An Oracle White Paper
July 2014

Optimizing Storage for Oracle Database 11*g* Release 2 with the Oracle ZFS Storage Appliance

Contents

Executive Overview

Databases are the backbone of today's business, whether providing transaction integrity for key business systems such as payment engines or providing the core analytical data for decision-making. These broad and varied uses require similarly varied resources and investment to meet different needs like potentially high IOP or capacity requirements. The architecture of the Oracle ZFS Storage Appliance makes it an ideally suited platform to provide the flexibility required for the ever-changing availability, capacity and performance requirements of the business. With its market-leading benchmark results in SPC-1, SPC-2 and SPEC-SFS, the Oracle ZFS Storage Appliance provides a high performance, low cost, low risk storage platform for databases and general-purpose storage.

Please see "Realizing the Superior Value and Performance of the Oracle ZFS Storage Appliance" to learn more about the business value of the Oracle ZFS Storage Appliance.

As growing businesses strive to capture and manage data in a structured fashion, while maintaining corporate compliance, increasing amounts of storage are required to hold the vast amount of data being generated. At the same time, performance requirements to maintain access times that are responsive for the business continually grow. Current storage arrays have evolved in feature benefits, but unlike their server counterparts, have not tracked Moore's Law in performance; instead, capacity increases while the performance is simply maintained, resulting in a utilization efficiency gap. Combine this utilization inefficiency with the inability to handle multiple database loads at a controller level, and today's storage requirements for hosting databases start to impact the data center in an uncontrollable fashion – with the ultimate impact on the IT budget and business profit and loss.

Unlike traditional solutions, Oracle's hardware and software solutions are engineered to work together. Combining this engineering with the extreme performance and market-leading efficiency inherent in the flexible architecture of the Oracle ZFS Storage Appliance produces the ideal platform for database consolidation.

The Oracle ZFS Storage Appliance takes advantage of the latest developments in CPU design as well as direct random access memory (DRAM) and flash technology to provide unprecedented caching performance for both reads and writes without the traditional need for lots of Hard Disk Drive (HDD) spindles demonstrated by traditional vendors. This breakthrough technology, implemented as a Hybrid Storage Pool, provides data centers efficiencies that reduce both acquisition costs and total cost of ownership while delivering market-leading performance.

Please see "Architectural Overview of the Oracle ZFS Storage Appliance" for more information on how these efficiency and performance benefits are achieved.

This paper provides an overall view of the architecturally driven features of the Oracle ZFS Storage Appliance and how they provide superior performance for Oracle Database – in this discussion, specifically Oracle Database 11*g* Release 2.

Why Oracle ZFS Storage Appliance for Oracle Databases

The Oracle ZFS Storage Appliance is a unified storage system that allows multiple heterogeneous physical connections, and which supports many storage protocols. The Oracle ZFS Storage Appliance can be physically connected to Ethernet, InfiniBand, or Fibre Channel networks, and can serve NFS (Network File System), SMB (Server Message Block), HTTP, TFTP/SFTP/FTP (HyperText, Trivial File, Secure or SSH File and simple File Transfer Protocols), iSCSI LUNs (Internet Small Computer System Interface LUNs), NDMP (Network Data Management Protocol), SRP (Secure Remote Password) and native Fibre Channel LUNs.

Running Oracle Databases on Oracle ZFS Storage Appliance over NFS provides users with an easy to manage and flexible infrastructure capable of running a large number of database instances.

Ethernet-based NFS facilitates simple provisioning, management, and sharing plus the flexibility to rapidly move and repurpose database and storage resources. NFS implementations are also typically cost-effective as they do not require special Fibre Channel SAN installation, expertise or training, and they allow IT teams to leverage standard Ethernet knowledge.

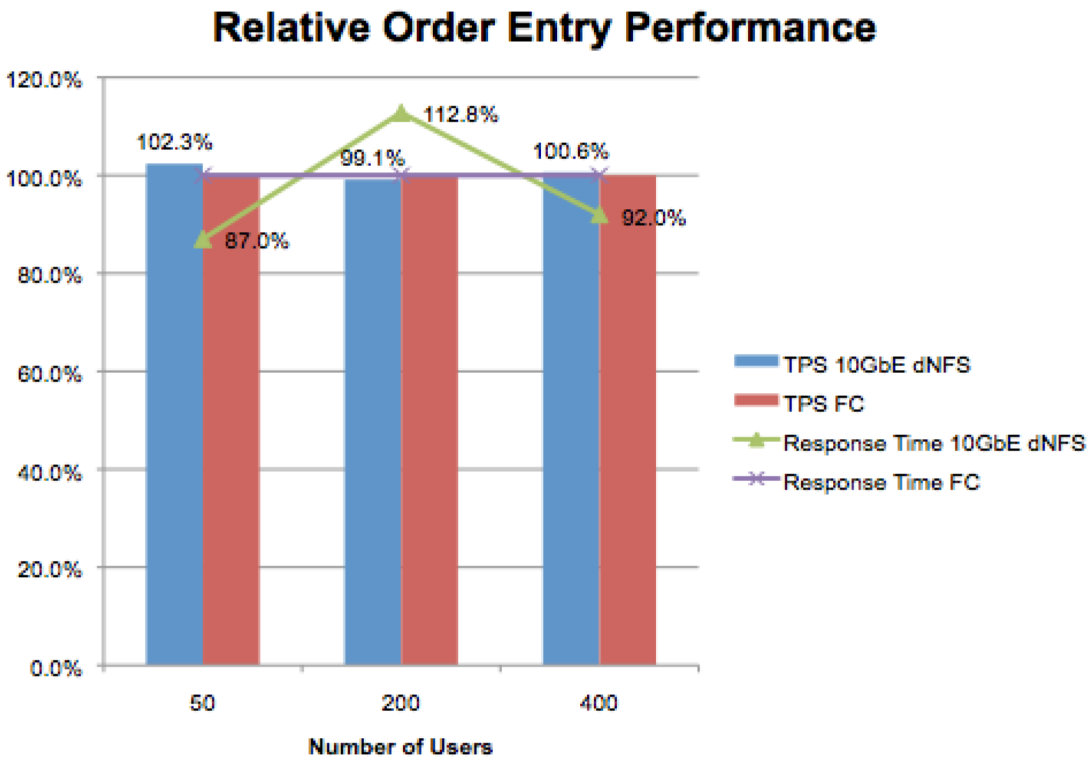


Figure 1. Comparison of Order Entry Transaction over different protocols

Oracle over NFS can be deployed for use in virtually any type of Oracle environment, ranging from test and development to 24x7 production online transaction processing (OLTP), enterprise resource planning (ERP), and custom applications. Oracle has tested multiple protocols against the same workload; figure 1 shows the results. Comparing order entry transactions across multiple workloads, the results show that when properly configured and tuned, Oracle on Direct NFS delivers comparable performance to Fibre Channel using identical hardware (aside from network interface cards [NICs] and host bus adapters [HBAs]) and test conditions. This is accomplished without the additional complexity and cost of storage area network (SAN) Fabrics and without the complexity of optimizing the NFS stack on both the storage array and the clients as would be needed with kernel NFS.

Enhancing NFS for Database Workloads

NFS version 3 (NFSv3), the version that is in wide use today, consists of server software – for instance, the software that runs on the Oracle ZFS Storage Appliance – and client software running on database servers or other hosts that require access to network storage. Achieving good performance and ensuring proper behavior require that both sides of the connection, client and server, be correctly implemented.

