ORACLE
ZFS STORAGE APPLIANCE

Protecting Exadata Database Machine with the Oracle ZFS Storage Appliance

Configuration Best Practices

ORACLE WHITE PAPER | JULY 2015

ORACLE

Table of Contents

Executive Overview

Protecting the mission-critical data that resides on an Oracle Exadata Database Machine is a top priority. The Oracle ZFS Storage Appliance is ideally suited for this task due to superior performance, enhanced reliability, extreme network bandwidth, powerful features, simplified management and cost-efficient configurations. The Oracle ZFS Storage Appliance offers:

Extreme Network Bandwidth – With a highly scalable architecture that can be built around InfiniBand or 10GbE, the Oracle ZFS Storage Appliance provides the networking performance and redundancy that is required when connecting to an Exadata Database Machine.

ZFS-Enhanced Disk Reliability – Hardened ZFS features such as copy-on-write, metadata check summing, and background data scrubbing ensure data integrity, detect the presence of even silent data corruption, and correct errors before it is too late.

Powerful Features – Storage Analytics allow customers to quickly and effectively identify performance bottlenecks. Oracle Intelligent Storage Protocol (OISP) enables unique database-aware storage that simplifies administration and optimizes performance. Technologies such as replication, ZFS snap/clone and encryption provide solutions for any data protection or dev/test provisioning challenge that may arise. These are just a few of the enterprise-class features available in the Oracle ZFS Storage Appliance.

Simplified Management – An easy-to-use web management interface and integration within Oracle Enterprise Manager cut down on training expenses and reduce administration overhead. The Oracle ZFS Storage Appliance's innovative scalable storage pool and fast filesystem provisioning make managing storage extremely easy.

Oracle Optimized Compression – Hybrid Columnar Compression (HCC) is only available on Oracle storage. The Oracle ZFS Storage Appliance integrates with the Oracle Database to provide a full range of compression options that are optimized for specific data usage and workload patterns.

Superior Performance – The Oracle ZFS Storage Appliance is capable of database backup rates up to 42 TB/hr and restore rates up to 55 TB/hr. It set world records in both the SPC-2 and SPECsfs industry-standard benchmarks. Superior hardware and tighter integration enable backup and restore throughput that is much higher than competitor storage products.

The following graph shows the maximum sustainable backup and restore performance. Physical throughput rates were measured at the network level for backup and restore workloads between Exadata and the Oracle ZFS Storage Appliance. For detailed rates of the Oracle ZFS Storage ZS4-4 and ZS3-2 with specific storage configurations, please see the performance sizing sections later in this paper.

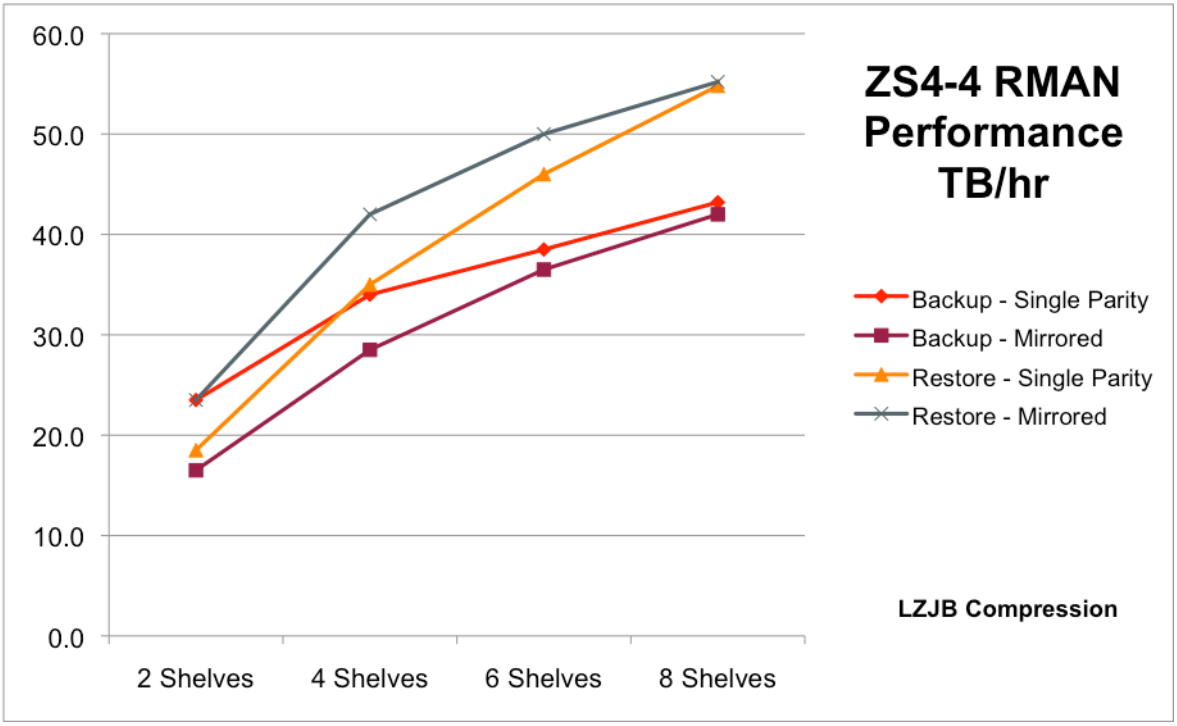

Figure 1: Maximum sustainable backup and restore throughput of an Oracle ZFS Storage ZS4-4

These are complete real world results using Oracle Database 12c and a large Online Transactional Processing (OLTP) database that is populated with sample customer data in a sales order entry schema. Advanced row compression is used at the database level to align with best practice recommendations for customers that are running OLTP workloads. These throughput rates were not obtained using a database or I/O generator test tool which can be misleading. They were not projected based on low-level system benchmarks. The backup and restore performance data collected for this document was measured using level 0 backup/restores of an otherwise idle database.

An Exadata Database Machine X5-2 full rack with an 80:20 DATA/RECO split and normal redundancy ASM DATA diskgroup is required to achieve some of these throughput rates such as the 55 TB/hr on restore. Alternatively, the same throughput can be achieved with multiple smaller Exadata configurations concurrently utilizing the same Oracle ZFS Storage ZS4-4. Full performance sizing details for smaller configurations are presented later in this document. When accounting for database level compression or

incremental backup strategies, effective backup rates that are much higher than the physical rates recorded in the previous chart are routinely observed.

Level 0 full database backups to an Oracle ZFS Storage ZS4-4 achieved a sustainable throughput rate of over 40 TB/hr. An effective backup rate of 40-80 TB/hr was attained for incremental backups with a daily change rate of 5-10 percent. With a mirrored storage profile, restore rates from an Oracle ZFS Storage ZS4-4 measured over 50 TB/hr. To put this into perspective, a 110 TB database which consumes over 250 TB of disk space when mirroring and temp are accounted for can be backed up in less than three hours using a level 0 backup or in nearly half that time when performing an incremental backup[1]. In the event of a failure, that same database can be restored in under 2.5 hours. Restoring a database this size could take days with competitive solutions. When recovery time objective (RTO) requirements are discussed, the difference between hours and days is huge. The extreme restore throughput capabilities of the Oracle ZFS Storage Appliance ensure that critical databases will be recovered and available as quickly as possible.

High performance is an important consideration when choosing a solution to protect an Oracle Exadata Database Machine. The following technologies make it possible for the Oracle ZFS Storage Appliance to achieve these backup and restore rates:

InfiniBand Support – The Oracle ZFS Storage Appliance can be configured with a highly redundant and scalable InfiniBand architecture. This allows for a seamless integration with the Oracle Exadata Database Machine and provides a high-bandwidth low-latency I/O path that generates relatively little CPU overhead.

RMAN Integration – Oracle Recovery Manager (RMAN) is a highly parallelized application that resides within the Oracle Database and optimizes backup and recovery operations. The Oracle ZFS Storage Appliance is designed to integrate with RMAN by utilizing up to 2000 concurrent threads that will distribute I/O across many channels spread across multiple controllers. This improves performance dramatically with sequential large-block streaming I/O workloads that are typical for most backup and restore situations.

Oracle Direct NFS – Oracle's optimized NFS client is an aggressive implementation that allocates individual TCP connections for each database process in addition to reducing CPU and memory overhead by bypassing the operating system and writing buffers directly to user space.

[1] Assumes a daily change rate of 5 percent.

One MB Record Sizes – The Oracle ZFS Storage Appliance now enables larger, 1 MB record sizes. This reduces the number of IOPS that are required to disk, preserves the I/O size from RMAN buffers to storage, and improves performance of large block sequential operations.

Hybrid Storage Pools – Oracle introduces an innovative Hybrid Storage Pool (HSP) architecture that utilizes dynamic storage tiers across memory, flash, and disk. The effective use of direct random access memory (DRAM) and enterprise-class software specifically engineered for multilevel storage is a key component that facilitates the superior performance of the Oracle ZFS Storage Appliance.

The Oracle ZFS Storage Appliance achieves world record setting throughput and is ranked number 1 in the price/performance metric of the independently audited SPC-2 industry standard benchmark. Combine this with the powerful features, simplified management, and Oracle-on-Oracle integrations and it is easy to see why it is a compelling solution for protecting mission-critical data on an Oracle Exadata Database Machine.

Introduction

Database, system, and storage administrators are faced with a common dilemma when it comes to backup and recovery of databases: how to back up more data, more often, in less time, and within the same budget. Moreover, practical challenges associated with real-world outages mandate that data protection systems be simple and reliable to ensure smooth operation under compromised conditions. The Oracle ZFS Storage Appliance helps administrators meet these challenges by providing a cost-effective and high-bandwidth storage system that combines the simplicity of the NFS protocol with ZFS-enhanced disk reliability. Through Oracle ZFS Storage Appliance technology, administrators can reduce the capital and operational costs associated with data protection while maintaining strict service level agreements with end customers.

The Oracle ZFS Storage Appliance is an easy-to-deploy unified storage system uniquely suited for protecting data contained in the Oracle Exadata Database Machine. With native InfiniBand (IB) and 10 gigabit (Gb) Ethernet connectivity, it is an ideal match for Oracle Exadata. These high-bandwidth interconnects reduce backup and recovery time, as well as reduce backup application licensing and support fees, compared to traditional network attached storage (NAS) storage systems. With support for both traditional tiered and incrementally updated backup strategies, Oracle ZFS Storage Appliance solutions deliver enhanced storage efficiency that can further reduce recovery time and simplify system administration.

When deploying an Oracle ZFS Storage Appliance for protecting the mission-critical data that resides on Exadata databases, backup window and recovery time objectives (RTO) must be met to ensure timely recovery in the event of a disaster. This paper describes best practices for setting up the Oracle ZFS Storage Appliance for optimal backup and recovery of Oracle databases and includes specific tuning guidelines for Oracle Exadata.

This paper addresses the following topics:

- Selecting a data protection strategy
- Configuring the Oracle ZFS Storage Appliance
- Configuring the Oracle Exadata Database Machine
- Best practices for traditional and incrementally updated backups
- RMAN backup/restore performance sizing for Oracle ZFS Storage ZS4-4 and ZS3-2

Selecting a Data Protection Strategy

Choosing the best backup strategy for your Oracle databases is an important step when building a data protection solution to meet your recovery time objectives (RTO), recovery point objectives (RPO), and version retention objectives (VRO). Most importantly, the backups need to be performed quickly and efficiently with no impact to end-user applications and with minimal resources consumed on production hardware. The type of backup strategy is also relevant when optimizing RMAN workloads with the Oracle ZFS Storage Appliance. Specific best practices are presented in the following chapters.

Traditional Backup Strategy

A traditional backup strategy is any strategy that uses unmodified full backups or any combination of unmodified level 0, level 1 cumulative incremental, and level 1 differential incremental backup to restore and recover any part of the database in the event of a physical or logical failure.

