database administrators face the challenge of efficiently duplicating their large mission-critical databases to support continuous demands for application development and subsequent testing of the code. This challenge is compounded by the fact that multiple clones are often required for each production database in order to support the many development and test activities associated with large production systems. It is also important that the cloning process have zero impact to the production database. While the basic method of creating a clone is to restore a database from a recent backup on to a different database server, it is quickly obvious how such an approach is inefficient and highly time-consuming. Without an efficient solution for cloning production databases, enterprises are saddled with substantial administrative burden that diverts attention away from more time-critical support functions along with increased storage consumption and high cost.

Many Oracle customers have already deployed Oracle Data Guard to provide high availability and disaster protection for their mission critical Oracle Databases. The addition of Oracle's Sun ZFS Storage Appliance (also referred to as *the appliance* in this document) to Oracle's solution portfolio offers Data Guard users a unique opportunity to address their cloning requirements. The appliance can easily be added to an existing Data Guard configuration and provide an industry-leading, low-cost, time and space efficient solution to enterprise requirements for snapshots and clones.

About This Document

This document describes how the Oracle Data Guard feature is deployed in conjunction with the *snapshot* and *cloning* features of the Sun ZFS Storage Appliance, enabling easy and efficient database cloning of a standby database. This document also includes Oracle-validated best practices and scripting to automate the database cloning operation.

The solution described in this document supports the following operating system and database versions.

Table 1: Supported Versions

Parameter	Version
Operating system	Oracle Solaris 10 operating system Sparc, X86 RedHat Linux 4.x, 5,x (and higher) Oracle Enterprise Linux 4.x, 5.x (and higher)
Oracle database	10.2.0.1 (and higher) Single instance standby database Oracle Data Guard, Active Data Guard **Oracle 11.2.0.3 and above to use with Oracle Exadata Hybrid Columnar Compression (HCC)**
Storage	Oracle Sun ZFS Storage Appliance
Protocol	NFSv3, NFSv4
Standby and cloned database	Sharing the same host Sharing the same host as that of another standby host Standby and cloned instances on separate hosts

A new standby database dedicated to supporting snapshot and clone operations is created on the appliance using the NFS protocol (Data Guard supports up to 30 standby databases in a single Data Guard configuration). The Oracle instance managing the standby database may be deployed on the same server or on a different server than the existing standby database(s). The storage used for the existing primary and standby databases can be from any vendor. In this fashion, the Sun ZFS Storage Appliance can be quickly deployed with zero disruption to existing Oracle Data Guard environments.

When the standby site is established on the appliance, a snapshot of the standby database's file systems is performed, yielding a copy of the data at that point in time. Then, the snapshot data is cloned and the file systems are again accessible for reads and writes. The cloned standby database is then activated and converted to a database which is then used for test, development, and QA purposes.

Audience

This document is written for Oracle database administrators, storage/system administrators, and technical sales personnel. It is assumed that the reader is familiar with the Oracle Data Guard feature. For a detailed file system creation procedure on the Sun ZFS Storage Appliance, refer to the documents listed in the section.

About the Sun ZFS Storage Appliance

Architecture Overview

This topic provides an overview of the Sun ZFS storage appliance. For features, functions, and details regarding various platform offerings, refer to:

http://www.oracle.com/us/products/servers-storage/storage/unified-storage/index.html

Sun ZFS Storage Appliance combines ease of management into a single appliance, multiple protocol connectivity, and data services for business continuity. The appliance supports NFS, Common Internet File System (CIFS), Internet Small Computer System Interface (iSCSI), InfiniBand (IB), and Fibre Channel (FC). Available as single head or a clustered head, for high availability, the appliance also supports Network Data Management Protocol (NDMP) for backup and restore purposes. Oracle Solaris operating system with Oracle Solaris ZFS is the core of the appliance with the ZFS file system powering all the data storage, management, and data services. You access these services using an intuitive user interface or CLI modes.

The storage architecture also utilizes hybrid storage pool (HSP), where DRAM, Flash, and physical disks are seamlessly integrated for efficient data placement. Based on the user IO request and pattern, the data movement between these tiers are automatically handled by the appliance. The storage also includes a powerful performance monitoring tool called *Analytics,* which provides details about the performance of various components including the network, storage, file systems, client access, and so forth. There are plenty of drill-down options available. For example, Analytics can show which clients are accessing which file systems and files, latency, size of transfer, and so on.

The storage also offers a variety of RAID protections to balance capacity, protection, and performance requirements of the applications.

Platforms

The Sun ZFS Storage Appliance is available in three models to meet customer requirements for price, performance, capacity, and protection capabilities. The mid- to high-end platforms offer up to 2TB of read cache which enables the appliances to have a response time typically in low single digit milliseconds. The write Flash on all four platforms provide response time for the synchronous writes with less than a millisecond.

The new Sun ZFS Storage Appliance platforms offer faster CPUs, bigger flash cache, larger storage capacity, better throughput, and broader bandwidth to meet the storage requirements of mission-critical applications.

Table 2 provides configuration details for each model.

Table 2: Sun ZFS Storage Appliance Models

Platform	Storage Capacity	Processor	Memory (DRAM)	Write Optimized SSD	Read Optimized SSD	Cluster Option
Sun ZFS Storage 7120	Up to 60 x 2TB SAS Disks [120TB]	1 x Quad Core Intel Westmere EP E5620 @ 2.4GHz	Up to 36GB	Up to 96GB	N/A	N
Sun ZFS Storage 7320 (details are per controller)	Up to 96 x 2TB SAS Disks [192TB]	2 x Quad Core Intel Westmere EP E5620 @ 2.4GHz	Up to 72GB	Up to 16 x 18GB	Up to 4 x 512GB	Y
Sun ZFS Storage 7420 (details are per controller)	Up to 576 x 2TB SAS Disks [1.1PB]	4 x 6C Intel Nehalem EX E7530 @ 1.86GHz [or] 4 x 8C Intel Nehalem EX X7550 @ 2GHz	Up to 512GB	Up to 96 x 18GB	Up to 4 x 512GB	Y

Concepts and Terminology

Storage Pool

The storage pool (similar to a volume group) is created over a set of physical disks. The file systems are then created over the storage pool. Users can create one or more storage pools over the available physical disks and assign the flash drives for the pool when configuring the storage pool.

Project

All file systems and LUNs are grouped into projects. A project can also considered as a *consistency group*. A project defines a common administrative control point for managing shares. All shares within a project can share common settings, and quotas can be enforced at the project level in addition to the share level. Projects can also be used solely for grouping logically-related shares together, so their common attributes (such as accumulated space) can be accessed from a single point.

Shares

Shares are file systems and LUNs that are exported over supported data protocols to clients of the appliance. File systems export a file-based hierarchy and can be accessed over CIFS, NFS, HTTP/WebDav, FTP, and InfiniBand. LUNs export block-based volumes and can be accessed over iSCSI and Fibre Channel protocols.

The *project/share* is a unique identifier for a share within a pool. Multiple projects can contain shares with the same name, but a single project cannot contain shares with the same name. A single project can contain both file systems and LUNs, and they share the same namespace.

Data Services

A number of data services are bundled with the appliance and comes with a license-free model. The following are some of the key data services relevant to the database cloning procedure.

Snapshots

Sun ZFS Storage Appliance supports unlimited snapshot capability. A snapshot is a read-only, point-in-time copy of a file system. It is instantaneously created and no space is allocated initially. Blocks are allocated as changes are made to the base file system (copy-on-write). The snapshots are either initiated manually or can be automated by scheduling at specific intervals. The snapshot data can be directly accessed for any backup purposes.

Any reads to the snapshot blocks are served by the base file system's block. When changes happen to the base file system, the older block is now referenced by the snapshot and the new changed block is referenced by the file system.

Project snapshots are the equivalent of performing snapshots for all the shares within the project.

Clones

The appliance supports unlimited number of clones. A clone is an instantaneously created read-writable copy of a snapshot. One or more clones are created from a single snapshot. These clones are presented to users as a normal file system(s). All the regular operations are allowed on

clones including taking a snapshot from the clone. Clones are typically used in test, development, QA, and backup environments.

Similar to snapshots, when the clone is created, no space is allocated. The reads to the clone are served by the base file system's blocks. The changed blocks are allocated only when the blocks are changed in the clone. Since space is shared between snapshots and clones, and since a snapshot has multiple clones, a snapshot cannot be destroyed without also destroying any active clones.

From a client's perspective, the clone file systems are shown as though they are an independent file system. No special requirement for accessing the clones is needed.

Snapshot rollback

Snapshot rollback is the process to bring the base file system to the point in time when the snapshot is taken. The rollback process discards all the changes that happened to the base file system from the time of the snapshot to the time of rollback. This removes the need for a data restore process.

Refer to the following documentation URL for a detailed explanation regarding all the data services available with Sun ZFS Storage Appliance:

http://wikis.sun.com/display/FishWorks/Documentation

Oracle Database Cloning Architecture and Procedure

Strategy Overview

The cloning of the standby database is performed using the snapshot and clone features of the Sun ZFS Storage Appliance. General procedures are given below.

Set Up a new Standby Database on Sun ZFS Storage Appliance

Note: This is a one-time set-up process.

- From the appliance, set up project and file systems for storing and serving the standby database.
- Mount the file systems in the database server.
- Using Oracle Recovery Manager (RMAN), duplicate the primary production database for the standby database.
- Set up Oracle Data Guard or Active Data Guard and establish real-time replication between the primary and new standby sites.
- Enable the managed recovery procedure. If you are using Oracle versions 11*g*R2 (or higher), you can also establish *cascaded standby*.