As Oracle engineers hardware and software together, optimizations continue to be made in both the Oracle ZFS Storage Appliance and Oracle operating systems (Oracle Solaris and Oracle Linux) to ensure desired behaviors. Oracle has developed a database-centric Direct NFS (dNFS) client that speeds access to Oracle databases over NFS protocols.

Oracle Direct NFS explained

The development of Oracle dNFS was based on Oracle's own experience of managing multiple databases on NFS storage across Oracle's internal and customer-hosted systems. This ensured design goals based on real world experience and business need.

Oracle dNFS was introduced in Oracle Database 11g Release 1. The release embeds an NFS client in the Oracle Database Kernel, with the design objective to improve performance while accessing NFS-based storage, increase high availability, and optimize scalability of the complete architecture, while eliminating complex NFS administration and optimizations. The resulting implementation ensures that consistent configurations can take place across multiple operating systems, including but not limited to Oracle Solaris, Oracle Linux and Microsoft Windows.

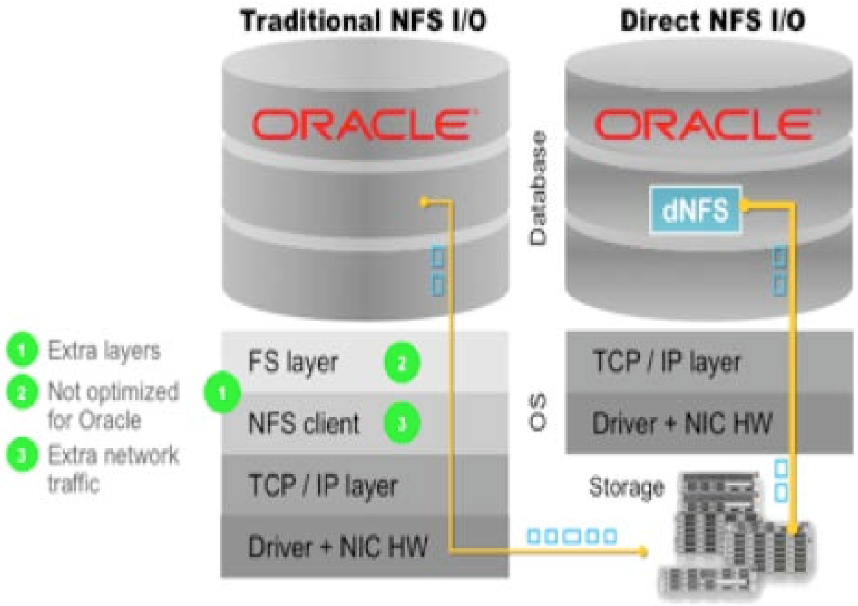


Figure 2. Traditional NFS compared to dNFS

By implementing dNFS in the Oracle Stack, storage can be accessed through the TCP/IP layer while removing the unnecessary overheads introduced with a typical operating system.

For more information on dNFS, consult the *Oracle Database Installation Guide* for your particular operating system, and reference Section 5.3.9, "Configuring and Using Direct NFS Client." Installation Guides are available at: http://docs.oracle.com/cd/E11882_01/nav/portal_11.htm.

The Oracle ZFS Storage Appliance Analytics feature allows users to easily see some of the benefits from using dNFS for Oracle databases. Analytics can provide in-depth feedback on I/O operations and the impact of the Hybrid Storage Pool and the disk drives themselves. This feedback enables educated decision making while sizing, for instance, and insights to understanding where appropriate upgrades can be made. The following figure shows the Oracle ZFS Storage Appliance browser user interface (BUI) displaying the creation of a simple tablespace using NFS shares mounted without dNFS enabled, using the following CLI command:

```
CREATE TABLESPACE ZFSTS_NFS DATAFILE '/ora2/NFS1.dbf' size 512m;
```

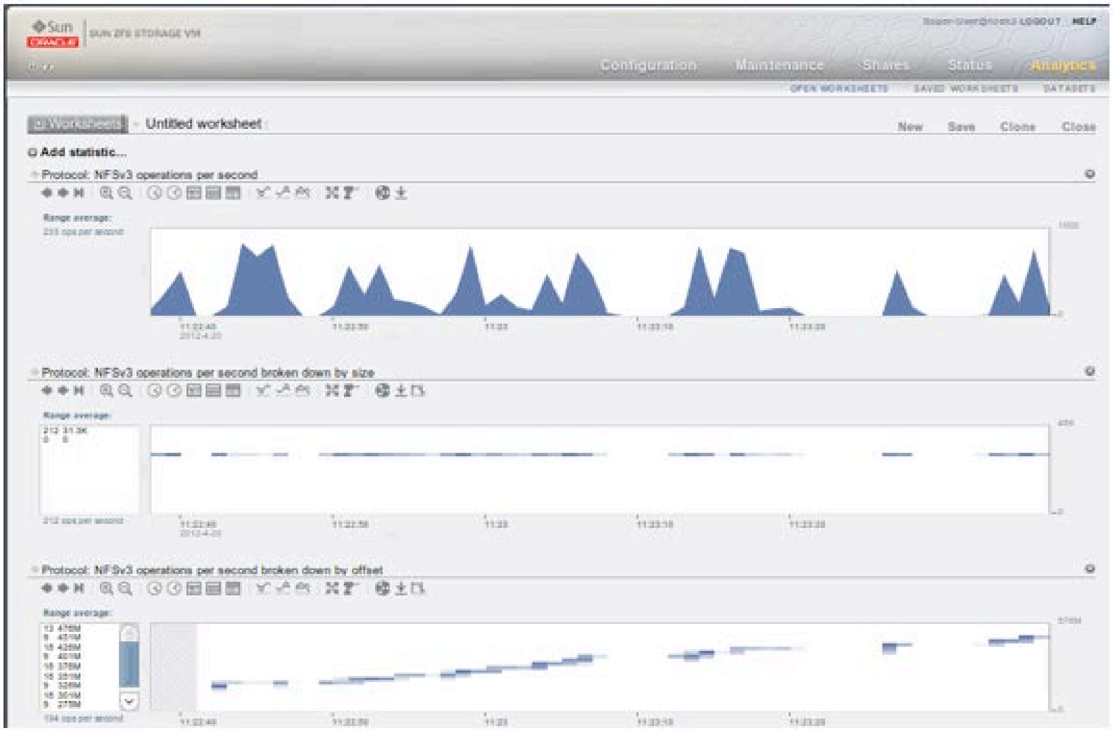


Figure 3. Oracle ZFS Storage Appliance Analytics demonstrating a Create Tablespace I/O profile

As can be seen, when creating the tablespace, Oracle performs a large amount of I/O as it sequentially writes to the data file. Following the current Oracle recommendations, the size of each I/O is limited to 32 k by the NFS mount options.

```
rsize=32768,wsize=32768
```

This results in a large number of small IOPS for table creation. It is possible to ignore the Oracle recommendations and increase the values of `rsize` and `wsize`; however, the actual maximum transfer size for NFS operations is also restricted by kernel limits and the NFS implementation for the client. For example, in Oracle Solaris 10, the system tunable parameters influence the size of NFS I/O.

```
nfs:nfs3_bsize
nfs:nfs3_max_transfer_size
```

By default, this is set to 32 k and needs to be tuned before changing the values of the NFS mount options.

Using dNFS allows Oracle to perform NFS operations to the Oracle ZFS Storage Appliance without being constrained by any NFS client implementation. The following figure this time shows the tablespace created in the same manner as the previous example but with dNFS enabled, so the Oracle Database is now able to create the tablespace more efficiently, using a smaller number of 1 MB I/Os rather than many more 32 k I/Os.