The simplest implementation of a traditional backup strategy is periodic full backup of the entire database. These backups can be performed while the database is open and active. Full backups are conducted on a user-defined schedule which could be weekly or daily. Database transactional archive logs should be multiplexed, with one copy stored directly on the Oracle ZFS Storage Appliance. These archive logs are used to recover a restored full backup and apply all transactions up until the time of the last online redo log switch before the failure. The RPO of this solution will never be more than 20 minutes, assuming that the redo logs are properly sized. The VRO would typically dictate that at least two full backups are kept active at all times with RMAN automatically expiring and eventually deleting older backups. Daily full backups may be ideal for a small database with a strict RTO.

A common implementation is a tiered approach that combines incremental level 0 and level 1 backup. Level 0 incremental backups are often taken on a weekly basis with level 1 differential or cumulative incremental backups performed daily.

RMAN block change tracking is used to improve the performance of incremental backup. The level 0 incremental backup scans the entire database but level 1 incremental backups use the block change tracking file to scan only the blocks that have changed since the last backup. This significantly reduces the amount of reads that are required on the database.

Here is an example of an RMAN traditional backup strategy that utilizes weekly level 0 backup combined with daily level 1 differential. Level 0 backups are equivalent in content to a full database backup. The differential means that each level 1 will only back up data that has been changed since the last level 0 or level 1 backup.

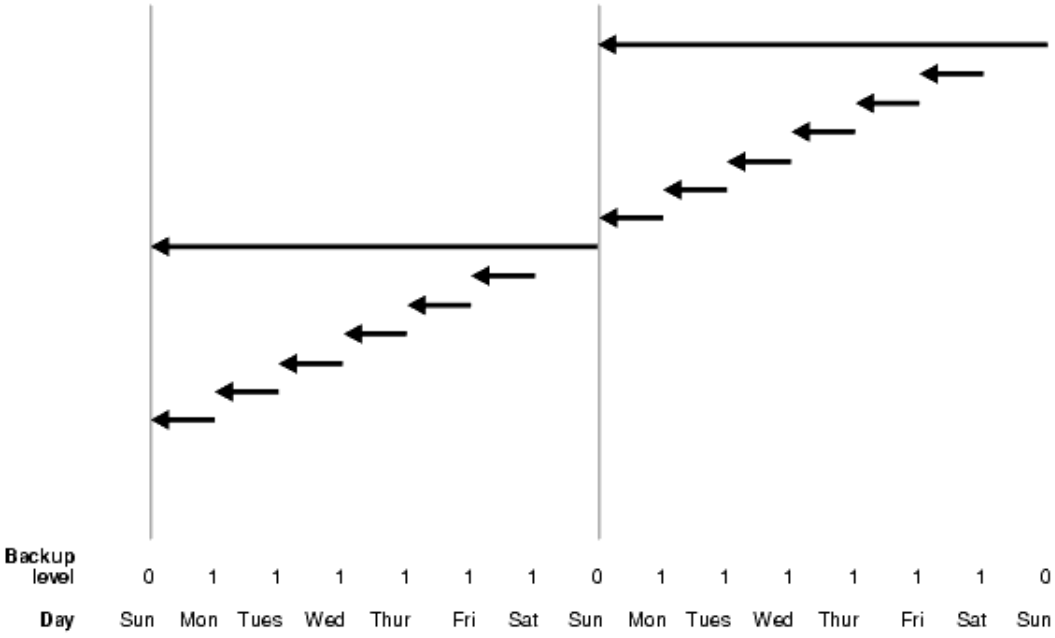

Figure 2: Traditional backup strategy with daily differential incremental backup

The VRO in this example specifies that two weeks of backup data is retained at all times. Level 0 incremental backups are performed every Sunday with level 1 differential incremental backups performed all other days. The level 1 differential incrementals are only backing up data that has changed within the last 24 hours. RMAN block change tracking is used so the backup operations performed Monday through Saturday are only scanning a small portion of the database and are only sending a small amount of data over the network for writing to disk. In this example, the archive logs are multiplexed with a copy stored directly on the Oracle ZFS Storage Appliance.

The database or part of the database can be restored and recovered to any point within the two-week span. RMAN will restore the most recent level 0 backup before the specified recovery point, restore all subsequent level 1 backups between the level 0 and the recovery point, and then apply the archive log transactions that are needed. Restoring data from backup is faster than applying transactions in archive logs. In this example, recovering from a failure on a Monday will be relatively fast and straightforward while recovering from a failure on a Saturday would be a longer process since five differential backups would need to be restored.

The next example utilizes weekly level 0 incremental backup combined with a mix of daily level 1 differential incremental and level 1 cumulative incremental. A cumulative incremental will include only data that has changed since the last level 0 backup. Cumulative incremental will consume more space than differentials but will streamline the restore process.

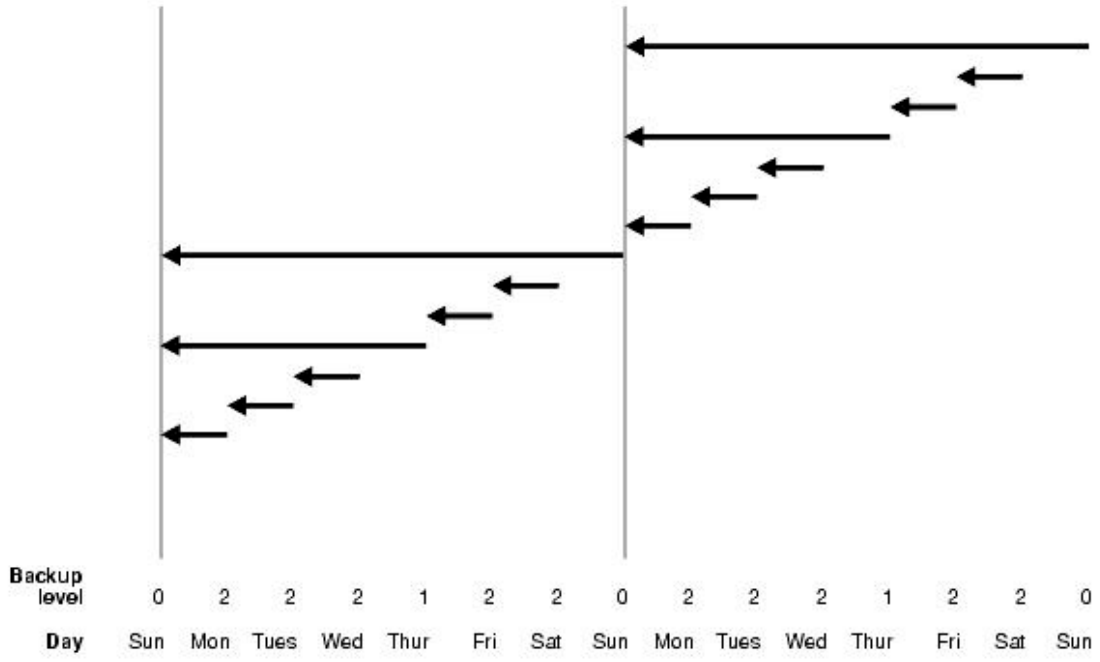

Figure 3: Traditional backup strategy with daily differential and cumulative incremental backup

The difference in this example is that the mid-week backups are now level 1 cumulative incremental. This makes for a larger Thursday backup but streamlines the restore process. For instance, if recovering from a failure that happened on a Friday, it would now only require restoring two backups.

A traditional tiered incremental backup strategy offers several advantages:

- Increased throughput rates – Up to 42 TB/hr backup and 55 TB/hr restore are possible with the Oracle ZFS Storage Appliance and traditional RMAN workloads due the ability to use large record sizes with parallel streaming I/O.
- Faster daily backups with less bandwidth consumption due to the use of block change tracking.
- Backups consume much less capacity than daily fulls due to incremental changed data approach.
- Synergies with tape archiving, native optimization for backup backupset.
- Bypasses unused datafile blocks, full multisection support.
- Second level 0 implicitly validates previous active level 0, no single point of failure,

Incrementally Updated Backup Strategy

An incrementally updated backup strategy creates an initial level 0 image copy backup. This is an identical image-consistent copy of the datafiles that is stored on the Oracle ZFS Storage Appliance. Subsequent backups are all differential incremental that only capture the changed data since the last backup. These are typically performed on a daily basis with the RMAN backupset containing only data that was changed

within the last 24 hours. Previous incremental backups are then applied to the image copy backup to roll it forward in time. This streamlines restores by providing a level 0 image-consistent copy that trails one or more days behind the active database. A visual representation of this process is displayed in the following figure.

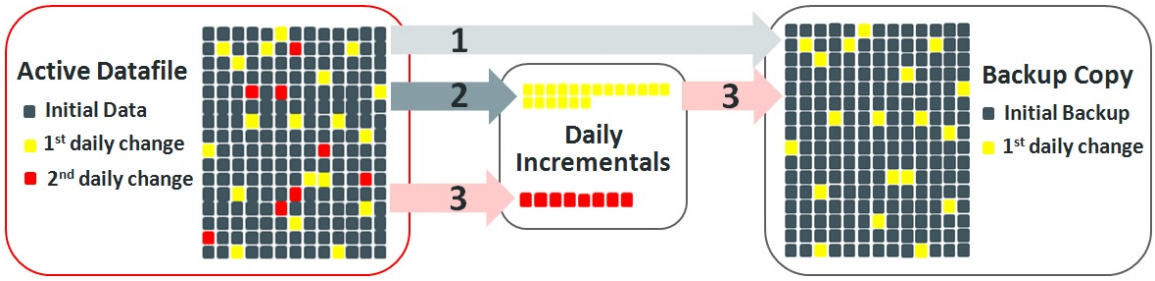

Figure 4: RMAN incrementally updated backup strategy

In this example, an identical RMAN job is run every night. On the first night, the job creates a level 0 image copy backup of all the datafiles in the database. This is represented by the grey blocks. On the second night, RMAN creates a backupset of only the data that was changed within the last 24 hours and stores these in a separate share. This is represented by the yellow blocks. On the third night, RMAN creates a backupset of only the data that was changed within the last 24 hours (red blocks) and then applies the previous night's changed data into the backup copy (yellow blocks). Every subsequent night that the backup is run it performs the same function of backing up just the data changed within the last 24 hours and applying previous changed data to the backup copy.

Just as with traditional RMAN strategies, incrementally updated backups can be performed with the database open and active. Archive logs should be multiplexed, with one copy directly stored on the Oracle ZFS Storage Appliance, ensuring that the RPO of the solution should never be more than 20 minutes.

An incrementally updated backup strategy offers several advantages:

- Limits the number of level 0 backups that are performed
- Simplifies the restore process in most situations
- Provides image copy synergies with provisioning for development and test
- Reduces disk space consumption, assuming only a single level 0 is maintained

Best Practices with the Oracle ZFS Storage Appliance

When protecting databases on Exadata with the Oracle ZFS Storage Appliance, a traditional RMAN backup strategy should be used in most situations.

RMAN traditional backup strategies benefit from technologies such as unused block skipping, null block compression, multisection support and multiple input files combined into a single backupset. RMAN traditional backup strategies exclusively utilize large streaming I/O which allows for 1M record sizes to be used on ZFS shares. RMAN uses non-shared buffers which means that less time is spent in a busy state waiting for another buffer to clear and per-channel throughput is higher. Level 0 backups complete faster and consume less space on disk. Throughput rates of restore operations are higher.

In an incrementally updated backup strategy, level 0 backups take longer to complete and consume more space on disk. Level 0 backups and merge operations that apply incremental changes to a level 0 backup utilize shared RMAN buffers with a reduced per-channel throughput and typically require that a smaller record size be used on ZFS shares. There is limited multisection support. Restore throughput rates are lower.

Additional benefits of traditional backup strategies are more flexibility in designing a backup strategy optimized to specific customer needs and standard integrations to seamlessly archive to tape. A lower price point can be achieved when configuring an Oracle ZFS Storage Appliance for a dedicated traditional RMAN backup use case since write-optimized flash is not required.