Database Cloning procedure

- Stop the managed recovery to place the standby database in a consistent state.
- Using the Sun ZFS Storage Appliance's snapshot feature, take snapshot of the project that stores the standby database.
- Resume the managed recovery of the standby database.
- From the snapshot taken in step # 2, clone the file systems in the Sun ZFS Storage Appliance.
- Mount the NFS cloned file systems in the database server.
- Convert the cloned standby database into a read/writable database for test, development, and QA.

Figure 1 shows the relationship between file systems, snapshots, and clones. There is a one-to-many relationship between the file system and snapshots, and between snapshots and clones.

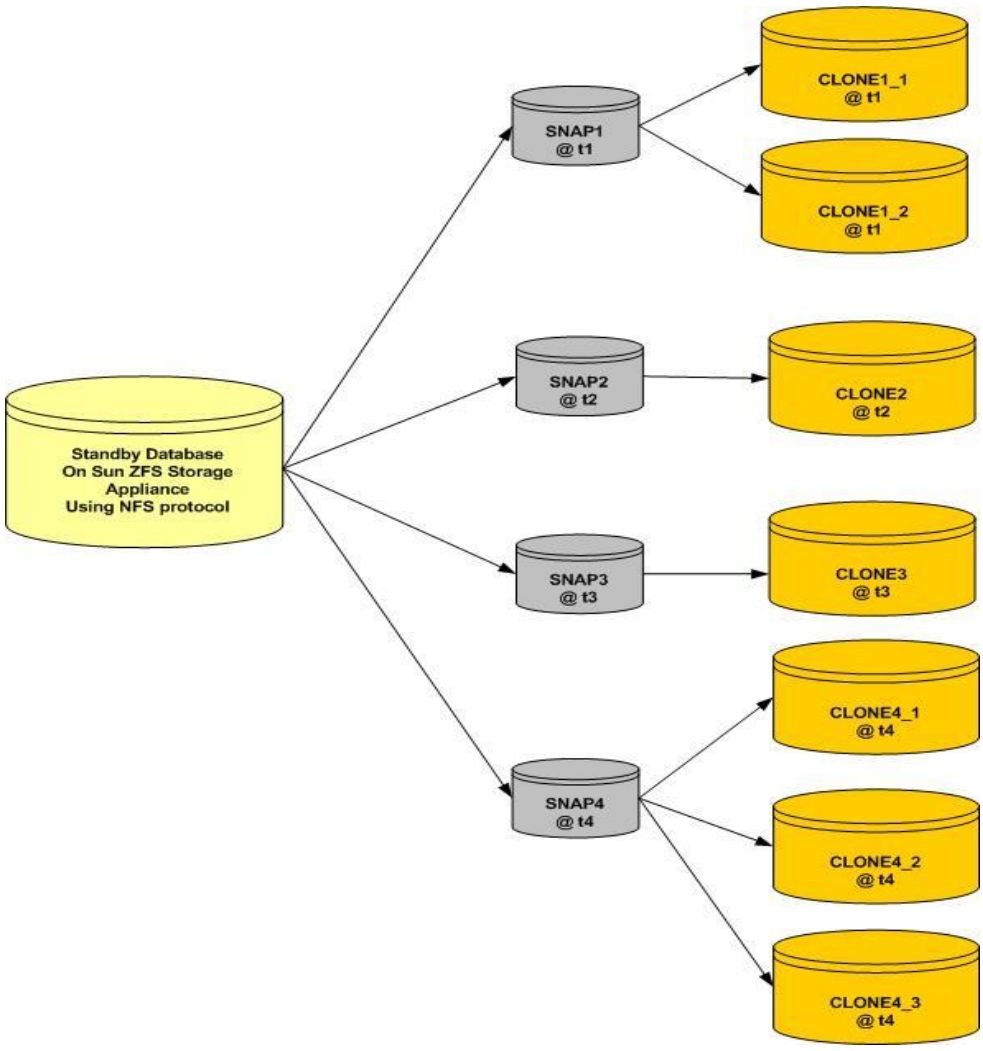


Figure 1: Snapshot and clone relationship with the base file system

Host and Storage Configuration

This section provides the configuration details used in the lab for testing the database cloning procedure.

Table 3: Primary Site Configuration

Parameter	Value
Host name	aie-6300b
System configuration	Sun Blade T6300 / UltraSPARC-T1 32 core CPU / 16GB
Operating System	Solaris 10 U 8 [Sparc]
Connectivity to storage	FC – 500GB LUNS
Oracle database	11.2.0.1 / Single instance / ASM
Oracle Instance Name	PRODDB
initPRODDB.ora {only that is modified for the new standby }	log_archive_dest_3= 'service=PRODDB_DR async db_unique_name=PRODDB_DR valid_for=(primary_role,online_logfile)' log_archive_config= 'dg_config=(PRODDB,PRODDB_SBY,PRODDB_DR)' log_archive_dest_state_2= ENABLE
ASM Diskgroup	+PRODDG
Data, Redo, Control files	+PRODDG/data/* +PRODDG/logs/*

Table 4: Standby Host Configuration for NFS Standby Database

Parameter	Value	
Host name	aie-6300c	
System configuration	Sun Blade T6300 / UltraSPARC-T1 32 core CPU / 16GB	
Operating System	Solaris 10 U 8 [Sparc]	
Connectivity to storage	10Gb Ethernet	
Oracle database	11.2.0.1 / Single instance / NFS	
Mount points	**CLIENT** /oradata/stby/data /oradata/stby/logs /oradata/stby/archive /oradata/stby/alerts	**Sun ZFS Storage Appliance** /export/standby_db/data /export/standby_db/logs /export/standby_db/archive /export/standby_db/alerts
NFS mount options	rw,bg,hard,nointr,rsize=32768,wsize=32768,proto=tcp,vers=3,actimeo=0	
ORACLE_SID	PRODDB_DR	
ORACLE_HOME	/oracle/products/11.2.0/db [local to the host]	
initPRODDB_DR.ora	*.control_files='/oradata/stby/logs/control0.ctl','/oradata/stby/logs/control1.ctl' *.db_block_size=8192 *.db_cache_size=4096M *.db_file_multiblock_read_count=128 ***.db_file_name_convert='+PRODDG/DATA','/oradata/stby/data'** *.db_files=200 *.db_keep_cache_size=256M *.db_name='PRODDB' *.db_recycle_cache_size=256M ***.db_unique_name='PRODDB_DR'** *.diagnostic_dest='/oradata/stby/alerts'	

Parameter	Value
	*.fal_client='PRODDB_DR' *.fal_server='PRODDB' *.file systemio_options='SETALL' *.instance_name='PRODDB_DR' *.java_pool_size=16M *.large_pool_size=16M ***.log_archive_dest_2='service=PRODDB async db_unique_name=PRODDB valid_for=(primary_role,online_logfile)'** ***.log_archive_dest_state_2=DEFER** *.log_buffer=104857600 *.log_checkpoint_interval=10000 *.log_checkpoint_timeout=1800 *.log_file_name_convert='+PRODDG/LOGS','/oradata/stby/logs' *.nls_date_format='MM/DD/YYYY HH24:MI:SS' *.processes=400 ***.service_names='PRODDB_DR'** *.shared_pool_size=256M *.log_archive_dest_1='location=/oradata/stby/archive_1' *.undo_management='auto' PRODDB_DR.undo_tablespace='UNDOTBS1' ***.log_archive_config='dg_config=(PRODDB,PRODDB_DR)'**
Listener.ora	SID_LIST_LISTENER = (SID_LIST = (SID_DESC = (GLOBAL_DBNAME = PRODDB_DR) (ORACLE_HOME = /oracle/products/11.2.0/db) (SID_NAME = PRODDB_DR)))
Tnsnames.ora	PRODDB_DR = (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = aie-6300c)(PORT = 1521))) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = PRODDB_DR) (ORACLE_SID=PRODDB_DR))) PRODDB = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = aie-6300b)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (ORACLE_SID = PRODDB) (SERVICE_NAME = PRODDB)))

Table 5: Cloned Database Host Configuration

Parameter	**Value**
Host name	aie-6300c [Same host as that of standby database] aie-2000d [Different host than that of standby database]
System configuration	8GB / CPU
Operating System	Solaris 10 U 8 [Sparc]
Connectivity to storage	10Gb and 1Gb Ethernet

Parameter	Value	
Oracle database	11.2.0.1 / Single instance / NFS	
Clone Name	CLONE1	
Clone based on the snapshot	SNAP1	
Mount points	**CLIENT** /oradata/CLONE1/data /oradata/CLONE1/logs /oradata/CLONE1/archive /oradata/CLONE1/alerts	**Sun ZFS Storage Appliance** /export/CLONE1/data /export/CLONE1/logs /export/CLONE1/archive /export/CLONE1/alerts
Reference to Standby DB's SRL	/oradata/stby/logs - If on same host as standby, then it is already mounted. - If on different host, then a soft link of /oradata/stby/logs pointing to /oradata/CLONE1/logs would suffice.	
NFS mount options	rw,bg,hard,nointr,rsize=32768,wsize=32768,proto=tcp,vers=3,actimeo=0	
ORACLE_SID	CLONE1	
ORACLE_HOME	/oracle/products/11.2.0/db [local to the host]	
initCLONE1.ora	**control_files='/oradata/CLONE1/logs/control0.ctl','/oradata/CLONE1/logs/control1.ctl'** db_block_size=8192 db_cache_size=4096M db_file_multiblock_read_count=128 l**og_file_name_convert='+PRODDG/','/oradata/CLONE1/','/stby/','/CLONE1/'** **db_file_name_convert='+PRODDG/','/oradata/CLONE1/','/stby/','/CLONE1/'** db_files=200 db_keep_cache_size=256M **db_name='PRODDB'** db_recycle_cache_size=256M **db_unique_name='CLONE1'** diagnostic_dest='/oradata/CLONE1/alerts' file systemio_options='SETALL' **instance_name='CLONE1'** java_pool_size=16M large_pool_size=16M log_archive_dest_state_2=DEFER log_buffer=104857600 log_checkpoint_interval=10000 log_checkpoint_timeout=1800 processes=400 service_names='CLONE1' shared_pool_size=256M log_archive_dest_1='location=/oradata/CLONE1/archive' undo_management='auto' CLONE1.undo_tablespace='UNDOTBS1'	
Listner.ora	SID_LIST_LISTENER = (SID_LIST = (SID_DESC = (GLOBAL_DBNAME = CLONE1) (ORACLE_HOME = /oracle/products/11.2.0/db) (SID_NAME = CLONE1)))	
Tnsnames.ora	CLONE1 = (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = aie-6300c)(PORT = 1521))) (CONNECT_DATA =	