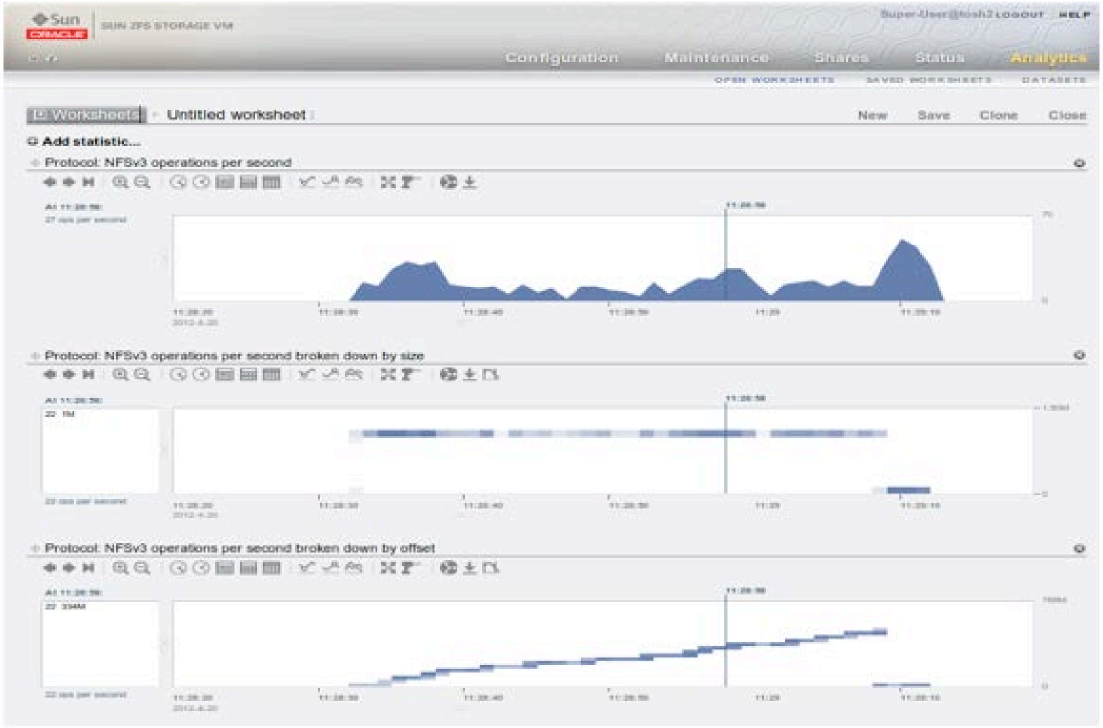


Figure 4. Oracle ZFS Storage Appliance Analytics demonstrating a Create Tablespace I/O profile with dNFS enabled

Removing the Capacity Performance Divide

Determining and controlling what type of media (SSD, Fibre Channel, SATA) to use for storing a consolidated database's data presents significant challenges to users of traditional storage systems. These challenges are further complicated as the access requirements and value of the data changes over time.

The Hybrid Storage Pool architecture of Oracle ZFS Storage Appliance provides an effective solution for efficiently optimizing data placement. Hybrid Storage Pools dynamically and intelligently cache active data over multiple tiers of storage media without the need for administrator intervention. Since the Hybrid Storage Pool provides around-the-clock automatic caching, changing business needs are easily and efficiently processed by the storage system. Figure 5 shows the Hybrid Storage Pool architecture.

To learn more about this unique, DRAM-centric architecture, please review the white paper entitled "Architectural Overview of the Oracle ZFS Storage Appliance."

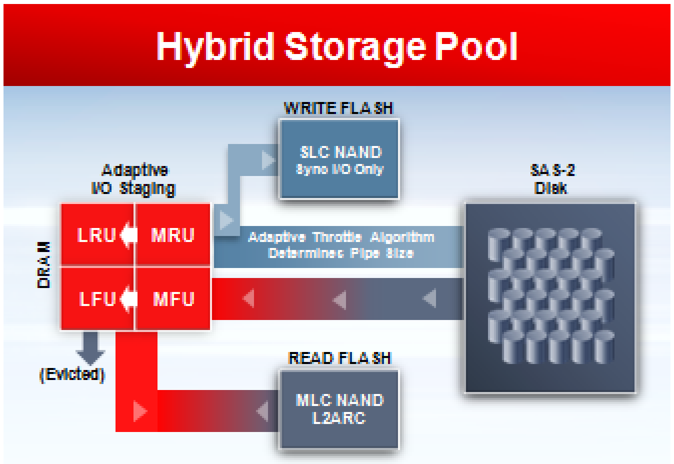


Figure 5: The ZFS Hybrid Storage Pool Architecture

Note the following abbreviations used in the figure diagram: block classifications LRU (least recently used), LFU, least frequently used, MRU (most recently used), MFU (most frequently used), SLC (Single Level Cell), MLC (Multi Level Cell), NAND (Flash technology that offers faster erase, write, and read capabilities.

Reducing Capacity Requirements

A major optimization implemented with the Oracle ZFS Storage Appliance combined with Oracle Database 11g Release 2 is Hybrid Columnar Compression[2]. Hybrid Columnar Compression was introduced to help address the ever-growing storage capacity requirements. As data is becoming more valuable for business and the need to meet ever-increasing compliance regulations, database sizes are growing from an average in the year 2000 of hundreds of gigabytes to terabytes in size. With large-scale data warehouses regularly exceeding 500 TB, database size introduces dramatic burdens on a business's data reporting, since often the growing volume of data can cause query times to exceed business service levels.

Customers who implement Hybrid Columnar Compression see an average of 10:1 compression ratios, with some data being compressed as high as 50:1. Oracle provides a PL/SQL script called `DBMS_COMPRESSION` in Oracle Database 11*g* Release 2 as a tool for estimating compression benefits when applying Hybrid Columnar Compression. The following figure shows a sample output comparing a traditional Advanced Compression Option suited to OLTP data and Hybrid Columnar Compression techniques, with some tables yielding a compression ratio of 52:1.

[2] Oracle Hybrid Columnar Compression on Exadata

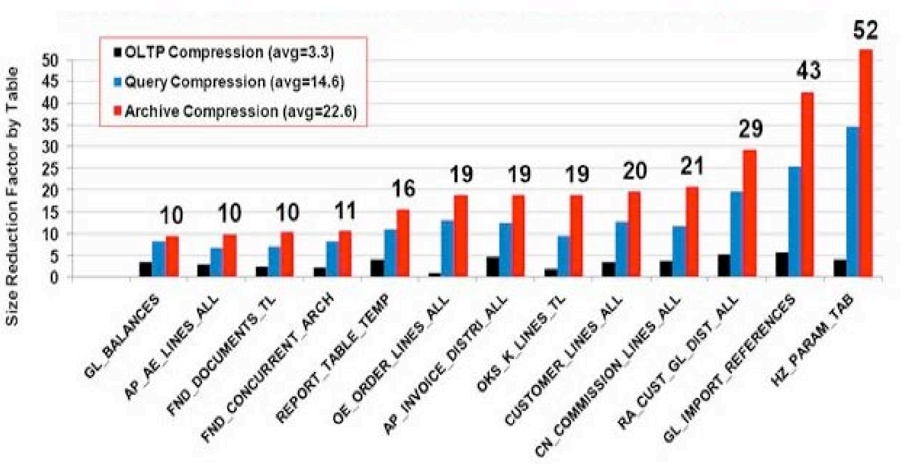


Figure 5. Comparing table compression methods in an Oracle E-Business Suite production environment

Appendix A provides further details on how to obtain these results and determine the benefits of applying Hybrid Columnar Compression on your database.