An incrementally updated backup strategy has advantages and should be used in situations where the source database is large and the daily change rate is very small. Since incremental update eliminates or reduces the need for level 0 backups on an ongoing basis and only changed data is sent over the network, there are significant advantages to implementing this backup strategy in these scenarios.

A general guideline is that an incrementally updated backup strategy should be considered when both of the following are true:

- The source database is large enough that a weekly level 0 backup could have a potential impact on network, disk, or server resources.
- The daily change rate of the database is 3 percent or less.

An incrementally updated backup strategy offers synergies with ZFS snapshot cloning for development and test provisioning, and can simplify the restore process. However, there is a potential for a single point of failure when maintaining only one level 0 backup, and most implementations provide only a narrow period of time that can be restored from backup. An n-1 trailing update is the standard.

Both incrementally updated and traditional strategies provide inline optimal deduplication for daily incremental backups. Only the changed data is transferred over the network. Both strategies satisfy demanding recovery point objectives; there should never be more than 20 minutes between the failure point and the most recent archive log backup.

Applying archive log transactions is more resource intensive than restoring backups. For databases with stringent recovery time objectives, a traditional incremental strategy may struggle to satisfy them if the failure happens to occur just before the next level 0 backup. In this case, multiple restores would be required, followed by applying redo transactions from the archive logs. In situations such as this, cumulative incremental backup can be substituted for differential to streamline the restore process.

The ability to customize backup strategies combined with the superior restore throughput of the Oracle ZFS Storage Appliance ensures that RTOs are met and exceeded.

Configuring the Oracle ZFS Storage Appliance

The following section provides best practices for optimizing an Oracle ZFS Storage Appliance to perform database protection in an Oracle Exadata environment.

Choosing a Controller

The Oracle ZFS Storage Appliance is available in two models: the ZS4-4 and ZS3-2.

Table 1: Oracle ZFS Storage Appliance Details

Features	ZS3-2	ZS4-4
CPU Cores	32	120
DRAM	512 GB or 1 TB	2 TB
Write Optimized Flash	4 TB	10.5 TB
Read Optimized Flash	12.8 TB	12.8 TB
Max Raw Capacity	1.5 PB	3.5 PB
HA/Cluster Option	Yes	Yes
Focus	Mid-Range	Scalability

The Oracle ZFS Storage ZS4-4 is the flagship product and offers maximum levels of scalability, CPU, and DRAM. With the potential to scale up to 2 TB of DRAM, 10 TB of write-optimized flash, and 12 TB of read-optimized flash, this is a highly scalable platform that can support up to 3.5 petabytes (PB) of raw storage capacity.

The Oracle ZFS Storage ZS3-2 is a cost-efficient model that can still achieve extremely high levels of throughput and redundancy. It can support up to 1 TB of DRAM, 4 TB of write-optimized flash, 12 TB of read-optimized flash and 1.5 PB of raw storage capacity.

Both of these models are excellent choices for protecting an Oracle Exadata Database Machine. When making the decision on which is best suited for your environment, there are a few factors to consider:

- Large sequential streaming workloads generally do not benefit from the presence of write- or read-optimized flash. Also, while DRAM is critical for achieving superior performance under these conditions, having an excessive amount of DRAM is unnecessary and will not further improve performance. If the Oracle ZFS Storage Appliance will be used 100 percent exclusively for traditional RMAN backup and restore workloads (large streaming I/O), then a system without write- or read-optimized cache is recommended.

- Write- and read-optimized cache is often recommended to achieve good performance and usability with most non-backup database I/O, incrementally updated backup workloads, cloning for dev/test provisioning, and many other mixed I/O scenarios. Having an Oracle ZFS Storage Appliance with a significant amount of write-optimized flash increases the flexibility for the type of workloads that it can be effectively used for. This may be important for current or future activity planning.

- If running direct transactional database workloads is a primary focus of the system, then having a large amount of DRAM and CPU resources will be a significant benefit.

- The Oracle ZFS Storage ZS3-2 offers exceptional throughput and redundancy at an extremely low price point. It is well suited for smaller configurations with two to four disk shelves but can still perform very well with larger configurations that focus on streaming I/O.

Choosing the Right Disk Shelves

The Oracle ZFS Storage Appliance offers two options for disk shelves, both with similar price points. The Oracle Storage Drive Enclosure DE2-24C features high-capacity 4 TB disks and the Oracle Storage Drive Enclosure DE2-24P features high-performance 900 GB disks. Each shelf contains 24 disks and both can be configured with the same write-optimized flash options (up to 4 disks per shelf can be replaced with SSD write flash accelerators). The Oracle ZFS Storage ZS4-4 and ZS3-2 can be customized based on disk shelf and write-optimized flash requirements.

Table 2: Oracle Storage Drive Enclosure Model (Disk Shelf) Details

Disk Shelf	Size/Disk	RPM	IOPS/Disk	MBPS/Disk	Rack Units
DE2-24P[2]	900 GB	10K	160	170	2
DE2-24C	4 TB	7.2K	120	180	4

The high-capacity Oracle Storage Drive Enclosure DE2-24C disk shelf is recommended when protecting an Oracle Exadata with an Oracle ZFS Storage Appliance. The larger capacity and slightly higher throughput give it a significant advantage for most backup use cases. The Oracle Storage Drive Enclosure DE2-24P may be considered in situations where the higher IOPS and lower latency would be a

[2] The Oracle Storage Drive Enclosure DE2-24P is also available with a 300 GB disk option which is not included here since it is not recommended for data protection use cases.

significant advantage or if rack space was a limiting factor and the desire was to maximize performance in a small partial-rack configuration.

Choosing a Storage Profile

When selecting a storage profile to protect an Oracle Exadata Database Machine, mirrored, single parity, and double parity are all worthy of consideration.

Table 3: Storage Profile Comparison

STORAGE PROFILE	USABLE CAPACITY[3]	ADVANTAGES	NEGATIVES
Mirrored	42.2%	Restore Performance Maximum Protection Maximum Flexibility	Costly
Single Parity	69.3%	Backup Performance Moderate Flexibility	Limited Redundancy
Double Parity	76.7%	Streaming Performance Most Efficient	Limited IOPS

Mirrored

Mirrored is a frequently recommended storage profile due to its strong redundancy and robust performance, particularly for restores. Because it generates twice as many virtual devices (vdevs) as a single parity implementation, a mirrored storage pool is capable of handling far more IOPS. This gives it the flexibility to perform well with large sequential I/O such as traditional RMAN workloads and also achieve exceptional performance with workloads that generate small random I/O such as direct database OLTP transactions.

Negatives of choosing a mirrored profiles are that it consumes more disk space than the other two options and generates more internal bandwidth on writes, which would be impactful in situations where SAS or PCI bandwidth are limiting factors.

Mirrored is recommended when there is an emphasis on achieving optimal performance, particularly restore performance, to provide the shortest RTO possible for databases running critical business apps.

[3] Usable capacity accounts for raw capacity lost due to parity, spares, and filesystem overhead as well as small amounts of space lost on each disk due to OS overhead, drive manufacturer overhead and scratch space reservations. This will vary slightly depending on the size of the storage pool; this example assumes a four disk shelf configuration.

It is also recommended in situations where incrementally updated backup strategies, cloning for dev/test provisioning, or direct database workloads are a focus. If the database runs an OLTP workload (characterized by mostly small transactions with a focus on changing and reading existing rows of data) or has an element of write transactions dispersed throughout the day, then this will generate small random I/O that will place an IOPS load on the storage pool with these use cases. A mirrored profile is best suited to handle heavy IOPS.

Single Parity

Single parity is a middle of the road option that is often recommended for data protection scenarios. It provides optimal backup performance for traditional RMAN workloads and is a particularly attractive option when usable capacity concerns would make mirrored a poor fit.

Single parity implements a narrow 3+1 stripe width and utilizes powerful ZFS features to provide exceptional performance with large streaming I/Os but also enough flexibility to handle some random or smaller I/O workloads.

Single parity is recommended when there is an emphasis on backup performance, when usable capacity concerns would make mirrored a poor fit, and when use cases such as incrementally updated backups, dev/test provisioning, and direct database workloads may be used sparingly but are not a primary focus. Because single parity uses a narrow stripe width, it still generates a moderate amount of vdevs (half as many as mirrored) and has flexibility to handle some workloads that generate non-streaming large I/O, but an IOPS-intensive workload such as an incrementally updated backup strategy for an OLTP database should utilize a mirrored profile for datafile copies.

Double Parity

Double parity provides the best usable capacity and will perform almost as well as single parity on large streaming I/O which is typical for traditional RMAN workloads. It accomplishes this by utilizing a wide stripe width. The width varies at the time of storage pool creation, depending on the number of disks in the configuration, but it ranges up to 14 disks. As a result, the number of vdevs in a double parity storage pool is far fewer than with mirrored or single parity. The ability to handle IOPS-intensive workloads is severely diminished.

Double parity is only recommended for situations where the Oracle ZFS Storage Appliance is 100 percent dedicated to large sequential workloads such as traditional RMAN backup and restore. It is not advisable if there is a possibility in the future of introducing additional use cases such as cloning for dev/test provisioning or utilizing an incrementally updated backup strategy. Mirrored or single parity profiles are more flexible for handling additional use cases that may result in heavier disk IOPS with lower latencies.

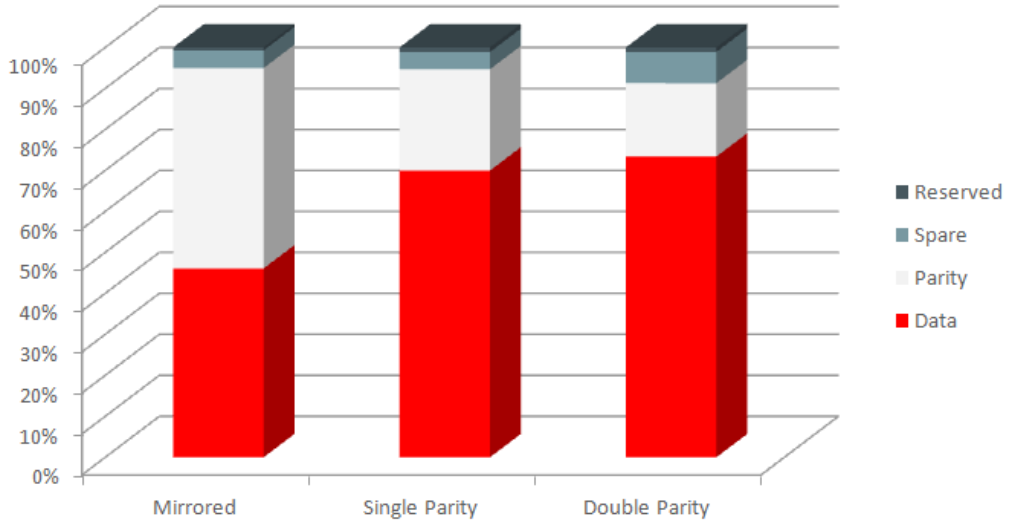

Figure 5: Raw disk capacity distribution

Configuring the Storage Pools

In most situations, it is recommended to configure a single storage pool on each controller. Each storage pool should be configured with half of the available hard disk drives (HDD) in each disk shelf. This allows for maximum performance and redundancy.

It is recommended to select the No Single Point of Failure (NSPF) option when configuring the storage pool. This will ensure that the loss of an entire disk shelf will not compromise the availability of data.

Using Write Flash Accelerators and Read-Optimized Flash

The Oracle ZFS Storage Appliance provides a unique, cost-effective, high-performance storage architecture that is built on a flash-first Hybrid Storage Pool model. A performance on demand approach allows optional write-flash accelerators and read-optimized flash devices to be configured into the storage pool.ool

Utilizing flash-based caching to unlock the power of Oracle's Hybrid Storage Pool architecture is critical to achieving optimal performance with transactional or mixed I/O workloads. However, traditional RMAN backup and restore workloads will not benefit from the presence of flash devices. These workloads generate streaming 1 MB I/Os, and datasets are very large (often greater than 1 TB). Throughput of the system during backups is typically determined by the rate at which data in memory can be synced to HDD. Additionally, RMAN backups are a low priority to populate level 2 read cache. The datasets are large and the frequency of restore is low.