Parameter	Value
	(SERVER = DEDICATED) (SERVICE_NAME = CLONE1) (ORACLE_SID=CLONE1)))

Table 6: Sun ZFS Storage Appliance Configuration

Parameter	Value	
Host name	aie-7420b	
System configuration	Sun ZFS Storage 7420 / 128GB Memory	
Connectivity	10Gb Ethernet for Data 1Gb for management	
Standby Database Project	**PROJECT NAME : PRODDB_DR** /export/standby_db/data /export/standby_db/logs /export/standby_db/archive /export/standby_db/alerts	**RECORD SIZE** 8KB 128KB 128KB 128KB
Cloned Database Project	**PROJECT NAME : CLONE1** /export/CLONE1/data /export/CLONE1/logs /export/CLONE1/archive /export/CLONE1/alerts	

Table 7: Switch Configuration

Parameter	Value
Network Switch (For storage, management)	Extreme X350-48t, 48 ports Cisco WS-X6708-10G-3C, 8 ports
Storage Switch (For FC)	Brocade 4100 32 ports full fabric 4GB SWL SFPs Fabric OS: v6.1.0

Architecture For Database Cloning

Using the hardware configuration mentioned in the previous section, Figure 2 illustrates the configuration and connectivity details for the Oracle database cloning procedure.

- Primary production database instance (PRODDB) runs on Sun Blade 6000 disk module using Sun Storage 6780 Fibre Channel (FC) array storage.
- Standby database instance (PRODDB_SBY) based on FC runs on another Sun Blade 6000 disk module with the data stored on a StorageTek 6140 array.

- NFS-based standby database instance (PRODDB_DR) configured to run on the same Sun Blade 6000 disk module with data stored on Sun ZFS Storage Appliance.
- Snapshot SNAP1 of the standby database files @ t1 is cloned into CLONE1, and SNAP2 @ t2 is cloned into CLONE2.
- CLONE1 database instance is accessed from the same Sun Blade 6000 disk module as that of the standby database.
- CLONE2 database instance is accessed from a Sun Fire T2000 server.

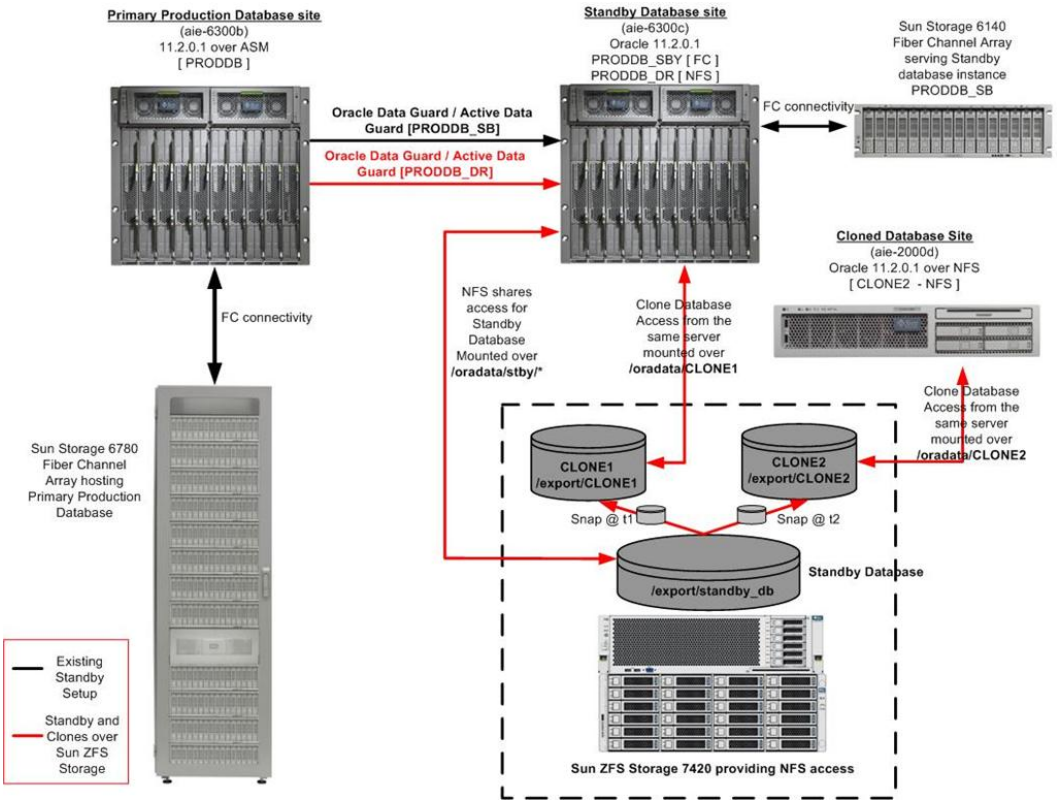


Figure 2: Architecture diagram for database cloning

Configuring a physical standby database

This section provides an overview for one of the methods used in creating a physical standby database on different storage with a different protocol. In this architecture, the primary database is stored over FC protocol and the new standby database will be created over NFS.

Note: This is a one-time setup procedure.

Set up the Primary Site Oracle Data Guard

The primary database is hosted on a high-performance Fiber Channel-based storage (such as the Sun Storage 6780 array) using Oracle ASM. The database could be an Oracle RAC or a single instance. Make these changes at the primary site's initialization parameters:

1. Add a new *log_archive_dest_#* to point to the new standby site.
2. Add the new standby site's instance to *log_archive_config*.
3. Set the *log_archive_dest_state_#* to DEFER until the standby site is established.
4. Then change the DEFER status to ENABLE.

Set up the Standby Database Over the Sun ZFS Storage Appliance

NFS protocol is used for storing the standby database's data files. The Oracle RMAN is used for database duplication. The conversion of the data file location and redo log locations are handled via initialization parameters *db_file_name_convert* and *log_file_name_convert* parameters.

1. Create file systems on the Sun ZFS Storage Appliance for storing the standby database physical files. Refer to the Deploying Oracle Databases in Sun Storage 7000 documentation listed in .

Project : PRODDB_DR

Appliance mount point	Database server mount point	Description
/export/standby_db/data	/oradata/stby/data	Stores all the data files
/export/standby_db/logs	/oradata/stby/logs	Stores the control files, online redo logs, and standby redo logs (SRL)
/export/standby_db/archive	/oradata/stby/archive	Stores archived redo logs which are shipped
/export/standby_db/alerts	/oradata/stby/alerts	Stores diagnostic details [Optional]. This can also be stored in the database server.

2. Mount these file systems at the standby host *aie-6300c*, using a script or setting in */etc/vfstab* (for the Solaris OS) o*r /etc/fstab* (for Linux)
3. Using Oracle RMAN, duplicate the database:

```
$ rman
RMAN> connect target  sys/oracle@PRODDB;
connect auxiliary sys/oracle@PRODDB_DR;
run {
 allocate channel p1 type disk;
 allocate channel p2 type disk;
 allocate auxiliary channel s1 type disk;
 allocate auxiliary channel s2 type disk;
 duplicate target database for standby from active database
 nofilenamecheck;
}
```

Result: Step 3 triggers the standby database creation directly from the production database. The standby redo logs are also populated.

```
SQL > select member from v$logfile;

MEMBER
-------------------------------------
/oradata/stby/logs/redo00-a.log
/oradata/stby/logs/redo01-a.log
/oradata/stby/logs/redo02-a.log
/oradata/stby/logs/redo03-a.log
/oradata/stby/logs/stby-1.log
/oradata/stby/logs/stby-2.log
/oradata/stby/logs/stby-3.log

7 rows selected.

SQL >select name from v$datafile;

NAME
---------------------------------
/oradata/stby/data/system00.dbf
/oradata/stby/data/sysaux00.dbf
/oradata/stby/data/undo00_a.dbf
/oradata/stby/data/undo01_a.dbf
/oradata/stby/data/data00.dbf
/oradata/stby/data/data01.dbf
/oradata/stby/data/data02.dbf
```

… and so on

5. Enable the log transfers from the primary to the new standby database.

```
$ sqlplus sys/oracle@PRODDB as sysdba
SQL> alter system set log_archive_dest_state_2=ENABLE;
```

6. If standby redo logs are used, do the managed recovery using the current log file.

```
SQL> alter database recover managed standby database using current
logfile disconnect;
```

7. If no standby redo logs are used, then:

```
SQL> alter database recover managed standby database disconnect;
```

8. Verify the SCNs are tracking each other.

```
SQL> select instance_name, substr(host_name,1,30) "Host Name" from
v\$instance;
SQL> select DATABASE_ROLE,current_scn,protection_mode from
v\$database;

INSTANCE_NAME  Host Name
---------------- ------------------------------
PRODDB         aie-6300b

DATABASE_ROLE  CURRENT_SCN PROTECTION_MODE
---------------- ----------- --------------------
PRIMARY            5990864 MAXIMUM PERFORMANCE
============================================================

INSTANCE_NAME  Host Name
---------------- ------------------------------
PRODDB_DR      aie-6300c

DATABASE_ROLE  CURRENT_SCN PROTECTION_MODE
---------------- ----------- --------------------
PHYSICAL STANDBY     5990862 MAXIMUM PERFORMANCE
```

Result: The standby database setup over Sun ZFS Storage Appliance is complete.