IT departments are often under pressure not just to reduce the storage space they need, but also the amount of data being stored. Hybrid Columnar Compression addresses this by changing the traditional row-based storage of the database to include a more columnar structure, with data stored and organized in database blocks based on similar characteristics. As the Hybrid Columnar Compression name implies, the hybridized storage method addresses both row and column, and in so doing, achieves compression efficiencies that cannot just be gained at the storage level. This hybrid approach achieves the compression benefits of columnar storage, while avoiding the performance shortfalls of a pure columnar format.

Figure 6 helps illustrate the compression methodology. A logical construct called the compression unit is used to store a set of hybrid columnar-compressed rows. When data is loaded, column values for a set of rows are grouped together and compressed. After the column data for a set of rows has been compressed, it is stored in a compression unit.

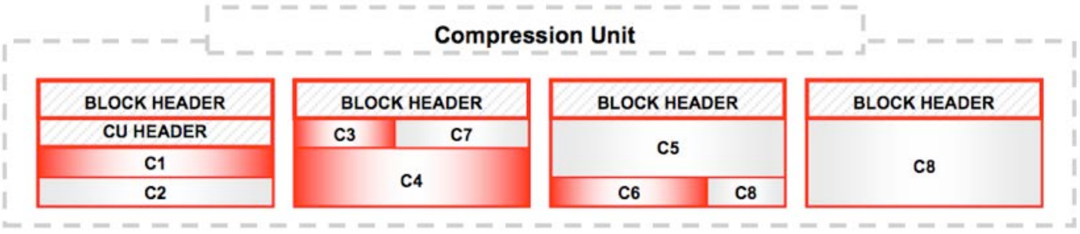


Figure 6. Conceptual illustration of a Logical Compression Unit

Queries that run against Hybrid Columnar Compression data are returned from disk in a compressed format to the server, reducing the amount of I/O required across a fabric and in turn reducing the amount of time a server sits in IO WAIT. With the traditional I/O bottleneck removed, the servers can be optimally utilized.

Although Hybrid Columnar Compression was developed to address the needs of Data Warehouses, Hybrid Columnar Compression is also relevant in traditional OLTP databases as part of an Information Lifecycle Management (ILM) strategy. Traditionally, as databases have grown in size, database administrators have utilized partitions to optimize for performance as well as place less accessed data on cheaper storage tiers. Hybrid Columnar Compression can be combined with partitions to reduce the space required for archive/historic data while still ensuring that the data remains online and accessible. Oracle Database features a Compression Advisor you can use to evaluate the benefits of Hybrid Columnar Compression on your particular system. More information on Compression Advisor is available at:
http://www.oracle.com/technetwork/database/options/compression/index-095686.html

Reducing the Cost of Operational Management

With increasing data requirements and the need for businesses to be operational 24/7 for 365 days of the year there is an increasing need to have a centralized management and reporting of the IT infrastructure. Oracle has continued to enhance its software stack with Oracle hardware to optimize operations management and to remove the complexity and the cost from the environment.

Oracle Enterprise Manager 12c Cloud Control

Oracle Enterprise Manager 12c Cloud Control provides the ability to monitor and report on applications activity through to the disk level, enabling easier reporting and faster root cause analysis while removing the dark art of mapping transactions and their effect on the infrastructure. The Oracle ZFS Storage Appliance, through its customized plug-in, fits into this framework with its Analytics for the storage workloads and integrates into the Cloud Control interface to provide a single pane of glass for Oracle ZFS Storage Appliance management and monitoring. This enables Oracle Database administrators to either own the storage or to work with the storage experts to understand service impacts and reduce the time to problem or incident resolution.

The Oracle Enterprise Manager Cloud Control Plug-in for the Oracle ZFS Storage Appliance enables performance and capacity reporting, threshold and hardware monitoring, and customizable alerts based on dozens of metrics collected from the Oracle ZFS Storage Appliance.

Storage virtualization features of the Oracle ZFS Storage Appliance

The storage virtualization features of the Oracle ZFS Storage Appliance can also reduce the burden on the IT department. The Oracle ZFS Storage Appliance enables Quotas, Thin Provisioning, Snapshots, Replication and Cloning, all managed through delegated administrators. These capabilities means database administrators no longer request storage far greater than their requirement, which has been a traditional approach to avoid the time lag required to have storage provisioned.

With the introduction of Oracle Real Application Clusters (Oracle RAC), the burden on storage management to reconfigure SAN fabrics and manage the added risk of making multiple LUNs available to multiple hosts at the same time increased. With the Oracle ZFS Storage Appliance environment, this complexity is removed. NFS is a shared file system by design and Oracle RAC is tested and certified with the Oracle ZFS Storage Appliance.

The Oracle ZFS Storage Appliance provides additional capabilities to the Oracle Database to ease data management.

Snapshots

The Oracle ZFS Storage Appliance Snapshot feature allows an unlimited number of consistent point-in-time copies to be created with minimal space utilization in the storage pools. This can provide the database administrator with an added level of protection for the database beyond the best practice provided by the Maximum Availability Architecture (MAA). Snapshots ensure the ability to restore key configuration files, or tables, from a known time point seamlessly without the need for an Oracle Recovery Manager (Oracle RMAN) restore operation.

When consolidating an application and database on the same storage array, snapshots can ensure their consistency at the point of the snapshot. This additional layer of data protection can still be integrated with an Oracle RMAN strategy enabling applications to be consistently protected at a known state.

Snapshots can also be integrated with the Oracle recommended backup approach of image copy and incremental merge to provide fast restore times for full copies and independent incremental restore points to correct known data corruptions.

Clones

The Oracle ZFS Storage Appliance provides the ability to create an 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 normal file systems. After presentation to a client, all regular operations are permitted, including taking further snapshots from a clone.

With this ability it becomes possible to provide databases cloned from production for test, development, QA and backup environments.

Similar to snapshots, when the clone is created, no space is allocated. The base file serves the reads to the clone 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; however, security settings can be set to limit client access to different clones and the master data.

Oracle Snap Management Utility for Oracle Database[3]

Oracle Snap Management Utility for Oracle Database automates the creation and management of snapshot-based copies and clones of Oracle databases, accelerating processes and empowering database administrators with quick test, development, and data protection environments. The tool provides a web-based BUI and a command-line interface for creating and managing snapshots and clones of Oracle databases that reside on the Oracle ZFS Storage Appliance.

Applying Best Practices for Deploying Oracle Database 11*g*R2 on the Oracle ZFS Storage Appliance

Although deploying Oracle Database on any NFS storage is straightforward, Oracle has formalized a variety of best practices to optimize performance and manageability. As the Oracle ZFS Storage Appliance can be procured in a variety of configurations, the following section focuses on the appropriate controller configuration and file system settings that apply to all models.

Storage Pool Configuration

The Oracle ZFS Storage Appliance offers a number of RAID levels, such as mirrored, single parity RAID-Z, dual parity RAID-Z2, and so on. The following RAID levels applicable for either production or a development test environment are discussed:

- Mirroring is used for storage for general-purpose applications and databases that require optimal performance. With mirroring, total raw capacity is reduced by 50 percent.
- RAID-Z offers 50 percent higher capacity than mirroring with reasonable performance, and can tolerate a single disk failure. RAID-Z is suitable when optimal capacity is required.
- RAID-Z2 increases capacity, with the benefit of higher availability, but reduces performance compared to RAID-Z. RAID-Z2 is ideal for streaming data but should be used with caution in high random I/O applications.