Because solid state disk (SSD) is significantly more costly than HDD, flash devices are not recommended for Oracle ZFS Storage Appliance configurations that will be used exclusively for traditional RMAN workloads. In these environments, more benefit will be realized by adding additional disk shelves (HDD), which will increase performance and capacity. However, if database cloning for dev/test provisioning is a customer use case or if an incrementally updated backup strategy is implemented, then flash devices may be recommended or even required. The best practices sections for traditional RMAN workloads and incrementally updated backup strategies provide more detailed guidelines on when to use flash.

Selecting InfiniBand or 10GbE

The Oracle ZFS Storage Appliance can be configured with either InfiniBand or 10GbE for the purpose of protecting data that resides on an Exadata Database Machine.

Exadata utilizes a native InfiniBand infrastructure that provides increased bandwidth, lower latency, and reduced system resources utilization compared to 10GbE. The Oracle ZFS Storage Appliance can integrate seamlessly by connecting to the two InfiniBand leaf switches (NM2-36P) that are pre-configured in the Exadata rack.

In most situations, it is recommended to utilize InfiniBand when connecting to Exadata. The RMAN backup/restore throughput rates and performance sizing data presented in this document were collected in the lab using InfiniBand. However, in uncommon situations, 10GbE will make a better choice. Some examples of these situations include distance limitations that make InfiniBand deployments prohibitive or backing up five or more isolated Exadatas to a single Oracle ZFS Storage ZS4-4.

RMAN backup and restore throughput rates for 10GbE configurations are not measurably different for smaller configurations that are disk limited. Large configurations that are limited by system resources will see a small decrease in throughput of less than 10 percent when switching to 10GbE. Configurations that are network bandwidth limited to begin with and do not add additional active 10GbE links will see a more significant decrease in throughput.

When connecting the Oracle ZFS Storage Appliance to the Exadata InfiniBand fabric, any available ports on the two leaf switches in each Exadata rack can be utilized. Ports 5B, 6A, 6B, 7A, 7B, 8B, and 12A are available in Exadata Database Machine X5-2 full rack configurations. In partial rack configurations, additional ports are available.

> *To optimize availability and tolerate the loss of an InfiniBand switch, in typical configurations port 1 on each HCA should be connected to the lower leaf switch and port 2 should be connected to the upper leaf switch. Detailed information is documented in this AIE whitepaper: "Configuring an Oracle ZFS Storage Appliance with Multiple Oracle Exadata Database Machines."*

Choosing Oracle Direct NFS

Oracle Direct NFS (dNFS) is highly recommended for all database RMAN workloads between Exadata and the Oracle ZFS Storage Appliance. It is required to achieve optimal performance.

dNFS is a custom NFS client that resides within the database kernel and provides several key advantages:

- Significantly reduces system CPU utilization by bypassing the operating system (OS) and caching data just once in user space with no second copy in kernel space
- Boosts parallel I/O performance by opening an individual transmission control protocol (TCP) connection for each database process
- Distributes throughput across multiple network interfaces by alternating buffers to multiple IP addresses in a round robin fashion
- Provides high availability (HA) by automatically redirecting failed I/O to an alternate address

These features enable increased bandwidth and reduced CPU overhead.

No additional steps are required on the Oracle ZFS Storage Appliance to enable dNFS although it is recommended to increase the maximum number of NFS server threads from the default of 500. To do this, access the Oracle ZFS Storage Appliance BUI, select Configuration→Services→NFS, and set the number of threads to 1000.

Oracle Intelligent Storage Protocol

Oracle Intelligent Storage Protocol (OISP) was introduced with dNFS in the 12c version of Oracle Database. It enables database-aware storage by dynamically tuning record size and synchronous write bias on the Oracle ZFS Storage Appliance. This simplifies the configuration process and reduces the performance impact due to configuration errors. Hints are passed from the Oracle database kernel to the Oracle ZFS Storage Appliance. These hints are interpreted to construct a workload profile that is used to dynamically optimize storage settings.

> *OISP is an optional protocol. In the current implementation, a properly configured environment that adheres to the best practices in this document will perform equally well without OISP. It requires NFSv4 and SNMP. Reference My Oracle Support (MOS) Document 1943618.1 for instructions on how to enable OISP.*

Configuring IP Multipathing

IP multipathing groups (IPMP) are recommended to provide full HA redundancy. dNFS can provide a level of HA but currently relies on the kernel NFS mount for file opens or creates. IP multipathing is required to provide full HA in all situations.

This example assumes that two InfiniBand HCAs are installed in each ZFS controller. Configure IP multipathing with the following steps:

1. Create InfiniBand data links for ibp0, ibp1, ibp2, and ibp3; set partition key to ffff and link mode to connected mode.

2. Create network interfaces for ibp0, ibp1, ibp2, and ibp3; use the address 0.0.0.0/8 for each network interface.

3. Create the first IPMP network interface (ib-ipmp-controller1) using ibp0 and ibp3 with both ports set as active; create two different IP addresses for this link for optimized performance with dNFS. The number of IP addresses should match the number of active IB interfaces.

4. Create the second IPMP network interface (ib-ipmp-controller2) using ibp1 and ibp2 with port ports set as active; create two different IP addresses for this link for optimized performance with dNFS. The number of IP addresses should match the number of active IB interfaces. This IPMP group will be owned by the second controller. It can either be created directly using the BUI on the second controller or, if configuring in a takeover mode, it can be created on the first controller and ownership can be changed in the BUI Configuration→Cluster screen prior to a failback.

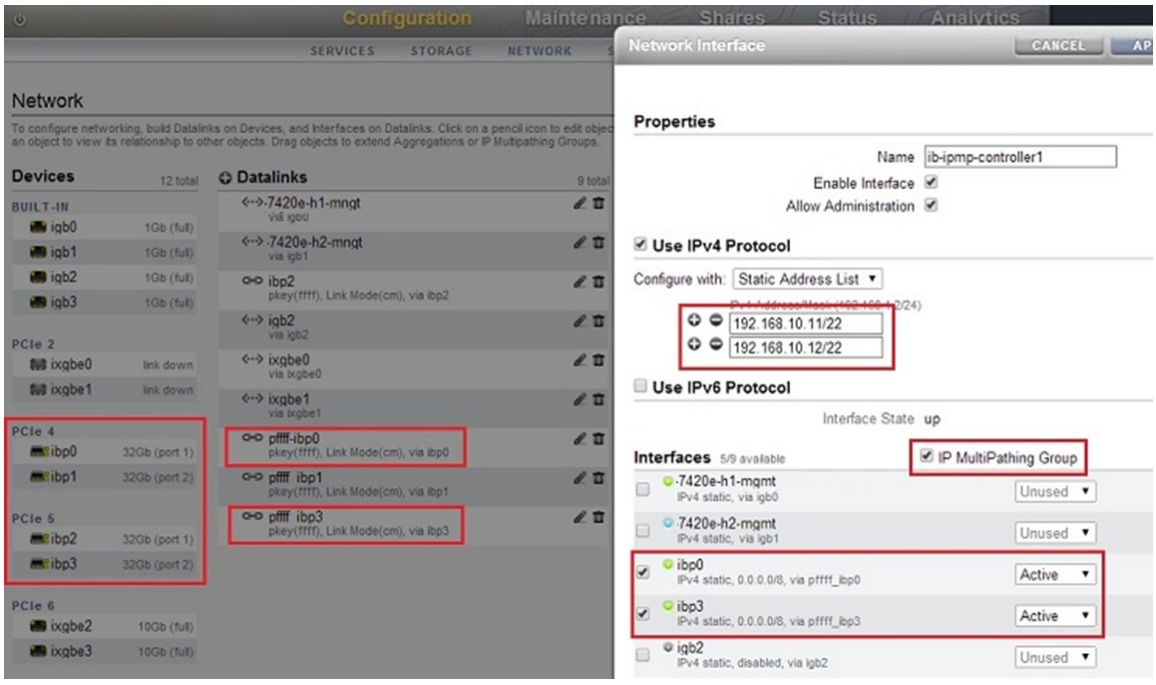

Figure 6: IPMP configuration with two HCAs per controller

Please note that active/active IPMP on the Oracle ZFS Storage Appliance requires as many IP addresses as active links to process traffic on all of the active links. To apply this in RMAN backup and restore applications you should use an oranfstab file to configure dNFS load spreading over multiple network interfaces.

If four InfiniBand or 10GbE cards are installed in each controller, then it is recommended to configure two IPMP network interfaces on each controller. Each IPMP group will include two active interfaces that are spread across different cards, PCI bridges, and network switches for optimal redundancy. Using two groups of two as opposed to one group of four reduces the overhead while still providing full HA redundancy.

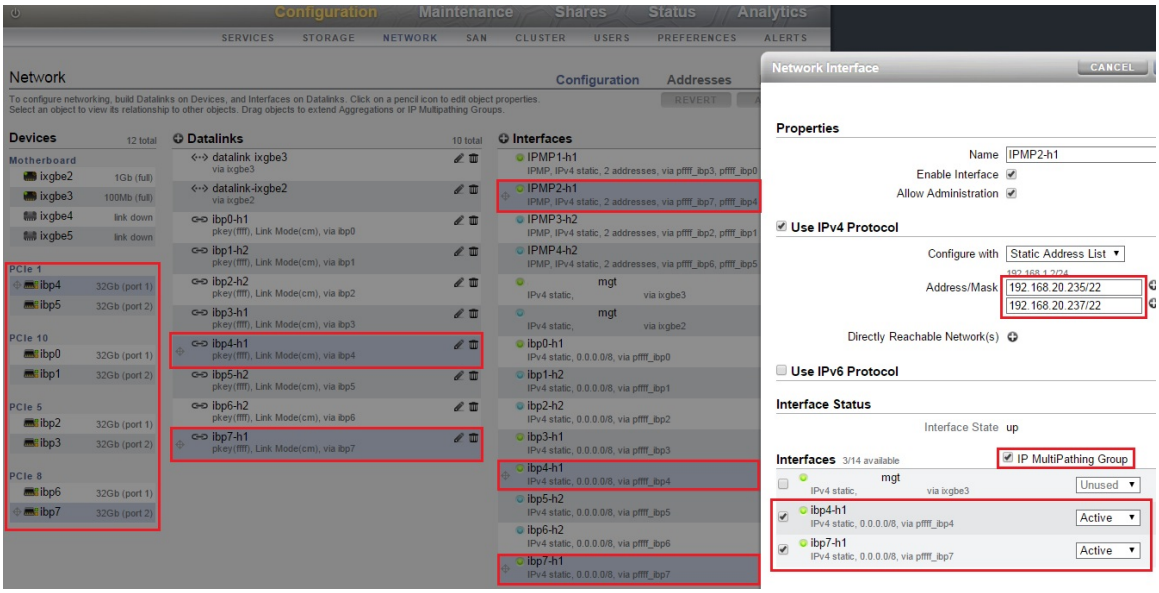

Figure 7: IPMP configuration with 4 HCAs per controller

The default IPMP failover settings, 10 second link failure detection and automatic failback, are not optimal for use with Exadata. When using IPMP with Exadata and Oracle ZFS Storage Appliance, change the link failure detection to 5000 ms (5 seconds) from the default of 10000 ms (10 seconds). This will match the Oracle ZFS Storage Appliance IPMP failover configuration to that of the Exadata. This can be done from the BUI or CLI as follows:

1. Select Configuration→Services→IPMP
2. Change Failover Detection Latency from 10000 ms to 5000 ms and apply

Enable adaptive routing for the multihoming policy when using IPMP to ensure that outbound traffic from the Oracle ZFS Storage Appliance is balanced over the network links and IP addresses. Access the BUI, select Configuration→Network→Routing, then select the multihoming=adaptive radio button.

Configuring the Oracle Exadata Database Machine

The following section provides guidance and options for configuring the Exadata Database Machine with Oracle ZFS Storage Appliance.