Cloning the Standby Database

Use this procedure to perform snapshots on the standby database, create clones, and convert the cloned database into a development database for testing.

Step 1. Creating a snapshot of the standby database file systems

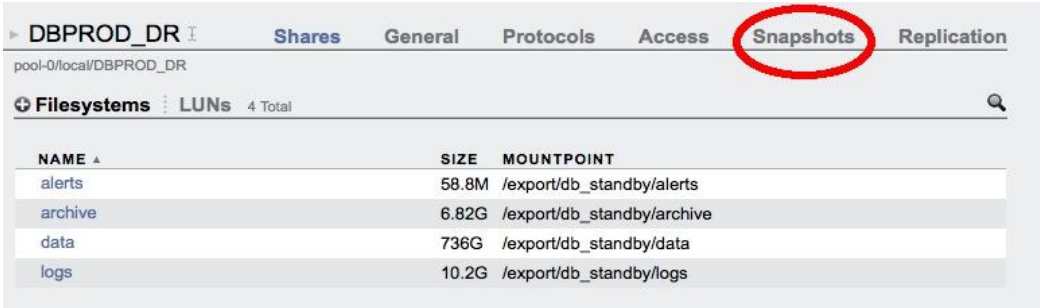
Suspend the managed recovery for the standby database

A point-in-time snapshot is taken for the standby database that is stored under the Sun ZFS Storage Appliance project PRODDB_DR. Before taking the snapshot, the managed recovery is canceled and the log shipment from the primary is stopped. These steps are addressed in the shell script *before_snap.sh*.

```
$ sqlplus sys/oracle@PRODDDB
SQL> alter system set log_archive_dest_state_2=DEFER;
$ sqlplus sys/oracle@PRODDDB_DR
SQL > alter database recover managed standby database cancel;
SQL> alter database recover managed standby database disconnect ;
SQL> alter database recover managed standby database cancel ;
```

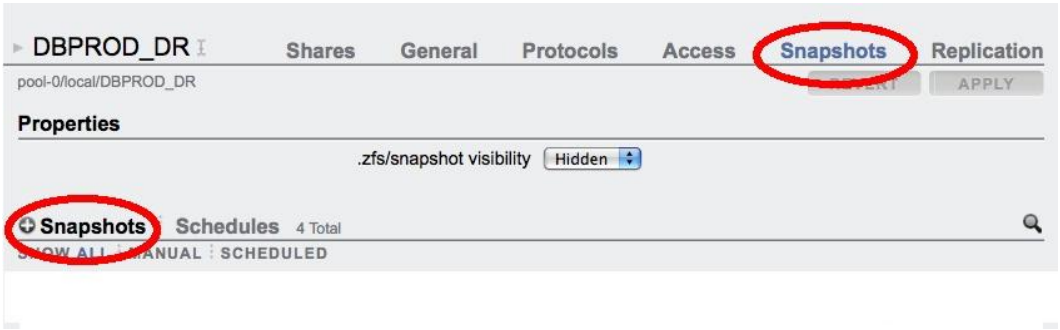
Snapshot the project PRODDB_DR

Snapshot the project in the Sun Storage 7000 appliance using the name *SNAP1*. You can do this from the graphical user interface (GUI) or command-line interface (CLI). By taking a snapshot at the project level, a consistent snapshot is taken across all file systems in the project. All file systems will have a snapshot created with the name *SNAP1*.



1. From the GUI, select the project *DBPROD_DR*, then click *Snapshots*.

2. Click (+) Snapshots.



3. Enter *SNAP1* for the snapshot name.

   SNAP1 appears in the snapshot list, with the creation date, time, and size.

With that step, the snapshot creation is complete.

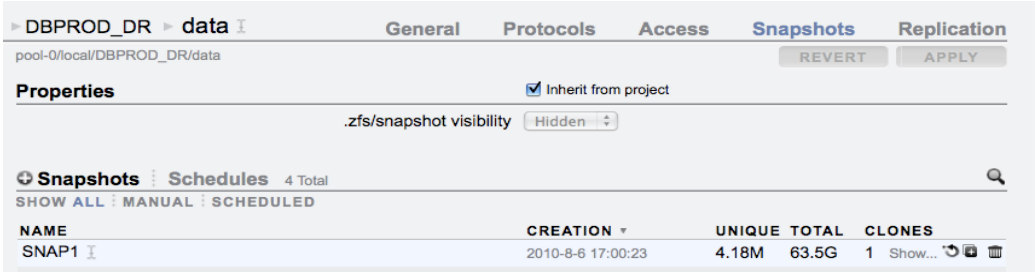
With the following CLI script, *take_snap.aksh*, a new snapshot is taken.

Note: Each snapshot within a project must have a unique name.

```
$ ssh -T root@aie-7420b < take_snap.aksh

$ cat take_snap.aksh

script
{
    var projName='DBPROD_DR';
    var snapName='SNAP1' ;
    var poolName='pool-0' ;
    printf("Snapshot the project..\n");
    run('cd /');
```




```
 run('shares');
 run ('set pool=' + poolName) ;
 run('select ' + projName );
 run('snapshots snapshot ' + snapName) ;
}
```

Resume the managed recovery of the standby database

Once the snapshot is taken, the managed recovery process is resumed for the standby database.

1. Use this script to resume the managed recovery process.

```
$ sqlplus sys/oracle@PRODDDB
SQL> alter system set log_archive_dest_state_2=ENABLE;
$ sqlplus sys/oracle@PRODDB_DR
SQL > alter database recover managed standby database using
current logfile disconnect;
```

Result: The standby database is now able to catch up with the primary production database.

Step 2. Creating clones In the Sun ZFS Storage Appliance

Though the snapshot is created at the Project level, a clone can be created only from the snapshot that is at the file system level. In this configuration, the clones are created from the data, log alerts, and archive file systems.

Create a clone project

In order to store the cloned file systems from the snapshot *SNAP1*, a project with name *CLONE1* is created first. Then, from the *SNAP1* snapshot from each file system from the snapshot project *DB_PRODDR*, the clones are created and stored under the *CLONE1* project.

From the Sun ZFS Storage Appliance:

1. If the clone project with the name *CLONE1* already exists, either it is deleted or renamed.

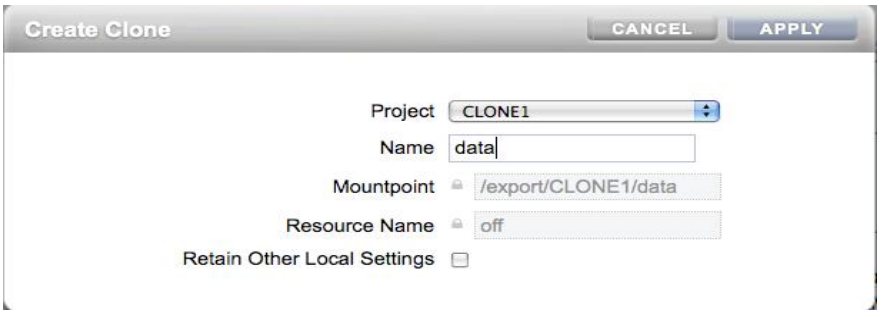
2. A project with name *CLONE1* is created. For optimal performance, default record size is set to 8K. The read access time update time is set to false.

Clone file systems from the snapshots

1. Select the project *DBPROD_DR*.

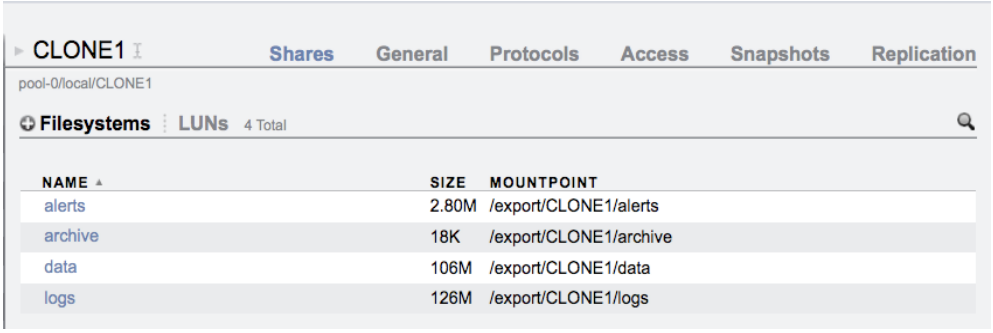
2. For every file system under that project (under the Snapshot tab), click the *clone* icon.

3. In the pop-up screen, select the project *CLONE1* from the drop-down box.
4. Enter the file system name, which is the same name as the original file system.
5. Enter the mount point */export/CLONE1/*, as shown in the following screenshot.



6. Repeat the procedure for the data, logs, archive, and alerts file systems.

   All of the file systems under CLONE1 are ready to be accessed for read/write operations by clients.



With that step, the cloning operation in the Sun ZFS Storage Appliance is complete. The same GUI procedure can be executed without any user intervention by running the script *create_clone.aksh*.