The Oracle SAME (Stripe and Mirror Everything) guidelines promote the use of mirroring for all database file systems. This approach is ideally suited to production databases and those that require optimum performance. For test and development environments, either mirrored or RAID-Z can be used based on the architect's choice of performance against available storage space required.

[3] Oracle ZFS Snap Management Utility

RAID-Z is also appropriate for large streaming workloads such as RMAN backups, which are recommended as part of a data protection strategy. To this end, when implementing a Fast Recovery Area (FRA) it can also be appropriate to utilize RAID-Z to provide a balanced configuration of performance and capacity.

The Oracle ZFS Storage Appliance allows configuration of multiple storage pools and, in dual controller configurations, a storage pool can be presented to each controller to ensure maximum utilization is provided by the array. In shared infrastructure where production and test development environments can coexist, it is recommended to create two pools and locate the correct database to the correct pool of storage. To ensure maximum storage utilization and efficiency it is recommended not to create more than two pools of storage per controller. This will help to maximize storage utilization and reduce administration burden.

For best performance, it is recommended that the use of ZFS storage pools be monitored, with the objective of remaining under 80 percent of total capacity. Due to the copy-on-write architecture of the ZFS file system, it is possible that a storage pool can fill up, at which point it becomes difficult to free up space by deleting files or snapshots. One solution to this problem is to add additional disk into the storage pool. The recommended approach is to create a 'dummy' file system in each pool with a reservation of at least 100 MB. If the pool becomes 100 percent full, then an administrator can reduce the reservation on this file system, freeing up space within the pool and allowing further files to be deleted or truncated.

Projects

To provide a level of granularity and control, creating projects for each individual database and its application is recommended. This provides a level of atomic consistency when utilizing snapshots or replication. Projects can also provide defaults of naming convention access and administration authorization in a Role Based Access Control security model.

Share Layout and the Optimal Flexible Architecture

Oracle recommends the use of the Oracle Optimal Flexible Architecture (OFA) standard, which helps to standardize database deployments for performance while addressing maintenance complexity. OFA has the following purposes:

- Find Oracle files in explicit locations
- Provide a software tree that eases patching and upgrades
- Ensure Oracle installations are the same across all environments
- Keep separate Oracle files and installation types
- Facilitate routine management tasks
- Manage and administer growth

- Facilitate layout for best practice

Using the following layout on a host is recommended:

TABLE 1. RECOMMENDED OFA HOST DIRECTORY STRUCTURE

Directory	Path
Oracle Base (`$ORACLE_BASE`)	`/u01/app/oracle`
Oracle Inventory	`$ORACLE_BASE/oraInventory`
Oracle Home	`$ORACLE_BASE/product/<Version>/`

The Oracle Inventory should be created on a local file system to the host where possible. If this is not possible, then the Oracle Inventory should be created under a specific hostname share on the Oracle ZFS Storage Appliance, resulting in the following structure:

TABLE 2. RECOMMENDED OFA SHARE STRUCTURE

Directory	Path
Oracle Base (`$ORACLE_BASE`)	`/u01/app/oracle`
Oracle Inventory	`$HOSTNAME/$ORACLE_BASE/oraInventory`
Oracle Home	`$HOSTNAME/$ORACLE_BASE/product/<Version>/`
Oracle Database Files	`$HOSTNAME/u02/oradata/<DB Name>`
Oracle Recovery Files	`$HOSTNAME/u03/flash_recovery_area/<DB NAME>`

Due to the nature of how the Oracle Database stores data in database blocks, different block sizes and caching can benefit different types of data. To simplify storage provisioning while providing a level of granularity to enable performance optimizations, use the recommended share layout shown in table 3. The table shows an example record size setting for an OLTP type database. For recommended tuning details, please see MOS note 30709.1 and MOS note 957459.1 on the My Oracle Support website, listed in References at the end of this document.

TABLE 3. EXAMPLE ZFS FILE SYSTEM PROPERTIES FOR AN OLTP ORACLE DATABASE

File System	ZFS Record Size	Synchronous Write Bias	Cache Device Usage	Share Name
Table Data	8 KB	Throughput	All data and metadata	`$HOSTNAME/u02/oradata/<DB Name>`
Redo Logs	128 KB	Latency (default)	Do not use Cache Device	`$HOSTNAME/u02/oradata/<DB Name>/redo`
Index Files	8 KB	Throughput	All data and metadata	`$HOSTNAME/u02/oradata/<DB Name>/index`
Undo Data	128 KB	Throughput	Don not use cache device	`$HOSTNAME/u02/oradata/<DB Name>`/undo
Temp Data	128 KB	Throughput	Do not use cache device	`$HOSTNAME/u02/oradata/<DB Name>/temp`
FRA	128KB	Throughput	Do not use cash device	`$HOSTNAME/u03/flash_recovery_area/<DB NAME>/`

The following discussion provides some context to the table columns' property values and why they are being applied.

ZFS Record Size

The ZFS record size governs the size of the file system block. This is the unit that ZFS validates through checksums. As file system blocks are stripped dynamically onto pooled storage, the importance of block alignment is the benefit of cache and storage efficiency resulting in throughput to the database.

Due to the nature of ZFS design – and unlike typical file systems that require one system for the entire file system – ZFS enables a granular approach at the share level. This capability can enable the user to finely tune block alignment to the database record size and achieve a balanced configuration and I/O throughput. Further tuning can be achieved from the outline in table 3 by ensuring that the Oracle `db_block_size` is aligned to the ZFS record size. This tuning may occur, for example, in data warehouse environments or where large batch workloads are required. For more information on database block size tuning, refer to MOS note 30709.1 and MOS note 957459.1.

The exception to the previous rule is when the database will handle Oracle Large Object (LOB). In this case, the LOB is generally stored over many Oracle data blocks. Because this type of data leads to large read and write I/Os, such as full table scan, consider creating a separate file system and specify a larger record size for the LOB table space. Consider the ZFS record size of 128 KB if the average size of the LOB is larger than 256 KB; however, unlike with normal Oracle data blocks, there is no need to align the ZFS record size to the size of the LOB.

Synchronous Write Bias

The Oracle ZFS Storage Appliance provides the ability at the share level to set how the storage array should handle synchronous writes. Because the Oracle ZFS Storage Appliance can utilize write-optimized flash for write caching, storage administrators can optimize their configuration of comparatively expensive write-optimized flash to particular workloads. This includes enabling latency-sensitive workloads (Oracle REDO, for example) to utilize the write-optimized flash while all other writes bypass the write-optimized flash. Such configuration takes advantage of all facilities that are appropriate for the workload and reduces the cost of ownership and acquisition of the storage array.

Cache Device Usage

By default, all data sets make use of any read-optimized flash on the system. However, for some workloads that are write intensive, with little read, it is advisable to ensure balanced usage of cache by disabling the use of caching. This in return should ensure more positive cache hits on access to indexes and table space.

Network Configuration

Network configuration can be a complex subject with many variables to consider. Please refer to the Oracle whitepaper entitled Networking Best Practices with the Oracle ZFS Storage Appliance for an exhaustive treatment of this subject.

Configuring the Client NFS Mount

When configuring the Oracle ZFS Storage Appliance, any server that accesses the Oracle ZFS Storage Appliance is considered a client. Configuring the client NFS mount includes creating a target directory structure for access to the Oracle ZFS Storage Appliance as well as specific NFS mount options[4]. To ensure that the latest improvements are included with the following configuration recommendations, refer to My Oracle Support MOS note 359515.1 before implementing them. Depending on the configuration you are running, you will require three different mount points for RAC or two different mount points for single instance.