Choosing NFSv3 or NFSv4

NFSv3 and NFSv4 are both excellent protocol choices when using dNFS with Exadata and Oracle ZFS Storage Appliance. NFSv4 implements several enhancements, such as a stronger security model, file locking managed within the core protocol, and delegations which can help improve the accuracy of client side caching.

NFSv4 incurs a more significant overhead. However, dNFS workloads utilize a large packet size and any potential performance impact is negligible in this environment. NFSv4 is required, along with a 12c version of Oracle Database, to enable Oracle Intelligent Storage Protocol.

Enabling NFS on Exadata

When using only NFSv4, no additional steps are necessary prior to configuring and mounting the share(s) on Exadata database servers.

If support for NFSv3 or NFSv2 connectivity is desired, then additional remote procedure call (RPC), mounting and locking protocols are needed. To enable these services on Exadata versions based on Oracle Linux 5 (`cat /etc/redhat-release`) then portmap, nfslock and nfs services should be started and made persistent across reboots.

The following example uses the Exadata `dcli` command to enable portmap, nfslock and nfs services on all database servers.

```
# dcli -l root -g /home/oracle/dbs_group chkconfig portmap on
# dcli -l root -g /home/oracle/dbs_group service portmap start
# dcli -l root -g /home/oracle/dbs_group chkconfig nfslock on
# dcli -l root -g /home/oracle/dbs_group service nfslock start
# dcli -l root -g /home/oracle/dbs_group chkconfig nfs on
# dcli -l root -g /home/oracle/dbs_group service nfs start
```

If your Exadata version is based on Oracle Linux 6 (12.1.2.1.0 and above), then portmap has been replaced by rpcbind. Additionally, rpcbind requires read access to `/etc/hosts.allow and /etc/hosts.deny`.

The following example uses the Exadata `dcli` command to allow read access of these files and to enable rpcbind, nfslock and nfs services on all database servers.

```
# dcli -l root -g /home/oracle/dbs_group chmod 644 /etc/hosts.allow
# dcli -l root -g /home/oracle/dbs_group chmod 644 /etc/hosts.deny
# dcli -l root -g /home/oracle/dbs_group chkconfig rpcbind on
# dcli -l root -g /home/oracle/dbs_group service rpcbind start
```

```
# dcli -l root -g /home/oracle/dbs_group chkconfig nfslock on
# dcli -l root -g /home/oracle/dbs_group service nfslock start
# dcli -l root -g /home/oracle/dbs_group chkconfig nfs on
# dcli -l root -g /home/oracle/dbs_group service nfs start
```

NFS Mount Options

If shares are dedicated to traditional RMAN backup, utilize the following mount options:

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

> *If shares are utilized for incrementally updated backups or there is a potential to utilize RMAN switch to copy for immediate database recovery, utilize the following mount options:*
>
> *`rw,bg,hard,nointr,rsize=32768,wsize=32768,tcp,actimeo=0,vers=3,timeo=600`*

Reference My Oracle Support (MOS) Document 359515.1 "Mount Options for Oracle files when used With NFS on NAS devices" for a complete description of which NFS mount options to use with which type of Oracle files.

Oracle Direct NFS does not utilize NFS mount options. However, setting the proper mount options is recommended to be in compliance with database requirements and to improve performance and functionality if dNFS is not available and the system must revert to NFS.

It is required that the backup shares be mounted on the database nodes running the RMAN backup and restore jobs. However, it is recommended to mount the shares on all database servers so that every node can execute a backup or a restore operation. This is particularly beneficial in failure scenarios where database instances and RMAN backup services may be automatically migrated to other database servers that they are not normally active on.

Configuring Oracle Direct NFS

In Oracle Database 12c, dNFS is already enabled by default. In Oracle Database 11g, Oracle Direct NFS may be enabled on a single database node with the following command:

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

Exadata `dcli` may be used to enable dNFS on all of the database nodes simultaneously:

```
$ dcli -l oracle -g /home/oracle/dbs_group make -f $ORACLE_HOME/rdbms/lib/ins_rdbms.mk dnfs_on
```

The database instance should be restarted after enabling Oracle Direct NFS.

Confirm that dNFS is enabled by checking the database alert log for an Oracle ODM message after database startup:

```
Oracle instance running with ODM: Oracle Direct NFS ODM Library Version 3.0
```

dNFS activity can also be confirmed by SQL query:

```
SQL> select * from v$dnfs_servers;
```

dNFS may be disabled with the following command:

```
$ make -f $ORACLE_HOME/rdbms/lib/ins_rdbms.mk dnfs_off
```

For a complete list of recommended patches reference My Oracle Support (MOS) Document 1495104.1 "Recommended Patches for Direct NFS Client".

Creating an `oranfstab` File

The `oranfstab` file is required to achieve the published backup and restore rates. The file is created in `$ORACLE_HOME/dbs/oranfstab` and applies to all database instances that share the Oracle home.

The `oranfstab` file configures load spreading of dNFS connections over multiple addresses on the Oracle ZFS Storage Appliance (represented by "path") or multiple addresses on the Exadata database server (represented by "local"). Load balancing over multiple interfaces will reduce or eliminate two possible system bottlenecks: network interface bandwidth and TCP/IP buffering.

It is recommended to match the number of path IP addresses defined in the `oranfstab` with the number of active interfaces on the Oracle ZFS controller. Also, it is recommended to match the number of local IP addresses defined in the `oranfstab` with the number of active interfaces on the database server.

Older Exadata models such as Exadata Database Machine X3-2 or X2-2 have just a single active interface for datapath (bondib0) on each compute node.

However, newer models like Exadata Database Machine X4-2 and X5-2 have the ability to operate in an active/active role with independent IP addresses defined on both ib0 and ib1. Some models such as the X3-8 and X4-8 have four active datapath interfaces. In these scenarios, multiple local IP addresses need to be defined in the `oranfstab` to enable dNFS to utilize all available paths.

Here is an example of an `oranfstab` file for use with a clustered Oracle ZFS Storage Appliance hosting two storage pools, one per head and one share per pool. The shares are defined with the NFS export path on the Oracle ZFS Storage Appliance and the local mount point owned by the oracle user. Shares are listed beneath the path addresses that they are associated with. The Exadata is an X3-2 and each Oracle ZFS Storage controller is configured with two active interfaces. Note that the local address is specific to each Exadata compute node so `oranfstab` files on additional compute nodes would define a different local address. In this example dNFS utilizes NFSv3 which is the default.

```
server: zfs-storage-a
local: 192.168.10.1 path: 192.168.10.50
local: 192.168.10.1 path: 192.168.10.51
export: /export/backup1 mount: /zfs/backup1
server: zfs-storage-b
local: 192.168.10.1 path: 192.168.10.52
local: 192.168.10.1 path: 192.168.10.53
export: /export/backup2 mount: /zfs/backup2
```

Here is a similar example that uses an Exadata Database Machine X5-2 with active/active InfiniBand interfaces. Note that a different local address is defined so that the workload will be spread across all available paths. A second share named "copy" is created on each pool and presented to the dNFS client. In this example dNFS utilizes NFSv4 by defining an `nfs_version` parameter for each server.

```
server: zfs-storage-a
local: 192.168.10.1 path: 192.168.10.50
local: 192.168.10.2 path: 192.168.10.51
nfs_version: nfsv4
export: /export/backup1 mount: /zfs/backup1
export: /export/copy1 mount: /zfs/copy1
server: zfs-storage-b
local: 192.168.10.1 path: 192.168.10.52
local: 192.168.10.2 path: 192.168.10.53
nfs_version: nfsv4
export: /export/backup2 mount: /zfs/backup2
export: /export/copy2 mount: /zfs/copy2
```

Here is an example of an `oranfstab` file on an Exadata Database Machine X4-8 connected to a clustered Oracle ZFS Storage Appliance with only one active network interface configured on each controller. This oranfstab configuration will load spread across all four active interfaces on the Exadata compute node going to the single active path address on each controller.

```
server: zfs-storage-a
local: 192.168.10.1 path: 192.168.10.50
local: 192.168.10.2 path: 192.168.10.50
local: 192.168.10.3 path: 192.168.10.50
local: 192.168.10.4 path: 192.168.10.50
export: /export/backup1 mount: /zfs/backup1
server: zfs-storage-b
local: 192.168.10.5 path: 192.168.10.51
local: 192.168.10.6 path: 192.168.10.51
local: 192.168.10.7 path: 192.168.10.51
local: 192.168.10.8 path: 192.168.10.51
export: /export/backup2 mount: /zfs/backup2
```

Finally, here is an example of an oranfstab file on an Exadata X4-8 connected to a clustered Oracle ZFS Storage Appliance with four active network interfaces configured on each controller.

```
server: zfs-storage-a
local: 192.168.10.1 path: 192.168.10.50
local: 192.168.10.2 path: 192.168.10.51
local: 192.168.10.3 path: 192.168.10.52
local: 192.168.10.4 path: 192.168.10.53
export: /export/backup1 mount: /zfs/backup1
export: /export/stage1 mount: /zfs/stage1
server: zfs-storage-b
local: 192.168.10.5 path: 192.168.10.54
local: 192.168.10.6 path: 192.168.10.55
local: 192.168.10.7 path: 192.168.10.56
local: 192.168.10.8 path: 192.168.10.57
export: /export/backup2 mount: /zfs/backup2
export: /export/stage2 mount: /zfs/stage2
```

Note that if the local address is not specified in the `oranfstab` then dNFS will utilize the first routable address/interface it discovers.

Examples in this section are geared towards InfiniBand due to synergies connecting to the Exadata native InfiniBand infrastructure, but the `oranfstab` syntax is agnostic and will be applicable to 10GbE as well. For a complete reference to the options available in the `oranfstab` file, consult the administration guide for your specific version of Oracle Database software.

Configuring RMAN Backup Services

RMAN backup services should be created and used to balance RMAN workloads across all Exadata compute nodes. Spreading a backup across multiple Oracle RAC nodes will improve performance, increase parallel tasks, and reduce utilization load on any single component. RMAN backup services will be automatically migrated to other database servers in the Oracle RAC when the preferred instance is unavailable.

The following configuration steps assume an Exadata Database Machine X5-2 half rack with four Oracle RAC nodes. "Hulk" is the database name.

The syntax is: `srvctl add service -d <db_name> -r <preferred instance> -a <alternate instance(s)> -s <name for newly created service>`

```
[oracle@ex01db01 ~]$ srvctl add service -d hulk -r hulk1 -a hulk2,hulk3,hulk4 -s hulk_bkup1
[oracle@ex01db01 ~]$ srvctl add service -d hulk -r hulk2 -a hulk1,hulk3,hulk4 -s hulk_bkup2
[oracle@ex01db01 ~]$ srvctl add service -d hulk -r hulk3 -a hulk1,hulk2,hulk4 -s hulk_bkup3
[oracle@ex01db01 ~]$ srvctl add service -d hulk -r hulk4 -a hulk1,hulk2,hulk3 -s hulk_bkup4

[oracle@ex01db01 ~]$ srvctl start service -d hulk -s hulk_bkup1
[oracle@ex01db01 ~]$ srvctl start service -d hulk -s hulk_bkup2
[oracle@ex01db01 ~]$ srvctl start service -d hulk -s hulk_bkup3
[oracle@ex01db01 ~]$ srvctl start service -d hulk -s hulk_bkup4

[oracle@ex01db01 ~]$ srvctl status service -d hulk
Service hulk_bkup1 is running on instance(s) hulk1
Service hulk_bkup2 is running on instance(s) hulk2
Service hulk_bkup3 is running on instance(s) hulk3
Service hulk_bkup4 is running on instance(s) hulk4
```

When the database is restarted, the RMAN backup services should be rebalanced. This can be accomplished with:

```
[oracle@ex01db01 ~]$ srvctl stop service -d hulk; srvctl start service -d hulk
```

InfiniBand Best Practices

For optimal IPoIB performance and stability, modifications may be necessary on the Exadata compute nodes.