Note : If the clone project name already exists, then either rename it or delete it before executing the script.

```
$ ssh -T root@aie-7420b < create_clone.aksh
```

```
$ cat create_clone.aksh
script
{
    var projName='DBPROD_DR';
    var cloneProj='CLONE1' ;
    var snapName='SNAP1' ;
    var poolName='pool-0' ;

    printf("Creating the clone project\n");
    run ('cd /');
    run ('shares');
    run ('set pool=' + poolName) ;
    run ('project ' + cloneProj);
    try {
    run ('set recordsize=8K');
    run ('set atime=false');
    run ('confirm commit');
    } catch (err) { printf("Done..\n"); }
    printf("Cloning the shares..\n");
    run('cd /');
    run('shares');
    run ('set pool=' + poolName) ;
    run('select ' + projName );
    var sharesList = list() ;
    for (var i = 0; i < sharesList.length; i++) {
       run('select '+ sharesList[i]);
       run('snapshots select ' + snapName );
       printf("Cloning the share %s \n", sharesList[i]);
       run('clone ' + cloneProj + ' ' + sharesList[i]);
       run('set mountpoint=/export/' + cloneProj + '/' + sharesList[i] );
       run('commit') ;
       run('cd ..');
       run('cd ..');
       run('cd ..');
    }
    printf("Cloning the project completed..\n");
}
```

Result: The cloning procedure of the file systems is complete in the appliance.

Step 3. Converting the cloned standby database to a read/writable database for test,dev

In this step, the cloned database – which is also a standby – is converted to a read/writable database which is then used for test, development, and QA purposes. This section provides configuration steps when the standby and clone databases exist on the same host and on different hosts.

Mount the clone database's file systems

If the Standby database and cloned database are in the same database server,

1. Mount the NFS file systems under */oradata/CLONE1/** directories.
2. Proceed to activate the cloned database.

If the Standby database and cloned database are in different database servers,

1. Mount the NFS file systems under */oradata/CLONE1/** directories.
2. The standby redo logs (in */oradata/stby/logs)* are still being referenced by the cloned database's control file. A soft link is placed:

```
# mkdir -p /oradata/stby >/dev/null
# ln -s /oradata/CLONE1/logs /oradata/stby/logs
```

3. Proceed to activate the cloned database.

Convert cloned standby database into read-writable database

1. From the cloned database server, start the *CLONE1* instance in a mount state.

```
SQL> startup mount
pfile=/export/home/oracle/products/11.2.0/db/dbs/initCLONE1.ora
```

The data files in the control file points to the standby database's location.

2. Use the following SQL command to rename only the data files to point from */oradata/stby/data/** to */oradata/CLONE1/data/**.

```
SQL>ALTER DATABASE RENAME FILE '/oradata/stby/data/data00.dbf' TO
'/oradata/CLONE2/data/data00.dbf';
SQL>ALTER DATABASE RENAME FILE '/oradata/stby/data/data01.dbf' TO
'/oradata/CLONE2/data/data01.dbf';
```

and so on ..

3. Drop the standby redo logs (for cleanup purposes). Note that these SRLs are still pointing to the stby location in the control file.

```
SQL>alter database drop standby logfile '/oradata/stby/logs/stby-
1.log';
```

```
SQL>alter database drop standby logfile '/oradata/stby/logs/stby-
2.log';
SQL>alter database drop standby logfile '/oradata/stby/logs/stby-
3.log';
```

4. Activate the cloned standby database to a primary database. Then, shut down and restart the database.

```
SQL> alter database activate standby database ;
SQL> shutdown immediate ;
SQL> startup
```

5. Verify the original standby database is intact and catching up with the primary production database.

```
 -bash-3.00$ ./check_scn.sh

INSTANCE_NAME   Host Name
---------------- ------------------------------
PRODDB          aie-6300b

DATABASE_ROLE   CURRENT_SCN PROTECTION_MODE
---------------- ----------- --------------------
PRIMARY             6119051 MAXIMUM PERFORMANCE

============================================================

INSTANCE_NAME   Host Name
---------------- ------------------------------
PRODDB_DR       aie-6300c

DATABASE_ROLE   CURRENT_SCN PROTECTION_MODE
---------------- ----------- --------------------
PHYSICAL STANDBY     6119049 MAXIMUM PERFORMANCE

============================================================

INSTANCE_NAME   Host Name
---------------- ------------------------------
CLONE1          aie-6300c

DATABASE_ROLE   CURRENT_SCN PROTECTION_MODE
---------------- ----------- --------------------
PHYSICAL STANDBY     5674818 MAXIMUM PERFORMANCE
```

Result: The cloned database is available for test, development, and QA process.

Example Deployment Scenarios

The standby database and the cloned database can be deployed in many combinations. A few scenarios are discussed in this section. In each scenario:

- One or more standby databases are already established and configured to be a primary fail-over target(s) for the Oracle Data Guard.
- Sun ZFS Storage Appliance has been integrated into the infrastructure
- The new standby database stored in the Sun ZFS Storage Appliance can only be used for cloning purposes.

The database cloning infrastructure falls under one of these broad categories.

1. New standby database instance and the cloned database instance(s) are on the same host.

   See sections and .

2. New standby database instance and the cloned database instance(s) are on different hosts.

   See section .

Note: In all of these configurations, the procedures to perform the snapshot and clone in Sun ZFS Storage are identical.

Deployment Scenario 1: One host for all the standby databases and clones

Use Case: The existing standby database server is to be configured for testing and development databases.

Figure 3 shows the Sun ZFS Storage Appliance is added to the infrastructure. The standby site is also configured with appropriate initialization parameters for the new NFS-based standby database and the cloned database. After the snapshot and clone operations, this host serves three database instances – two standby and one cloned.

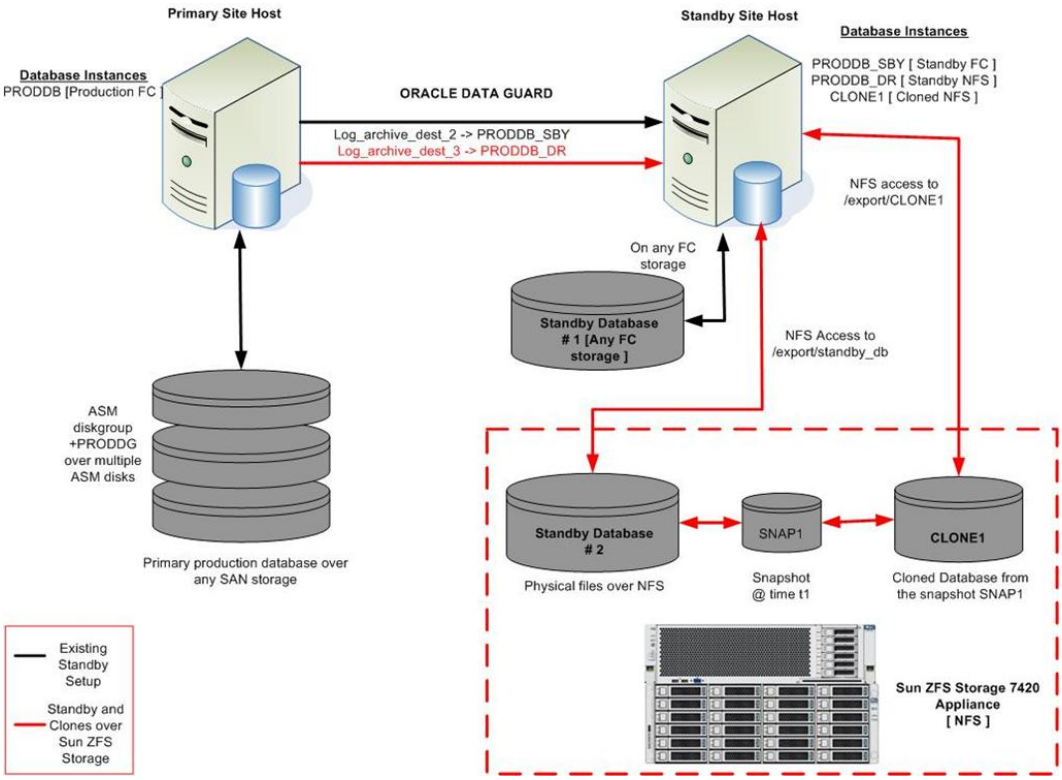


Figure 3: Scenario 1–Same host configured to host the new standby and cloned database

Deployment Scenario 2: Separate host for the cloned databases

Use Case: The standby database host has resources for one more instance. Development and testing databases will be hosted on a separate server.

Figure 4 shows the existing standby site configured to also serve the NFS-based standby database. A dedicated host is configured for hosting one or more cloned database instances. In this configuration, the test, development, and QA operations don't interfere with the standby databases.