Single Instance Linux mount point options:

Data files

```
rw,bg,hard,rsize=1048576,wsize=1048576,vers=3,nointr,timeo=600,tcp,actimeo=0
```

Binaries

```
rw,bg,hard,rsize=1048576,wsize=1048576,vers=3,nointr,actimeo=600,tcp
```

RAC Linux mount point options:

Data files

```
rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,actimeo=0, vers=3,timeo=600
```

Binaries

```
rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,vers=3, timeo=600, actimeo=0
```

CRS voting disk or OCR

```
rw,bg,hard,nointr,rsize=rsize=1048576,wsize=1048576,tcp,noac,vers=3,
timeo=600,actimeo=0
```

[4] NFS Mount options are covered in MOS Note 359515.1.

Single Instance Oracle Solaris mount point options:

Data files

```
rw,bg,hard,rsize=1048576,wsize=1048576,vers=3,[forcedirectio
orllock],nointr,proto=tcp,suid
```

Binaries

```
rw,bg,hard, rsize=1048576,wsize=1048576,vers=3,nointr,proto=tcp,suid
```

RAC Oracle Solaris mount point options

Data files

```
rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto=tcp,noac,forcedirectio,
vers=3,suid
```

Binaries

```
rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto=tcp,noac,vers=3,suid
```

CRS Voting Disk or OCR

```
rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto=tcp,vers=3,noac,forcedirectio
```

NFS Mounts for Oracle RMAN

Along with being a solid platform for Oracle databases, the Oracle ZFS Storage Appliance can also be utilized as an Oracle RMAN target or a Fast Recovery Area for the Oracle Database. For these purposes, use this additional mount point configuration. (Refer to MOS note 1354980.1 for the most up-to-date configuration details.)

RMAN mount point option for Oracle Solaris

```
rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto=tcp,vers=3,foredirectio
```

RMAN mount point option for Linux

```
rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,noac,vers=3,timeo=600
```

When using RMAN with InfiniBand in engineered systems backup, configure the following.

```
rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,vers=3,timeo=600
```

The following figure shows the Oracle ZFS Storage Appliance Analytics screen displaying a sustained throughput of 1.9GB/sec.

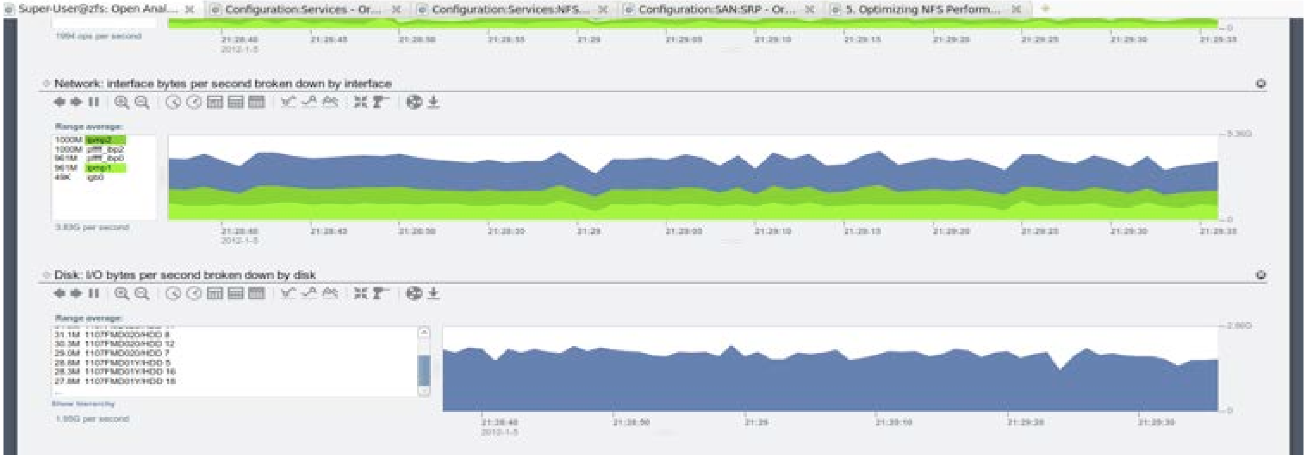


Figure 7. Oracle ZFS Storage Appliance Analytics screen showing RMAN backup stream achieving 1.9 GB/sec

Appendix A: How to Calculate Hybrid Columnar Compression Benefits Using `DBMS_COMPRESSION`

The `DBMS_COMPRESSION` package is included with Oracle Database 11*g* Release 2 and calculates compression-related information within a database environment. This allows a DBA to sample a database table and estimate the likely space savings that can be achieved by implementing different types of compression, including Hybrid Columnar Compression, when run on Oracle Exadata or using Oracle storage such as Pillar Axiom for FC presentation or the Oracle ZFS Storage Appliance for NFS presentation.

Prior to Oracle Database 11*g* Release 2, customers could download the "Oracle Advanced Compression Advisor." However, this tool is only able to report on compression options provided by the Oracle Advanced Compression Option (ACO) and has no recognition of Hybrid Columnar Compression.

The `DBMS_COMPRESSION` package works by taking a sample of the database and applying the relevant compression algorithm to estimate the overall compression ratio. The following procedure demonstrates how to run the compression advisor for a table called SALES that resides in the AX database running under Oracle Database release 11.2.0.3. Log in to the database. Note that the following example reflects that the user is already logged in to the operating system as the Oracle user. Set up the Oracle environment and call `sqlplus` as a user with access to the table. The example shows a login as user `demo`.

```
oracle $ id -a
uid=500(oracle) gid=502(orainst) groups=501(dba),502(orainst
oracle $ . oraenv
ORACLE_SID = [AX] ?
The Oracle base for ORACLE_HOME=/u01/app/oracle/product/11.2.0/db01 is
/u01/app/oracle
oracle
$ sqlplus demo/password
SQL*Plus: Release 11.2.0.3.0 Production on Wed Nov 23 18:32:55 2011
Copyright (c) 1982, 2011, Oracle.  All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options
SQL>
```

List available tables:

For this example procedure, run the compression advisor against the SALES table.

Describe the SALES table:

```
SQL> select * from cat;
TABLE_NAME TABLE_TYPE ------------------------------ ------------------------------
NSEQ SEQUENCE REGIONS TABLE SALES TABLE STATS TABLE T1 TABLE
7 rows selected.

SQL> desc SALES Name Null? Type ----------------------------------------- -------- --
--------------------------

SALE_DATE
SALE_AGENT
ORDER_ID
ITEM
AMOUNT
QUANTITY
REGION
SOURCE_WH
DATE
VARCHAR2(100)
NUMBER(20)
VARCHAR2(100)
NUMBER(10,2)
NUMBER(5)
VARCHAR2(100)
VARCHAR2(100)
```

Display the uncompressed size of the SALES table:

```
SQL> col bytes format 999,999,999,999
SQL> col segment_name format a30
SQL> select segment_name,bytes from user_segments;
SEGMENT_NAME                              BYTES
------------------------------ ----------------

REGIONS T1 SALES_ARCH_HI SALES SALES_BIG
Elapsed: 00:00:00.17
SQL>
65,536 15,800,991,744 949,944,320 6,994,001,920  13,432,520,704
```

List the number of rows in the SALES table:

```
SQL> select count(*) from sales;
COUNT(*) ---------- 100000000
```

```
SQL>
```

Create a SQL script file to calculate likely compression ratios

There are five types of compression that can be enabled within the database. The four Hybrid Columnar Compression levels are:

- COMP_FOR_QUERY_HIGH – High compression level for query operations
- COMP_FOR_QUERY_LOW – Low compression level for query operations
- COMP_FOR_ARCHIVE_HIGH – High compression level for archive operations
- COMP_FOR_ARCHIVE_LOW – Low compression level for archive operations

The Compression Advisor can also be used to calculate compression rates using the 'OLTP' compression methodology. This is recommended for frequently updated data and is a part of the Advanced Compression option.