1. Implement the scatter gather feature.

- Enables non-contiguous memory allocations for InfiniBand buffering. This reduces latency in the IPoIB stack and eliminates the possibility of page allocation failures due to Linux memory fragmentation.
- Available and enabled by default when using both Exadata Storage Server Software 12.1.2.1.0 or later and Oracle ZFS Storage Appliance OS 8.2.7 (2013.1.2.7) or later
- Exadata software version 12.1.2.1.0 supports both 12c and 11g versions of the database. Due to an unrelated security enhancement, it was superseded by 12.1.2.1.1 which is the recommended minimum version.
- Confirm if the scatter gather feature is available on the Exadata database server.

  ```
  $ cat /sys/module/ib_ipoib/parameters/cm_ibcrc_as_csum
  ```

  1 = enabled for supporting end devices, 0 = disabled

2. Revert MTU of Exadata database servers to connected mode setting of 65520.
   - When compared to a setting of 7000, this means 8x fewer TCP packets to process, which reduces latency and queue congestion while improving performance and stability of IPoIB workloads.

3. Confirm Exadata database servers are using a IPoIB receive queue size of 2048.
   - With a default setting of 512 which was used on older Exadata versions, IPoIB receive queues can fill up and introduce significant latency to the IPoIB stack in the form of receiver not ready (RNR) messages which necessitate a retry delay.
   - Increased queue size of 2048 eliminates retry delays due to RNR and has a significant impact in avoiding IPoIB ordered queue congestion.
   - To check IPoIB receive queue size, execute:

     ```
     $ cat /sys/module/ib_ipoib/parameters/recv_queue_size
     ```

Preparing the Database for Backup

The following configurations are recommended when preparing the database for backup.

Archivelog Mode

Archiving of the online redo logs is enabled when the database is configured to operate in archivelog mode. Benefits of using archivelog mode include:

- Protection in the event of media failure.
- Ability to recover database transactions that occurred after the most recent backup.
- Backups can be performed while the database is open and active.
- Can use inconsistent backups to restore the database.

It is recommended that the database run in archivelog mode and the archivelogs be multiplexed with one copy on Exadata storage and one copy on the Oracle ZFS Storage Appliance.

Block Change Tracking

Block change tracking is an RMAN feature that records changed blocks within a datafile. The level 0 backup scans the entire datafile but subsequent incremental backups rely on the block change tracking file to scan just the blocks that have been marked as changed since the last backup.

It is recommended that block change tracking be enabled to improve performance for incremental backups. If the chosen backup strategy only includes full or level 0 backups, then block change tracking should not be enabled.

Best Practices for Traditional RMAN Backup

This section details the recommended configuration steps necessary to achieve optimal performance and functionality when using an RMAN traditional backup strategy to protect an Oracle Exadata Database Machine with an Oracle ZFS Storage Appliance. Best practices presented in this section deal specifically with traditional RMAN backup strategies while recommendations included in previous sections have general application.

Configuring the Network Shares

A single share per storage pool is recommended when backing up a database with a traditional RMAN strategy. A storage pool is typically configured on each controller for maximum performance and the ability to fully utilize hardware resources on both controllers. An Oracle database can then be backed up using two shares with one owned by each controller.

Alternatively, when backing up multiple databases concurrently or even multiple Oracle Exadata Database Machines, an individual RMAN backup operation can be configured to use just a single share owned by one controller with other RMAN backup workloads effectively utilizing the other controller. Full HA redundancy is still provided in this configuration with the ability to fail over storage resources to the other controller.

Access permissions of the shares should be aligned to match the user id of the oracle user and the group id of the dba group. A standard Exadata configuration will be 1001 and 1002 respectively. In most configurations the directory permissions will be set to rwxr-x---.

Record Size

The ZFS record size is a setting that is configured at the share level and influences the size of back-end disk I/O. Optimal settings depend on the network I/O sizes used by the application, in this case RMAN. Traditional RMAN workloads with Oracle Direct NFS generate large 1 M writes and reads at the network layer. In this case, a 1 M record size setting should be used. The ability to use large record sizes has significant advantages such as increased throughput performance, which is critical for bandwidth-intensive workloads. Other benefits include reduced utilization of controller CPU resources.

In recent years HDD capacities have grown as quickly as ever, yet the IOPS these disks are capable of delivering has leveled off. RMAN workloads often generate datasets on the TB scale with only a small frequency of read back. As such, caching is not an optimal solution for handling IOPS. Maximizing the throughput and limiting the IOPS to disk is an important factor for achieving the best performance from the backup solution. RMAN traditional backup strategies enable this by delivering large multichannel network I/O which benefits greatly from large record sizes on the ZFS share.

Synchronous Write Bias

Synchronous write bias is a share setting that controls behavior for servicing synchronous writes. It can be optimized for latency or throughput.

All writes are initially written to the ZFS adaptive replacement cache (ARC) regardless of whether they are asynchronous, synchronous, latency-optimized, or throughput-optimized. Also, all writes are copied from the ARC to the storage pool. An asynchronous write returns an acknowledgement to the client after the write to ARC completes. When synchronous writes are optimized for throughput, an acknowledgement is not returned until the write is copied to the storage pool. When optimized for latency, an additional copy is written to persistent storage so that acknowledgements can be returned to the client faster. When write-optimized flash is configured in the storage pool it is used as the persistent storage for latency-sensitive synchronous writes.

The synchronous write bias share setting should be configured to throughput. There are two reasons:

1. Write optimized flash is a limited resource and is much more expensive than HDD. It provides a major boost to latency-sensitive writes, but traditional RMAN backups are bandwidth-sensitive, large 1 M streaming writes. If flash is not present in the storage pool configuration, a lower price point can be achieved by not adding it. HDD can be used to fill the slots that flash may occupy and provide additional throughput and capacity in the process. If write optimized flash is configured in the storage pool, it is a shared resource that other network shares and LUNs can access and it should be reserved for latency-sensitive workloads where it can provide benefit.

2. Setting the synchronous write bias to latency generates additional data transfer and will reduce performance when processing bandwidth-sensitive workloads. When optimized for latency, an additional copy is read from ARC and written to flash. When log devices are mirrored, an additional two copies are written to flash. Write optimized flash devices are designed to service a lot of IOPS but can easily become bandwidth saturated with high throughput workloads. Even if there are many idle flash devices configured in the storage pool and there is an adequate amount of flash bandwidth available, SAS bandwidth will become a limiting factor. In a configuration with mirrored storage pools, mirrored log devices and synchronous write bias set to latency, synchronous writes are written first to ARC and then four more copies are written. SAS bandwidth would be four times larger than network bandwidth. Throughput optimized writes will generate better performance for bandwidth-sensitive workloads.

Read Cache

Read optimized flash should not be used for caching traditional RMAN workloads since there is little benefit from storing RMAN backupsets in cache. Moreover, the level 2 ARC is not intended for streaming workloads. The cache device usage share setting should be configured to not use cache devices.

Data Compression

The data compression setting determines whether any compression algorithms are applied by the Oracle ZFS Storage Appliance at the share level. Compression is most effective at the database level and the best

practice is to use Hybrid Columnar Compression (HCC) for read-focused transactional workloads and Advanced Row Compression for write-focused transactional workloads.

For traditional RMAN workloads, LZJB compression should be enabled at the share level. It provides additional benefit when combined with database compression by reducing the bandwidth to back-end disk with only a minimal impact to CPU utilization. Physical network throughput is actually increased when using LZJB because SAS bandwidth and HDD utilization are typically limiting factors for a traditional RMAN workload. With Advanced Row Compression enabled for an OLTP database, LZJB often provides additional space savings in the range of 1.4 to 2.1x. Gzip based compression algorithms are costly on CPU overhead for high-performance workloads and should not be considered in this context.

Figure 8 shows a backup share that was created using the settings discussed in this section.

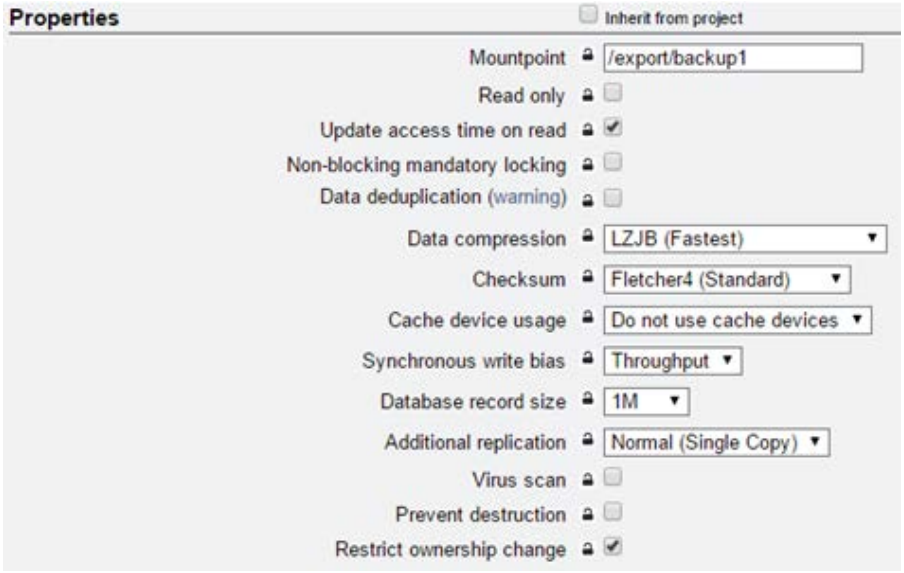

Figure 8: Example share for traditional RMAN backup

RMAN Configuration

The following factors should be considered for traditional RMAN backup configuration.

Backup Format

The level 0 backups can be either backupset or image copy format; however, backupset is recommended. Backupset format is required to achieve many of the benefits highlighted in the section on selecting a data protection strategy.

Optimizing Channels

Determining the number of RMAN channels to use is an important aspect of tuning a backup solution. When RMAN opens a new channel, it allocates a new set of input and output buffers. Each channel has the ability to take a datafile or a section of a datafile and process the backup or restore job in parallel to work being done by other channels. Channels can be assigned to different nodes in the Oracle RAC and can have different backup destinations with shares potentially owned by different controllers of the Oracle ZFS Storage Appliance.

> *Additional channels will increase scalability and can provide significantly improved performance, more efficient resource utilization, load balancing across the database nodes, a more robust HA architecture, and workload spreading between storage controllers.*

As hardware limits are approached, allocating additional RMAN channels will provide diminishing returns. It is not recommended to over-allocate channels because there will be little to no performance gain despite additional memory and CPU resources being allocated for more RMAN buffers and added complexity in the form of more backup pieces being created.

Determining the recommended number of channels for a particular configuration depends on the hardware factor that will limit overall performance in an optimally configured solution. Performance limiting components could be many things, including Exadata, network, HDD, CPU or SAS resources. Thorough testing is always recommended when implementing major changes in a production environment. However, the following matrix provides guidance for how many RMAN channels to configure in a traditional RMAN backup strategy depending on hardware configuration. This table assumes an Exadata Database Machine X5-2 and an Oracle ZFS Storage ZS4-4 with storage balanced across both controllers. It assumes that network and SAS bandwidth are not limiting factors, that the best practices in this document are implemented, and that there are no other significant concurrent workloads during the backup window.

Table 4: Traditional RMAN backup strategy; suggested RMAN channels per configuration

	Eighth Rack	Quarter Rack	Half Rack	Full Rack
1 Disk Shelf	8	8	8	8
2 Disk Shelves	8	12	12	12
3-4 Disk Shelves	8	16	16	16
5-6 Disk Shelves	8	16	24	24
7-8 Disk Shelves	8	16	32	32
9+ Disk Shelves	8	16	32	40

When RMAN channels are configured or allocated, they should be alternated across the Oracle RAC nodes and storage shares.

Tuning Buffers

A certain number of input buffers and output buffers are allocated to each channel. During a backup these memory buffers are used to read sections of a datafile, copy to an output buffer, and write to a backupset piece on a backup share.