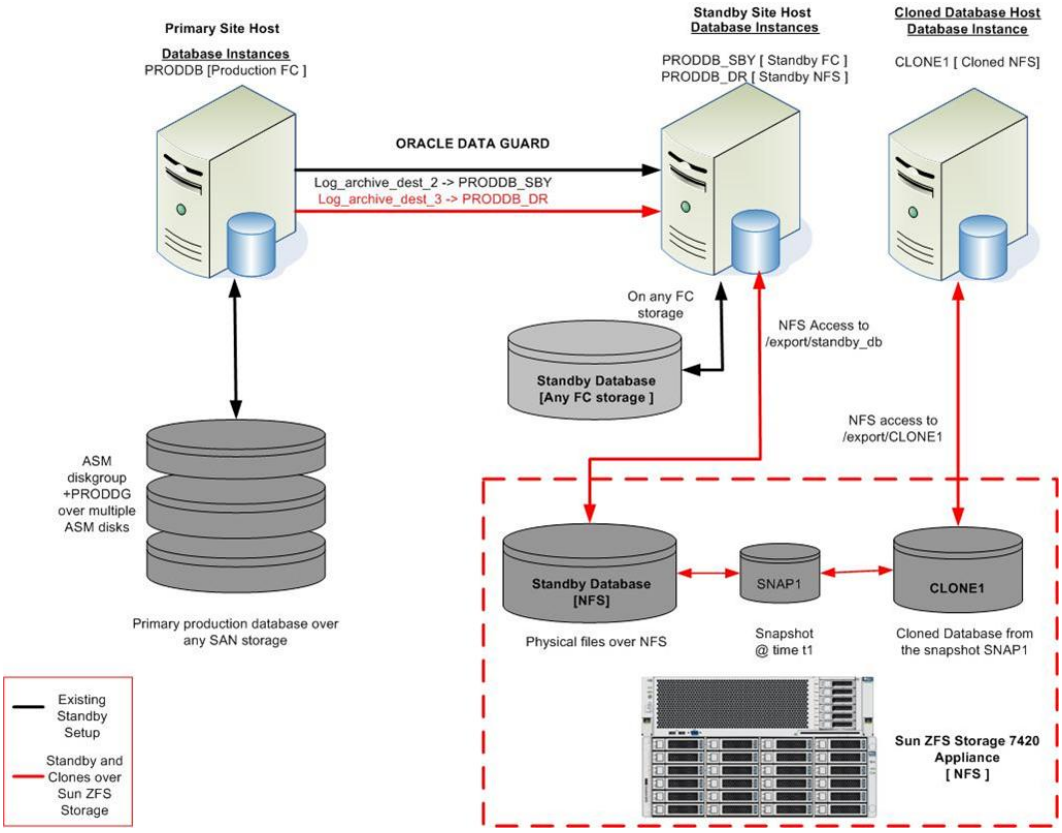


Figure 4: Scenario 2–Separate host for the cloned database(s)

Deployment Scenario 3: New host for standby and clone

Use Case: Existing standby database setup should not be disturbed for test and development purposes.

Figure 5 shows a separate host identified for both the NFS-based standby database (*PRODDB_DR*) and the cloned databases (*CLONE1, CLONE2*) for development and testing purposes. The clones are created based on two snapshots – *SNAP1* and *SNAP2* – taken at different times in the Sun ZFS Storage Appliance. In this new host, *ORACLE_HOME* is shared by all the three instances – *PRODDB_DR, CLONE1,* and *CLONE2*.

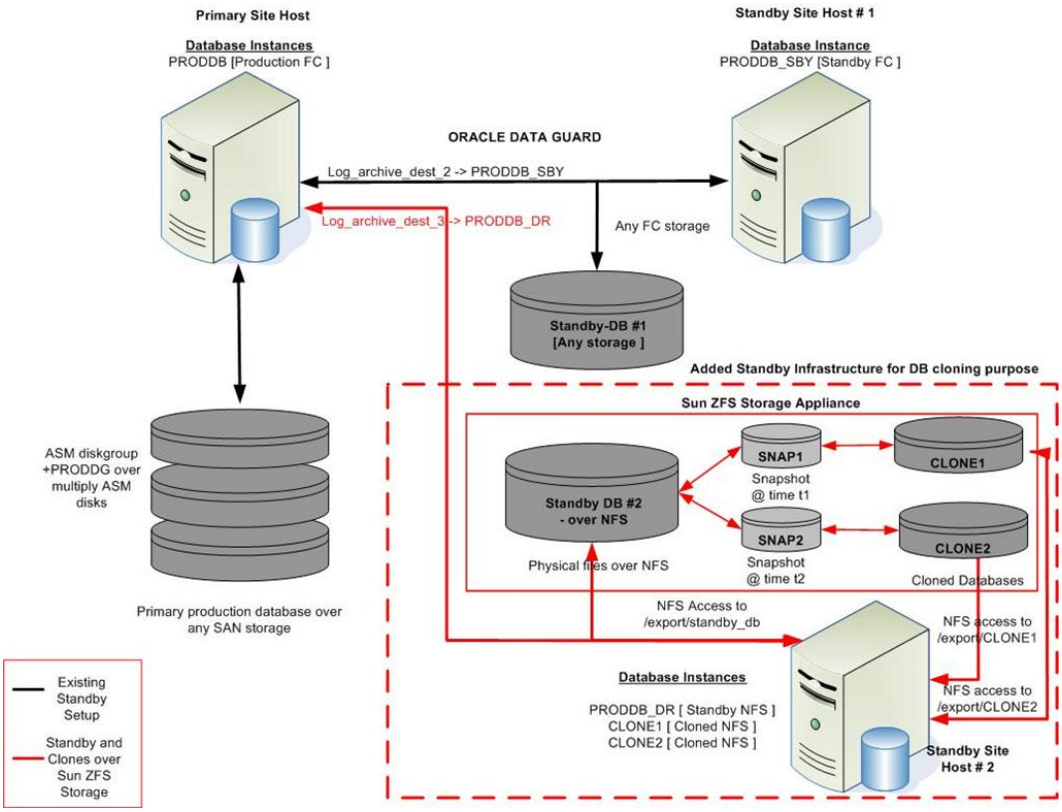


Figure 5: Scenario 3–New host for standby and clone

Benefits of Deploying Database Cloning with Sun ZFS Storage Appliances

Oracle-tested and validated solution

The procedure described in this document has been tested and validated by Oracle. The provided solution description and script samples will help you accelerate test and development cycles, yielding faster time to deployment.

Ease of deployment and management

With multi-protocol support, the Sun ZFS Storage Appliance fits into any infrastructure. The user interface provides intuitive and convenient ways to manage the appliance. The entire database cloning solution can be scripted and executed repeatedly, reducing the time and resources required for the cloning operation.

Efficient backup, business continuity, and disaster recovery

All the data services, such as replication, snapshots, cloning, etc., are included in the price of the appliance. There is no additional cost involved for enabling any feature or protocol. All these features can be used in combinations to meet specific needs for easier backup (certified with a number of leading backup applications), business continuity for almost instantaneous restores, and remote replication for disaster recovery purposes.

Oracle Maximum Availability Architecture best practices recommend that Oracle Data Guard be utilized for disaster recovery protection for the Oracle Database. Sun ZFS Storage Appliance complements Oracle Data Guard by providing replication and disaster protection for file system data that resides outside of the Oracle Database.

Unlimited database cloning with efficient space utilization

There are many traditional methods available to perform the duplication of the production Oracle database. Traditional methods create a full, point-in-time copy (or clone) of the Data Guard copy at the secondary site. These methods, however, are time-consuming and require huge amounts of storage. For example, cloning six copies of a 1TB database would require nearly six times more storage space.

When the snapshot is taken, no space is initially allocated. Likewise, when the clone is created from the snapshot, new blocks are written only when changes are made. So, when accessing the clone, the data may be referencing to the data block of the standby database. This enables efficient space utilization when deploying more clones. More than one clone can be created from the same snapshot and a clone can be destroyed any time. This provides a huge advantage if

something goes wrong with a cloned instance, because the clone can be destroyed and recreated again. The appliance allows unlimited snapshots and clones for maximum deployment flexibility.

Hybrid storage pool enables quicker response time

The database data can take advantage of the Hybrid Storage Pool (HSP) feature of the ZFSSA to utilize the DRAM, Flash, and hard disk media, enabling clients to have faster access. The database's latency sensitive writes are served by the write-optimized Flash, providing sub-millisecond faster response time. The read-optimized Flash acts as a second tier of cache in the storage that stores the recently and frequently accessed blocks. The least accessed data is stored on high capacity disks. For the snapshots and cloned copies, if multiple clones access the same block, the data is served from either DRAM or the Flash, which provides accelerated response times. Figure 6 illustrates the various components involved in the HSP model.