- COMP_FOR_OLTP – Advanced Compression for OLTP

The following example calculates the compression rates achievable using the COMP_FOR_ARCHIVE_HIGH compression type against the SALES table owned by the user DEMO.

Create a SQL script file as follows:

```
oracle $ cat sales.sql
DECLARE
l_blkcnt_cmp BINARY_INTEGER ;
l_blkcnt_uncmp BINARY_INTEGER ;
l_row_cmp BINARY_INTEGER ;
l_row_uncmp BINARY_INTEGER ;
l_cmp_ratio NUMBER (10,5);
l_comptype_str VARCHAR2(2000) ;
BEGIN
DBMS_COMPRESSION.get_compression_ratio (
scratchtbsname
ownname
tabname
partname
comptype
blkcnt_cmp
blkcnt_uncmp
row_cmp
row_uncmp
cmp_ratio
comptype_str
=> 'USERS', => 'DEMO', => 'SALES', => NULL, =>
DBMS_COMPRESSION.COMP_FOR_ARCHIVE_HIGH, => l_blkcnt_cmp,
=> l_blkcnt_uncmp,
=> l_row_cmp,
=> l_row_uncmp,
=> l_cmp_ratio,
=> l_comptype_str);
DBMS_OUTPUT.put_line ('l_blkcnt_cmp=' || l_blkcnt_cmp);
DBMS_OUTPUT.put_line ('l_blkcnt_uncmp=' || l_blkcnt_uncmp);
DBMS_OUTPUT.put_line ('l_comptype_str= '|| l_comptype_str);
DBMS_OUTPUT.put_line (‘l_cmp_ratio=’ || l_cmp_ratio);
END; /
```

The `scratchtbsname` should be set to a tablespace that can be used to create a sample of the SALES tablespace. USERS is usually adequate for this purpose. The `comptype` should be changed to the compression type to be calculated, one of COMP_FOR_QUERY_HIGH, COMP_FOR_QUERY_LOW, COMP_FOR_ARCHIVE_HIGH, COMP_FOR_ARCHIVE_LOW or COMP_FOR_OLTP.

Calculate the compression ratio

Within SQLPLUS, run the `sales.sql` script.

```
SQL> set serveroutput on SQL> @sales Compression Advisor self-check validation
successful. select count(*) on both Uncompressed and EHybrid Columnar Compression
Compressed format = 1000001 rows l_blkcnt_cmp=784 l_blkcnt_uncmp=8161
l_comptype_str="Compress For Archive High" l_cmp_ratio=10.4
PL/SQL procedure successfully completed.
```

Note: This produced a compression ratio of 10.4:1 which is exactly the same result as observed when the whole table was compressed using 'archive high' but with 100 million rows.

Appendix B: Enabling Oracle Hybrid Columnar Compression on dNFS Mounted Shares

To take advantage of Oracle Hybrid Columnar Compression using dNFS, the following prerequisites are required:

- Oracle 11*g* Release 2 Enterprise Edition 11.2.0.3 and patch 1334343 or later
- Oracle ZFS Storage Appliance with SNMP enabled and accessible from all Oracle Database nodes that want to take advantage of Hybrid Columnar Compression
- An enabled Oracle dNFS client to access the Oracle ZFS Storage Appliance

Enabling the SNMP Service on the Oracle ZFS Storage Appliance

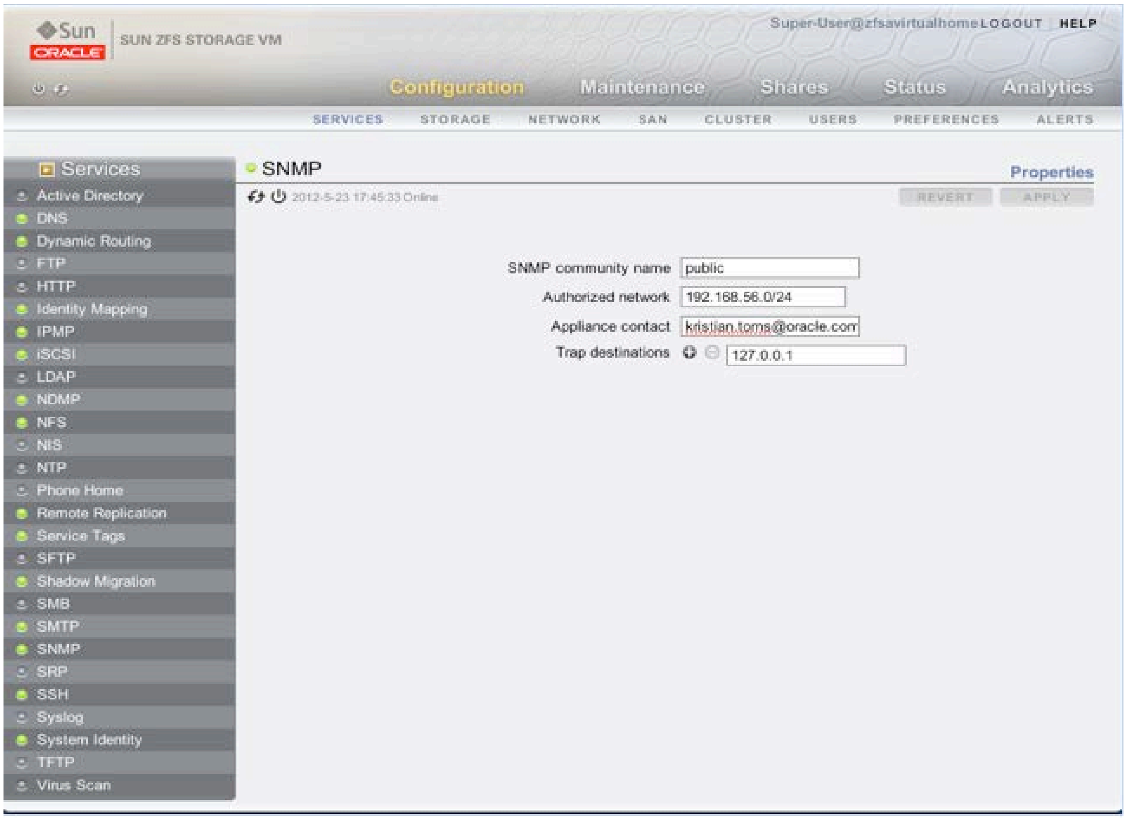
Oracle Database requires that SNMP be enabled and configured for use by Hybrid Columnar Compression when using the Oracle ZFS Storage Appliance. To enable Hybrid Columnar Compression, use the following steps.

Configuring SNMP to serve Oracle ZFS Storage Appliance status and traps

Use either the Oracle ZFS Storage Appliance BUI or CLI for the following:

1. Set the community name, authorized network and contact string.
2. If desired, set the trap destination to a remote SNMP host; otherwise, set this to 127.0.0.1.
3. If desired, set the authorized network to allow SNMP clients; otherwise, set this to 127.0.0.1/8. One of the authorized networks should include the network with the Oracle Database Servers.
4. Apply/commit the configuration.
5. Restart the service.