RMAN jobs running from an ASM diskgroup will have buffers dynamically tuned. However, the following settings should be used to achieve optimal restore throughput. Backup throughput will remain unchanged.

```
_backup_disk_bufcnt=20

_backup_disk_bufsz=2097152
```

Section Size

Enabling highly parallel RMAN workloads is critical for achieving optimal performance and resource utilization from the backup solution. One challenge is when a very large datafile is encountered. If processed by a single RMAN channel, throughput will slow significantly and other hardware resources in the environment will sit idle waiting for the outlier datafile to complete.

RMAN has provided a solution to this problem with the ability to break large files up into smaller pieces that can be processed in parallel by multiple channels. This is called multisection support and is determined by the section size parameter. It is recommended to set the section size to 100 G.

Filesperset

The filesperset parameter determines how many datafiles or sections of datafiles are included in each backupset. When multiple input files are read to create a single backupset, it can improve performance, particularly when the read or copy phases are limiting factors. The default is 64; however, this is detrimental for single file or partial database restores where the entire backupset needs to be read back even though only a small section will be used. Also, an excessively large filesperset can impact the load balancing and performance scaling properties of RMAN. The objective is to have all RMAN channels effectively utilized throughout the backup. If there are a limited number of datafiles or data sections it may not be possible to create full backupsets on every channel.

As a general practice, it is recommended to set the filesperset to 3. Testing has shown that this strikes a good balance between handling partial database restores, load balancing across all channel,s and still optimizing backup performance with multiple input files feeding each backupset.

Sample Run Block

Here is a sample run block for a weekly level 0 backup that can be included as part of an incremental backup strategy. This example assumes an Exadata Database Machine X5-2 half rack backing up to both

controllers of an Oracle ZFS Storage Appliance configured with four disk shelves. RMAN backup services are used to evenly spread channels across all four RAC nodes. Channels are alternated between the two storage shares with one owned by each controller.

```
run
{
sql 'alter system set "_backup_disk_bufcnt"=20 scope=memory';
sql 'alter system set "_backup_disk_bufsz"=2097152 scope=memory';
allocate channel ch1 device type disk connect 'sys/passwd@exa-scan/bkup_db1' format '/zfs/bkup1/%U';
allocate channel ch2 device type disk connect 'sys/passwd@exa-scan/bkup_db2' format '/zfs/bkup2/%U';
allocate channel ch3 device type disk connect 'sys/passwd@exa-scan/bkup_db3' format '/zfs/bkup1/%U';
allocate channel ch4 device type disk connect 'sys/passwd@exa-scan/bkup_db4' format '/zfs/bkup2/%U';
allocate channel ch5 device type disk connect 'sys/passwd@exa-scan/bkup_db1' format '/zfs/bkup1/%U';
allocate channel ch6 device type disk connect 'sys/passwd@exa-scan/bkup_db2' format '/zfs/bkup2/%U';
allocate channel ch7 device type disk connect 'sys/passwd@exa-scan/bkup_db3' format '/zfs/bkup1/%U';
allocate channel ch8 device type disk connect 'sys/passwd@exa-scan/bkup_db4' format '/zfs/bkup2/%U';
allocate channel ch9 device type disk connect 'sys/passwd@exa-scan/bkup_db1' format '/zfs/bkup1/%U';
allocate channel ch10 device type disk connect 'sys/passwd@exa-scan/bkup_db2' format '/zfs/bkup2/%U';
allocate channel ch11 device type disk connect 'sys/passwd@exa-scan/bkup_db3' format '/zfs/bkup1/%U';
allocate channel ch12 device type disk connect 'sys/passwd@exa-scan/bkup_db4' format '/zfs/bkup2/%U';
allocate channel ch13 device type disk connect 'sys/passwd@exa-scan/bkup_db1' format '/zfs/bkup1/%U';
allocate channel ch14 device type disk connect 'sys/passwd@exa-scan/bkup_db2' format '/zfs/bkup2/%U';
allocate channel ch15 device type disk connect 'sys/passwd@exa-scan/bkup_db3' format '/zfs/bkup1/%U';
allocate channel ch16 device type disk connect 'sys/passwd@exa-scan/bkup_db4' format '/zfs/bkup2/%U';
configure snapshot controlfile name to '/zfs/bkup1/snapcf_dbname.f';
BACKUP AS BACKUPSET
  SECTION SIZE 100G
  INCREMENTAL LEVEL 0
  DATABASE
  FILESPERSET 3
  TAG 'BKUP_SUNDAY_L0';
}
```

Best Practices for Incrementally Updated Backup

This section details the recommended configuration steps necessary to achieve optimal performance, reliability, and usability when utilizing an RMAN incrementally updated backup strategy to protect an Oracle Exadata Database Machine with an Oracle ZFS Storage Appliance. Best practices presented in this section deal specifically with an incrementally updated backup strategy. It is intended to be used as supplementation for the technological details and general best practices that are fully described in previous sections.

Configuring the Oracle ZFS Storage Appliance

It is recommended to use mirrored storage profiles and SSD write flash accelerators for an incrementally updated backup strategy. The process where the previous incremental level 1 backupset is merged into the image copy creates a large amount of small block random I/O for databases running OLTP workloads. Mirrored storage pools are well suited for handling IOPS-intensive workloads. Small I/O updates to the backup copy are latency sensitive and require write-optimized flash present in the storage pool for optimal performance.

Creating Shares

Two shares are recommended when backing up a database with an incrementally updated backup strategy. The initial image copy backup should be placed in one share and the daily incremental in the other.

Access permissions of the shares should be aligned to match the user id of the oracle user and the group id of the dba group. A standard Exadata configuration will be 1001 and 1002 respectively. In most configurations the directory permissions will be set to rwxr-x---.

Record Size

Reads and writes involving the daily incremental share will generate large 1 M I/O. It is recommended to configure the ZFS record size of this share to 1 M.

Updates to the backup copy files will generate smaller writes since it is only updating blocks within the backup datafile that have changed on the active datafile. Determining the optimal ZFS record size for the copy share depends on the characteristics of the transactional workload to the source database.

If the source database is running an online transactional processing (OLTP) workload, the ZFS record size should be set to 32 K. An OLTP workload is characterized by mostly small transactions with a focus on changing and reading existing rows of data. This generates relatively small write I/Os that can be dispersed throughout the datafile. Setting the ZFS record size to 32 K minimizes read-modify-write overhead while still providing good performance for streaming workloads such as the initial backup and subsequent restore or restore validate operations.

If the source database is running an online application processing (OLAP) workload the ZFS record size should be set to 128 K. An OLAP workload is characterized by large queries and batch appends which generate large write I/Os when updating a backup copy.

Synchronous Write Bias

Synchronous write bias is a share setting that controls behavior for servicing synchronous writes. It can be optimized for latency or throughput.

The synchronous write bias should be configured to throughput for the daily incremental share and configured to latency for the copy share. Applying changes to the backup copy is a latency-sensitive workload that requires I/O to be copied to write-optimized flash so that a faster acknowledgement is returned to the client. This significantly improves the IOPS capabilities of the share and the overall performance of the solution.

Read Cache

Read optimized flash is not required but is recommended to optimize performance during the update phase in an incrementally updated backup strategy. The cache device usage share setting on the copy share should be configured for all data and metadata.

Data Compression

LZJB compression is only recommended on the daily incremental share. The copy share should have no ZFS share-level compression enabled.

Configuring the RMAN Environment

The following factors should be considered for traditional RMAN backup configuration.

Backup Format

In an incrementally updated backup strategy, the level 0 backup has to be in image copy format and daily incremental are always backupsets. There are no choices to be made here.

Optimizing Channels

Channels allocated for an image copy operation have a lower per-channel throughput capacity than channels allocated for a backupset operation because they use shared buffers for input and output functions. RMAN spends more time in a wait state where the shared buffer has not cleared its busy status. This does not impact the overall performance of the backup solution in environments that can properly scale to use many channels spread across multiple hardware platforms, but more channels may be required to achieve the same amount of throughput as a backupset operation.

Tuning Buffers

A certain number of input buffers and output buffers are allocated to each channel. During a backup these memory buffers are used to read sections of a datafile, copy to an output buffer, and write to a backupset piece on a backup share.

These settings should be used to achieve optimal throughput for incrementally updated backups:

```
_backup_disk_bufcnt=64

_backup_disk_bufsz=1048576

_backup_file_bufcnt=64

_backup_file_bufsz=1048576
```

Section Size

Multisection support for image copies was introduced in Oracle Database 12c. When running previous version of database software, using image copy format to backup bigfile tablespaces is not recommended because RMAN performance scaling and load balancing cannot be assured. This is not a concern with standard tablespaces since multiple datafiles will guarantee optimal performance scaling and load balancing across all RMAN channels.

Sample Run Block

Here is an example run block that is designed to be run repeatedly every time a backup of the database is performed. It uses a two-share, incrementally updated backup strategy and will place level 0 image copies of each file in bkup1 and subsequent level 1 incremental backups in bkup2. The same run block is executed every night, but it performs different tasks depending on the existence of prior level 0 and level 1 backups.

Upon initial execution, it creates a level 0 image copy backup of all the datafiles in the database. On the second night, it creates a backupset of only the data that was changed within the last 24 hours and stores these in a separate share. The third night, it creates a backupset of only the data that was changed within the last 24 hours and then applies the previous night's changed data into the backup copy. Every subsequent night that the backup is run, it performs the same function of backing up just the data changed within the last 24 hours and applying previous changed data to the backup copy. The RMAN tag command is used to track this.

This example assumes a half rack Exadata with four database servers. RMAN backup services are used in the connect string to spread the channels across all available nodes in the Oracle RAC.

```
RUN
{
sql 'alter system set "_backup_disk_bufcnt"=64 scope=memory';
sql 'alter system set "_backup_disk_bufsz"=1048576 scope=memory';
sql 'alter system set "_backup_file_bufcnt"=64 scope=memory';
sql 'alter system set "_backup_file_bufsz"=1048576 scope=memory';
allocate channel ch1 device type disk connect 'sys/passwd@exa-scan/bkup_db1' format '/zfs/bkup1/%U';
allocate channel ch2 device type disk connect 'sys/passwd@exa-scan/bkup_db2' format '/zfs/bkup1/%U';
allocate channel ch3 device type disk connect 'sys/passwd@exa-scan/bkup_db3' format '/zfs/bkup1/%U';
allocate channel ch4 device type disk connect 'sys/passwd@exa-scan/bkup_db4' format '/zfs/bkup1/%U';
allocate channel ch5 device type disk connect 'sys/passwd@exa-scan/bkup_db1' format '/zfs/bkup1/%U';
allocate channel ch6 device type disk connect 'sys/passwd@exa-scan/bkup_db2' format '/zfs/bkup1/%U';
allocate channel ch7 device type disk connect 'sys/passwd@exa-scan/bkup_db3' format '/zfs/bkup1/%U';
allocate channel ch8 device type disk connect 'sys/passwd@exa-scan/bkup_db4' format '/zfs/bkup1/%U';
allocate channel ch9 device type disk connect 'sys/passwd@exa-scan/bkup_db1' format '/zfs/bkup1/%U';
allocate channel ch10 device type disk connect 'sys/passwd@exa-scan/bkup_db2' format '/zfs/bkup1/%U';
allocate channel ch11 device type disk connect 'sys/passwd@exa-scan/bkup_db3' format '/zfs/bkup1/%U';
allocate channel ch12 device type disk connect 'sys/passwd@exa-scan/bkup_db4' format '/zfs/bkup1/%U';
allocate channel ch13 device type disk connect 'sys/passwd@exa-scan/bkup_db1' format '/zfs/bkup1/%U';
allocate channel ch14 device type disk connect 'sys/passwd@exa-scan/bkup_db2' format '/zfs/bkup1/%U';
allocate channel ch15 device type disk connect 'sys/passwd@exa-scan/bkup_db3' format '/zfs/bkup1/%U';
allocate channel ch16 device type disk connect 'sys/passwd@exa-scan/bkup_db4' format '/zfs/bkup1/%U';
configure device type disk parallelism 16;
RECOVER COPY OF DATABASE
  WITH TAG 'incr_zfs';
BACKUP
   INCREMENTAL LEVEL 1
  FOR RECOVER OF COPY WITH TAG 'incr_zfs'
  DATABASE format '/zfs/bkup2/%U';
}
```

Oracle ZFS Storage ZS4-4 RMAN Performance Sizing

The following graph shows sustainable throughput attained in Application Integration Engineering (AIE) during RMAN backup and restore to an Oracle ZFS Storage ZS4-4. These rates were collected running level 0 backups as part of a traditional RMAN backup strategy. The graph demonstrates maximum sustainable throughput for a variety of Oracle ZFS Storage ZS4-4 configurations ranging from two to eight disk shelves.