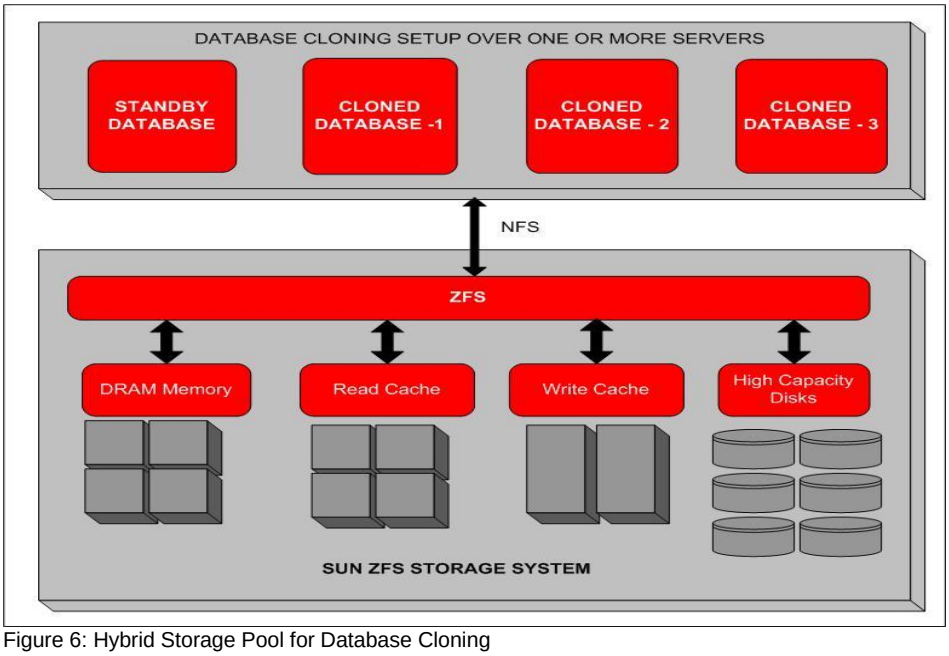
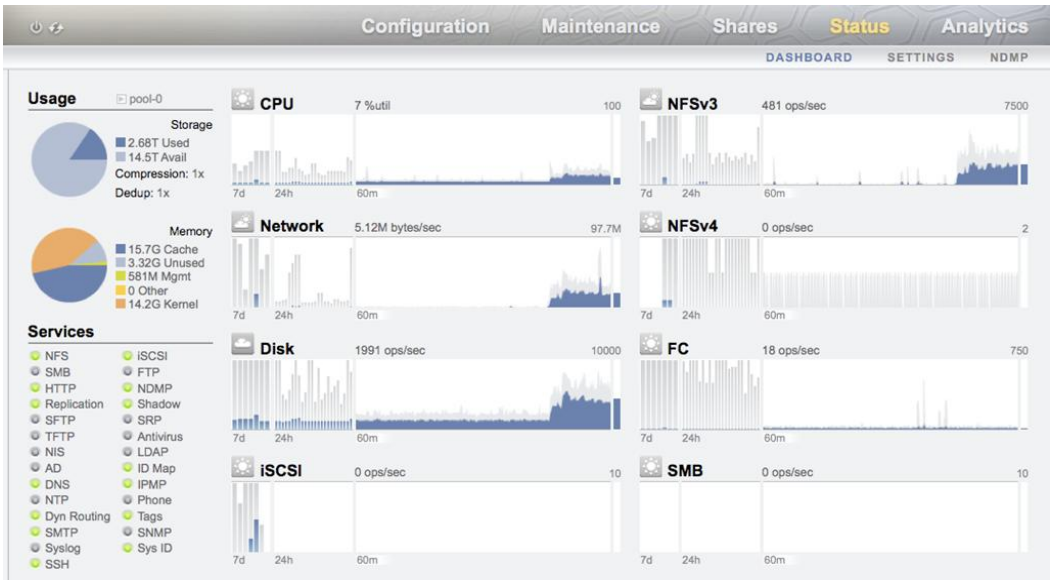


Figure 6: Hybrid Storage Pool for Database Cloning

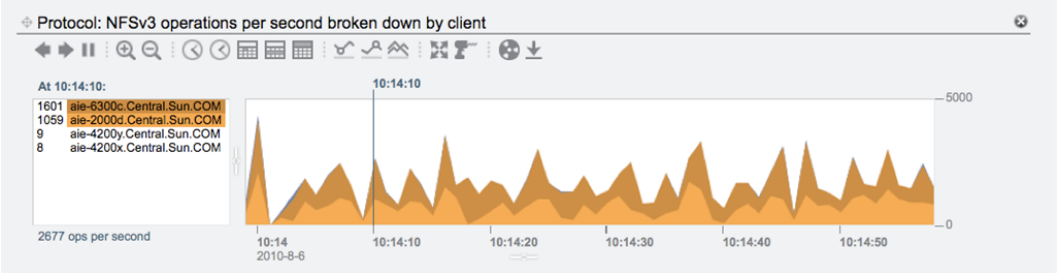
Analytics for faster resolution and planning

The Analytics feature of the ZFS Storage Appliance provides a helpful graphical representation regarding the performance of the various components of the storage appliance. For example:

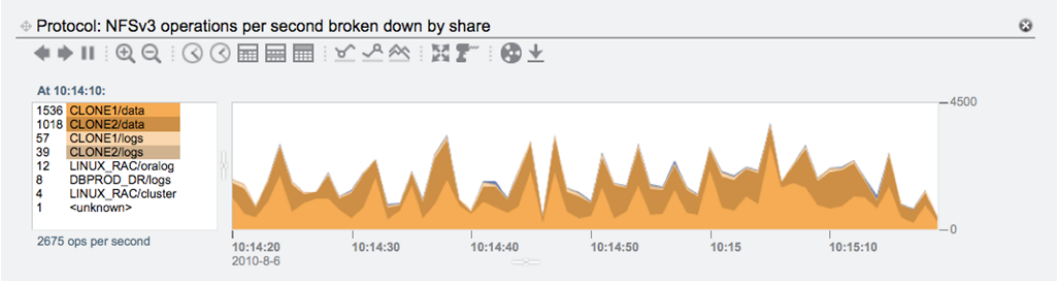
- The Dashboard provides an overall view of what is going on with the system. This screenshot shows space utilization and activity for the different protocols.



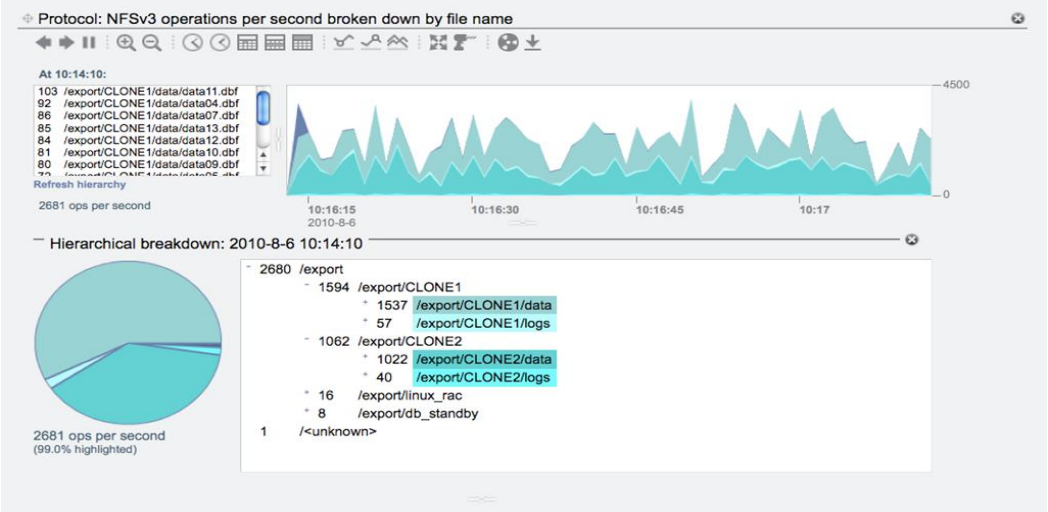
- The following Analytics screenshot shows which clients are accessing the system via NFSv3 protocol.



- The following Analytics screenshot shows which file systems are being accessed.



- The following Analytics screenshot shows the detailed access pattern for the various file systems.



The Analytics feature helps in

a) understanding the overall health of the appliance

b) observe the IOPS, latency, and throughput of the appliance

c) understanding the client(s) access pattern for supporting SLA agreements

d) capacity planning requirements

e) identifying and resolving problems by comparing the performance data between the clients and the appliance

f) exporting the historical data for analysis and so on.

Conclusion

The combination of Oracle Data Guard and Sun ZFS Storage Appliance is an ideal solution for database cloning. The appliance comes with a user-friendly interface for easier management, a full set of data services for business continuity and disaster recovery purposes, multi-protocol support to cater to any infrastructure, analytics for monitoring and resolution purposes, and a hybrid storage pool for faster response time for test, development, and QA efforts. Using Oracle Data Guard in conjunction with the scripts and workflows described in this paper, the entire cloning procedure can be automated, executed efficiently and repeatedly.

With unlimited snapshots and cloning possibilities, many concurrent database instances can be launched for various purposes without impacting the production database. All of this makes Sun ZFS Storage Appliance a viable solution platform for database cloning purposes.

Appendix

Sample Scripts

Note that certain values are hard-coded in the script. This includes database server host names, ORACLE_SID, storage host names, storage pool name, project names, snapshot names, cloned project name, mount points, and so forth. Replace them with your environment's values.

Script Name / Description	Script
Shell Scripts	
initiate_clone.sh This script performs the entire cloning operation. It suspends the standby database, creates a snapshot, creates clones out of the snapshot, and moves them under the clone project. It then mounts the cloned file system in the client, mounts the database, then converts the standby database to a read/writable database for test, development, and QA purposes.	echo "Suspending the automatic recovery of the standby database before taking the snapshot.." ./before_snap.sh echo "Doing the snapping and cloning process in the Sun ZFS Storage.." echo "Enter password for the root user for the S7000 storage at the prompt.." ssh -T root@aie-7420b.central.sun.com < s7000_clone_project.aksh echo "Resuming the automatic recovery of the standby database.." ./after_snap.sh echo "Unmounting and remounting the clone mount points in the DB server" echo "Enter password for the root user for this system at the prompt.." ssh root@aie-6300c /export/orahome/DBCLONE_1/scripts/mount_clone.sh $1 echo "Converting the clone standby to a clone primary DB and open for read/write" #echo "Press <Enter> to continue, <Ctrl+C> to abort.." #read c ORACLE_SID=$1; export ORACLE_SID sqlplus -S / as sysdba @clonedb echo "Cloning the database is complete !! .. Press <Enter> to continue.. <Ctrl+C> to abort.." read b echo "Verifying the content.." ./verify_db.sh PRODDB ./verify_db.sh $1 echo "Cloning process complete !!" ./check_scn.sh
before_snap.sh This procedure is to be done at the standby site before initiating the snapshot.	ORACLE_SID=PRODDB_DR; export ORACLE_SID sqlplus -S / as sysdba << EOF select DATABASE_ROLE,current_scn,protection_mode from v\$database; alter database recover managed standby database cancel; alter database recover managed standby database disconnect; alter database recover managed standby database cancel; EOF
after_snap.sh This procedure is to be done after the snapshot is taken.	ORACLE_SID=PRODDB_DR; export ORACLE_SID sqlplus -S / as sysdba << EOF alter database recover managed standby database using current logfile disconnect; EOF
check_scn.sh This script verifies the status and current SCN from the primary, standby, and the cloned database.	sqlplus -S sys/oracle@PRODDB as sysdba << EOF1 select instance_name, substr(host_name,1,30) "Host Name" from v\$instance; select DATABASE_ROLE,current_scn,protection_mode from v\$database; exit EOF1 echo "==" ORACLE_SID=PRODDB_DR; export ORACLE_SID sqlplus -S / as sysdba << EOF2 select instance_name, substr(host_name,1,30) "Host Name" from v\$instance; select DATABASE_ROLE,current_scn,protection_mode from v\$database; exit