Figure 8 shows a sample configuration in which the authorized network with the Oracle Database Server is 192.168.56.0/24.



2012-5-23 17:45:33 Online

SNMP community name: public
Authorized network: 192.168.56.0/24
Appliance contact: kristian.toms@oracle.com
Trap destinations: 127.0.0.1

Figure 8. Configuring SNMP in the Oracle ZFS Storage Appliance BUI

To test that the database nodes are receiving the correct SNMP service, use the following command on Oracle Linux or Oracle Solaris.

```
$snmpget -v1 -c public <IP for mount point> 1.3.6.1.4.1.42.2.225.1.4.2.0
```

The command should return the following output:

```
SNMPv2-SMI::enterprises.42.2.225.1.4.2.0 = STRING: "Sun ZFS Storage 7420"
```

Configuring Oracle dNFS

As previously discussed, dNFS is an optional configuration that can improve throughput for Oracle databases. It is recommended that as a minimum Oracle Database Patch 8808984 is applied to ensure optimal dNFS operation. To configure dNFS, complete the following. For the most up-to-date patch and configuration settings, refer to MOS note 762374.1 on My Oracle Support.

1. a. For version 11.2.0.2 or greater of the Oracle Database software, enter:

```
$ make –f \
$ORACLE_HOME/rdbms/lib/ins_rdbms.mk
dnfs_on
```

1. b. For a version prior to 11.2.0.2, enter:

```
$ ln –sf \
$ORACLE_HOME/lib/libnfsodm11.so \
$ORACLE_HOME/lib/libodm11.so
```

Update the `oranfstab` (`/etc/oranfstab`) file with entries showing the channels and shares accessed on the Oracle ZFS Storage Appliance. The following example shows how to access the data share on aie-7420a-h1 over IP addresses 192.168.36.200 server: aie-7420a-h1.

```
$ORACLE_HOME/dbs/oranfstab
server: aie-7420a-h1  <=== ZFS Storage Appliance Hostname
path: 192.168.36.200 <=== First path to ZFSSA i.e. NIC 1
local: 192.168.36.4  <=== First Client side interface
path: 192.168.36.201 <=== Second path to ZFSSA i.e. NIC 2
local: 192.168.36.5  <=== Second Client side interface
export: /export/dbname/data mount: /ZFSSA/dbname/data
export: /export/dbname/index mount: /ZFSSA/dbname/index
export: /export/dbname/redo mount: /ZFSSA/dbname/redo
export: /export/dbname/undo mount: /ZFSSA/dbname/undo
export: /export/dbname/temp mount: /ZFSSA/dbname/temp
```

2. Restart the Oracle Database software instance.

Appendix C: Enabling Oracle Hybrid Columnar Compression on iSCSI or Fibre Channel LUNs

To take advantage of Oracle Hybrid Columnar Compression using iSCSI or Fibre Channel LUNs, the following prerequisites are required:

- Oracle 11*g* Release 2 Enterprise Edition 11.2.0.4 or higher
- An Oracle Automatic Storage Management (ASM) diskgroup with specific attributes configured and consisting only of LUNs residing on the Oracle ZFS Storage Appliance
- The Oracle ZFS Storage Appliance running firmware version 2013.06.05.0.1 or higher

Creating LUNs and Adding Them to an ASM Diskgroup

Before you create an ASM diskgroup, create your iSCSI or Fibre Channel LUNs and discover them using the operating system tools.

For information on how to configure Fibre Channel LUNs on the Oracle ZFS Storage Appliance, refer to the white paper:
"Understanding the Use of Fibre Channel in the Oracle ZFS Storage Appliance,"
http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/o12-019-fclun-7000-rs-1559284.pdf

For information on how to configure iSCSI LUNS with Linux, refer to the article:
"Using Sun ZFS Storage Appliance iSCSI LUNs in an Oracle Linux Environment"
http://www.oracle.com/technetwork/articles/servers-storage-admin/o11-104-iscsi-luns-linux-518400.html

or for Oracle Solaris, refer to the article:
"Using Sun ZFS Storage Appliance iSCSI LUNs in an Oracle Solaris Environment"
http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/iscsi-quickstart-v1-2-051512-1641594.pdf

Use the `asmca` tool or `asmcmd` command to create an ASM diskgroup consisting only of LUNs that reside on the Oracle ZFS Storage Appliance. Refer to the Oracle Automatic Storage Management Administrator's Guide for information on using `asmca` or `asmcmd` to create an ASM diskgroup.
http://docs.oracle.com/cd/E11882_01/server.112/e18951/toc.htm

Log in to the ASM instance as the grid user and issue the following commands (in this example, the diskgroup name is DATA. Substitute your own diskgroup name.)

```
[oracle@server bin]$ export ORACLE_SID=+ASM

[oracle@server bin]$ export ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid

[oracle@server bin]$ cd $ORACLE_HOME/bin

[oracle@server bin]$ ./sqlplus / as sysasm

SQL*Plus: Release 11.2.0.4.0 Production on Fri Nov 22 16:03:43 2013
```

```
Copyright (c) 1982, 2013, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Automatic Storage Management option
SQL> alter diskgroup DATA set attribute 'compatible.asm'='11.2.0.4';
Diskgroup altered.
SQL> alter diskgroup DATA set attribute 'compatible.rdbms'='11.2.0.4';
Diskgroup altered.
SQL> alter diskgroup DATA set attribute 'storage.type'='ZFSSA';
Diskgroup altered.
SQL>
```

A user may now create databases in the `+DATA` ASM diskgroup that are eligible to use the Hybrid Columnar Compression mode of compression.

Appendix D: References

See the following resources for additional information relating to the products covered in this document.

References to Sun ZFS Storage Appliance, Sun ZFS Storage 7000, and ZFS Storage Appliance all refer to the same family of Oracle ZFS Storage Appliance products. Some cited documentation may still carry these legacy naming conventions.

- Oracle ZFS Storage Appliance Documentation Library, including Installation, Analytics, Customer Service, and Administration guides:
  http://www.oracle.com/technetwork/documentation/oracle-unified-ss-193371.html
- The *Oracle ZFS Storage Appliance Administration Guide* is also available through the Oracle ZFS Storage Appliance help context.
  The Help function in Oracle ZFS Storage Appliance can be accessed through the browser user interface.
- Oracle ZFS Storage Appliance software releases can be downloaded from the following URLs:

  https://wikis.oracle.com/display/FishWorks/Software+Updates

  http://www.oracle.com/technetwork/server-storage/sun-unified-storage/downloads/index.html
- Oracle Support Center
  http://www.oracle.com/support
- Patches and updates downloads from My Oracle Support (MOS)
  (search under Oracle ZFS Storage Software Patches)
- Oracle ZFS Storage Appliance Plug-ins
  http://www.oracle.com/technetwork/server-storage/sun-unified-storage/downloads/zfssa-plugins-1489830.html
- Oracle Storage Product Information
  http://www.oracle.com/us/products/storage/overview/index.html
- Oracle ZFS Storage Appliance Technical White Papers and Solution Briefs,
  including "Best Practices for Oracle ZFS Storage Appliance and VMware vSphere5.x"
  http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/index.html

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

Optimizing Storage for Oracle Database 11*g*
Release 2 with the Oracle ZFS Storage Appliance
July 2014, Version 2.0
Author: Kristian Toms, Tom Connolly;
Update: Scott Ledbetter, Bryce Cracco

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