	EOF2 echo "==" ORACLE_SID=CLONE1; export ORACLE_SID sqlplus -S / as sysdba << EOF3 select instance_name, substr(host_name,1,30) "Host Name" from v\$instance; select DATABASE_ROLE,current_scn,protection_mode from v\$database; exit EOF3 echo "=="

mount_clone.sh
 	
Creates mount points and mounts the cloned database shares.
	#!/bin/sh
	echo "Creating directories and mounting the clone $1"
	shares="data logs archive alerts"
	
	for i in $shares; do
	  echo "Processing the share $i"
	  /usr/bin/mkdir -p /oradata/$1/$i > /dev/null
	  /usr/sbin/umount /oradata/$1/$i > /dev/null
	  /usr/sbin/mount -F nfs -o
	rw,bg,hard,nointr,rsize=32768,wsize=32768,proto=tcp,vers=3,actimeo=0 aie-7420b:/export/$1/$i /oradata/$1/$i
	done
	df -k | grep $1

Standby DB SQL scripts	
mount_standbydb.sql This SQL script mounts the standby database and enables the managed recovery using current logfile.	startup nomount pfile=/oracle/products/11.2.0/db/dbs/initPRODDB_DR.ora alter database mount standby database; alter database recover managed standby database using current logfile disconnect; recover managed standby database cancel; alter database open read only; alter database recover managed standby database disconnect using current logfile; select instance_name, host_name from v$instance; select DATABASE_ROLE,current_scn,protection_mode from v$database; exit
Clonedb.sql ORACLE_SID=CLONE1 This SQL script mounts the cloned standby database, renames the data files, and activates the standby to a primary database for test, development, and QA.	shutdown immediate startup mount pfile=/oracle/products/11.2.0/db/dbs/initCLONE1.ora @rename_files select name from v$datafile; alter database drop standby logfile '/oradata/stby/logs/stby-1.log'; alter database drop standby logfile '/oradata/stby/logs/stby-2.log'; alter database drop standby logfile '/oradata/stby/logs/stby-3.log'; select member from v$logfile; # alter database activate standby database ; # shutdown immediate startup pfile=/oracle/products/11.2.0/db/dbs/initCLONE1.ora select instance_name, substr(host_name,1,25) "Host Name",status,instance_role from v$instance; exit
Sun ZFS Storage Appliance scripts	

take_snap.aksh
 	
This script will create a snapshot with the name snapName.
	
	script
	{
	  var projName='DBPROD_DR';
	  var snapName='SNAP1' ;
	  var poolName='*pool-0*' ;
	  printf("Snapshotting the project..\n");
	  run('cd /');
	  run('shares');

	run ('set pool=' + poolName) ;
	run('select ' + projName);
	run('snapshots snapshot ' + snapName) ;
	}

create_clone.aksh
 	
This script creates a clone project and then assigns the snapshots of all the shares of the project under this cloned project.
	script
	{
	var projName='*DBPROD_DR*';
	var cloneProj='*CLONE1*' ;
	var snapName='*SNAP1*' ;
	var poolName='*pool-0*' ;
	
	printf("Creating the clone project\n");
	run ('cd /');
	run ('shares');
	run ('set pool=' + poolName) ;
	run ('project ' + cloneProj);
	try {
	run ('set recordsize=8K');
	run ('set atime=false');
	run ('confirm commit');
	} catch (err) { printf("Done..\n"); }
	printf("Cloning the shares..\n");
	run('cd /');
	run('shares');
	run ('set pool=' + poolName) ;
	run('select ' + projName);
	var sharesList = list() ;
	for (var i = 0; i < sharesList.length; i++) {
	run('select '+ sharesList[i]);
	run('snapshots select ' + snapName);
	printf("Cloning the share %s \n", sharesList[i]);
	run('clone ' + cloneProj + ' ' + sharesList[i]);
	run('set mountpoint=/export/' + cloneProj + '/' + sharesList[i]);
	run('commit') ;
	run('cd ..');
	run('cd ..');
	run('cd ..');
	}
	printf("Cloning the project completed..\n");
	}

Glossary

CIFS

Common Internet File System protocol. Also called Server Message Block (SMB) protocol.

Clones

The appliance supports an unlimited number of clones. A clone is an instantaneously-created, read-writable copy of a snapshot. One or more clones can be created from a single snapshot. These clones are presented to users as a normal file system. All the regular operations are allowed on a clone, including taking a snapshot from the clone. Clones are typically used in a test, development, QA, or backup environment.

Similar to snapshots, no space is allocated when a clone is created.. The reads to the clone are served by the base file system's blocks. The changed blocks are allocated only when the blocks are changed in the clone. Since the space is shared between snapshots and clones, and since a snapshot has multiple clones, a snapshot cannot be destroyed without also destroying any active clones.

From a client's perspective, the clone file systems are shown as though they are an independent file system. No special requirement for accessing the clones is needed.

NDMP

Network Data Management protocol.

NFS

Network File System protocol.

Project

A project defines a common administrative control point for managing shares. All shares within a project can share common settings, and quotas can be enforced at the project level in addition to the share level. Projects can also be used solely for grouping logically-related shares together, so their common attributes (such as accumulated space) can be accessed from a single point. All file systems and LUNs are grouped into projects. Typically, every application has its own project. Also considered a "consistency group."

Remote replication

Data is asynchronously replicated to the targets, which can then be used for disaster recovery purposes. The replication can be set to happen continuously, according to the user-defined

schedule or on demand. Sun ZFS Storage Appliance supports remote replication of data from one appliance to one or more appliances.

Shares

Shares are file systems and LUNs exported over supported data protocols to clients of the appliance. A share is created under a project. File systems export a file-based hierarchy and can be accessed over CIFS, NFS, HTTP/WebDav, and FTP. LUNs export block-based volumes and can be accessed over iSCSI.

The project/share is a unique identifier for a share within a pool. Multiple projects can contain shares with the same name, but a single project cannot contain shares with the same name. A single project can contain both file systems and LUNs, and share the same namespace.

Snapshot

The Sun ZFS Storage Appliance has unlimited snapshot capability. Snapshots are the read-only point-in-time copies of a file system, instantaneously created and with no space allocated initially. Blocks are allocated as and when changes are made to the base file system (copy-on-write). Snapshots are either initiated manually or can be automated by scheduling at specific intervals. These snapshot data can be directly accessed for any backup purposes.

Any reads to the snapshot blocks are served by the base file system's block. As the changes happen to the base file system, the older block referenced by the snapshot and the new changed block is referenced by the file system.

Project snapshots are the equivalent of performing snapshots on all shares within the project.

Snapshot rollback

The process that returns the base file system to the point in time when the snapshot was taken. The rollback process discards all changes that occurred to the base file system from the time the snapshot was taken until the time of the rollback. Snapshot rollback removes the need for data restore process.

Storage Pool

The storage pool (similar to a volume group) is created over a set of physical disks. File systems are then created over the storage pool. One or more storage pools are created over the available physical disks and flash drives are assigned. The storage pool is configured with a RAID layout such as mirrored , RAID-Z (single parity), RAID-Z2 (dual parity) and so on.

Resources

- Oracle Maximum Availability Architecture Web site
http://www.otn.oracle.com/goto/maa

- *Oracle Database High Availability Overview (Part #B14210)*
http://otn.oracle.com/pls/db111/db111.to_toc?partno=b28281

- *Oracle Database High Availability Best Practices (Part B25159)*
http://otn.oracle.com/pls/db111/db111.to_toc?partno=b28282

- *Oracle Data Guard with Oracle Database 11g Release 2 Technical Information*
http://www.oracle.com/technetwork/database/features/availability/dataguardoverview-098960.html

- *Oracle Data Guard 11g Release 2 Technical White Paper*
http://www.oracle.com/technetwork/database/features/availability/twp-dataguard-11gr2-1-131981.pdf

- *Sun ZFS Storage Appliance Documentation*

http://wikis.sun.com/display/FishWorks/Documentation

- *Deploying Oracle Databases on Sun Storage 7000 Unified Storage*

http://www.sun.com/bigadmin/features/articles/7000_oracle_deploy_2009q3.jsp

- *Configuring Sun Storage 7000 for Oracle Databases*

http://wikis.sun.com/display/SystemsComm/Configuring+Sun+Storage+7000+Unified+Storage+Systems+for+Oracle+Databases

- *Backup and Recovery of Oracle Databases with Sun Storage 7000*

http://www.sun.com/bigadmin/features/articles/oracle_7000_snapshot.jsp

ORACLE

Oracle Database Cloning Solution Using Sun ZFS Storage Appliance And Oracle Data Guard

January 2012

Author: Sridhar Ranganathan
Contributing Authors: Joseph Meeks, Larry Carpenter, David Krenik

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

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

0109