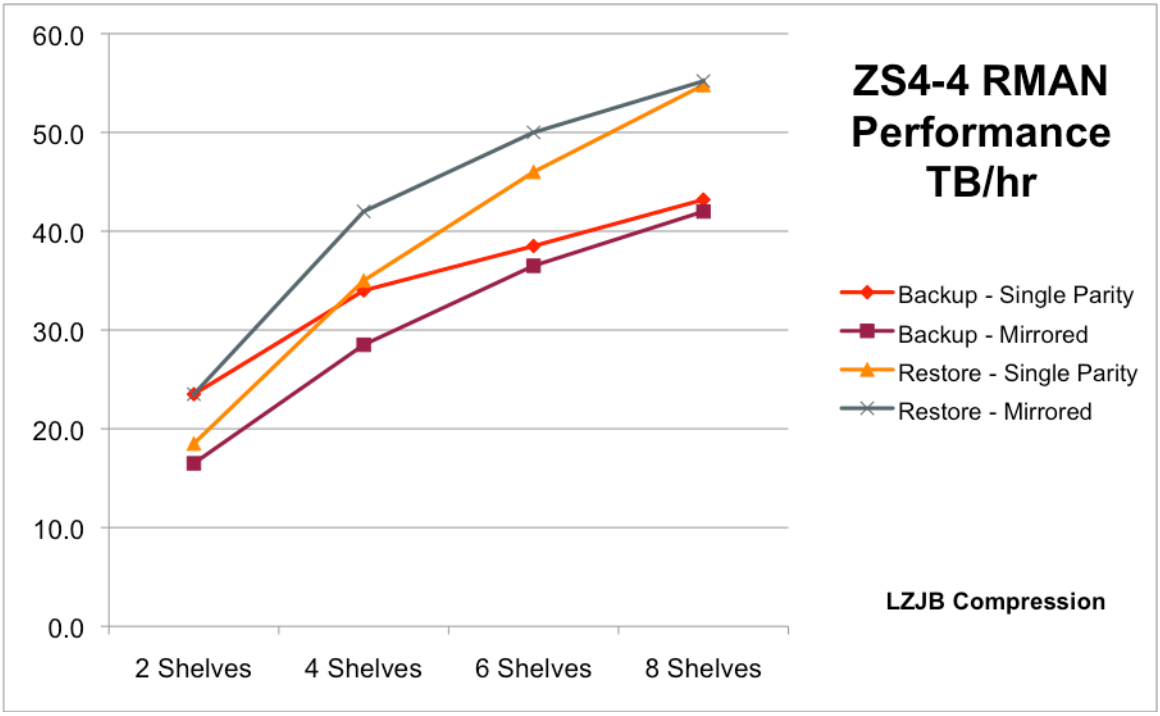

Figure 9: Oracle ZFS Storage ZS4-4 RMAN Throughput with ZFS LZJB Compression

These are complete real-world results using Oracle Database 12c and a large OLTP database that is populated with sample customer data in a sales order entry schema. Advanced row compression is used at the database level since this is the best practices recommendation for customers that are running OLTP workloads. This is a fully functional database with the ability to run live transactional workloads before, during or after an RMAN backup operation. These throughput rates were not obtained using a database or I/O generator test tool which can be misleading and only indirectly applicable to real world use cases. They were not projected based on low level system benchmarks.

In all scenarios, sustainable throughput is determined by measuring physical I/O at the network layer. An average is collected over an extended period of time. These graphs demonstrate the maximum Oracle Database backup and restore rates that the Oracle ZFS Storage ZS4-4 has been proven capable of sustaining in a real-world Exadata backup environment. Maximum sustainable rates of 55 TB/hr restore and 42 TB/hr backup were demonstrated. All of these rates can be achieved with an Exadata Database Machine X5-2 full rack with an 80:20 split between DATA and RECO and normal ASM redundancy for

the DATA diskgroup. Performance was collected using an Exadata and Oracle ZFS Storage ZS4-4 that were both otherwise idle during the Oracle Recovery Manager operation. Test environments were configured following the best practices presented in this white paper.

This next graph shows sustainable throughput attained with no ZFS share-level compression in use.

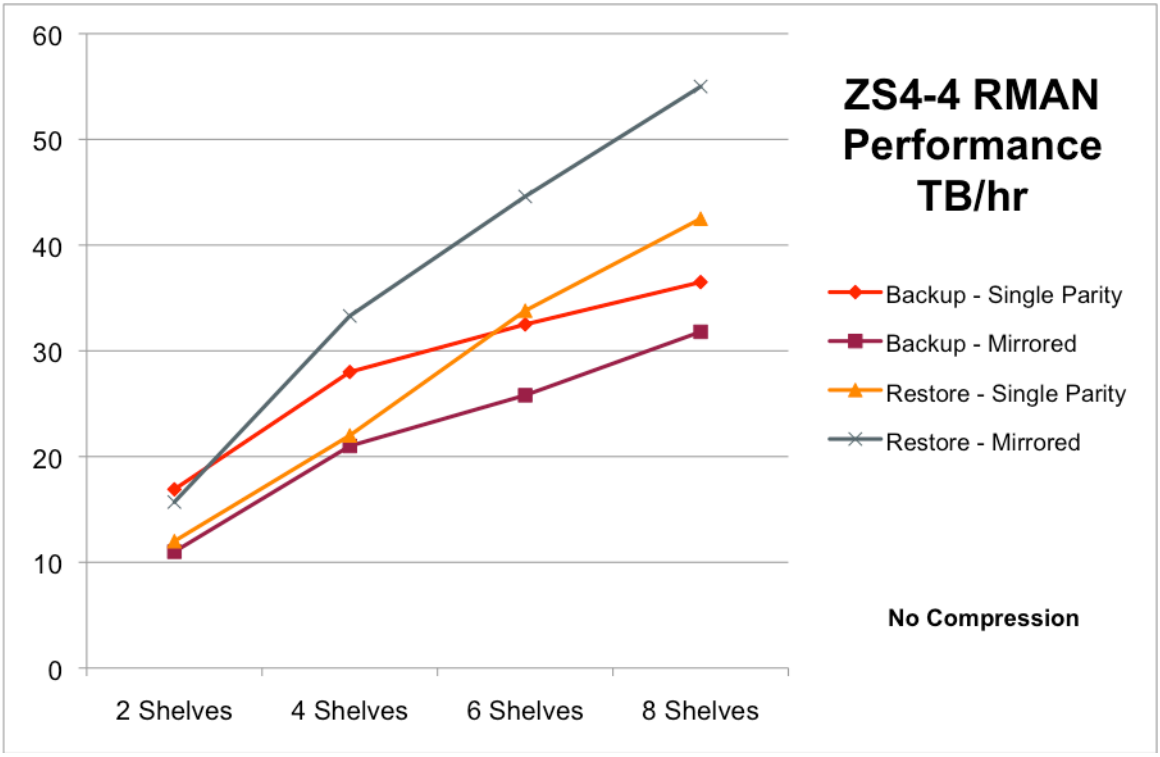

Figure 10: Oracle ZFS Storage ZS4-4 RMAN throughput with no ZFS compression

With certain scenarios, such as restore from mirrored storage, the network throughput is unaffected. It achieves 55TB/hr with eight disk shelves in either case. In other scenarios where HDD utilization or SAS bandwidth are limiting factors in the overall performance of the backup, using LZJB data compression on the ZFS share can alleviate back-end bottlenecks and enable higher throughput over the network.

The final graph shows maximum sustainable RMAN throughput when backing up or restoring from just a single Oracle ZFS Storage ZS4-4 controller. Only one share was used in these level 0 backup and restore operations. Because a share can only be owned by one controller at a time, the resources to run these RMAN workloads came from a single controller. These jobs were completed using half the CPU, memory, network, PCIe, and SAS resources. Performance was collected using storage pool configurations ranging from one to four shelves and including both mirrored and single parity profiles.

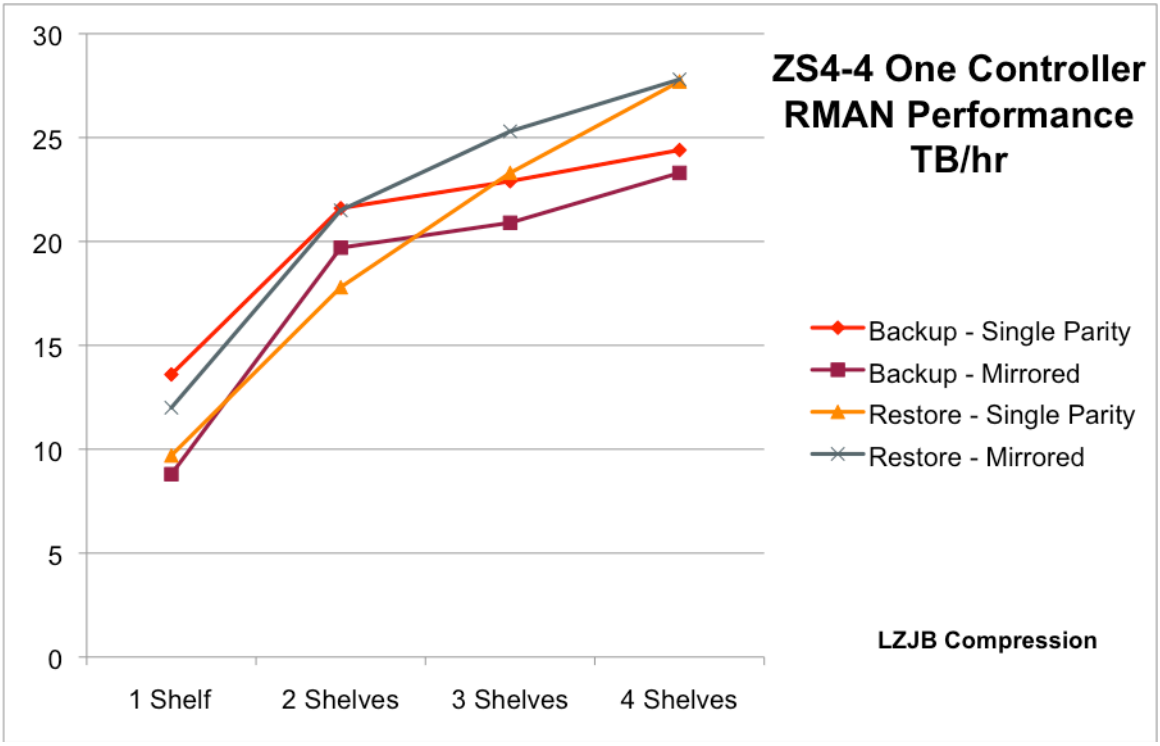

Figure 11: Oracle ZFS Storage ZS4-4 single controller RMAN throughput

Performance sizing with RMAN workloads focused on mirrored and single parity storage profiles since these are most commonly deployed. Double parity provides similar performance characteristics to single parity with RMAN workloads. Performance is slightly reduced but is typically within 5-10 percent of single parity.

Oracle ZFS Storage ZS3-2 RMAN Performance Sizing

The following graph shows sustainable throughput attained in Application Integration Engineering (AIE) during RMAN backup and restore to an Oracle ZFS Storage ZS3-2. These rates were collected running level 0 backups as part of a traditional RMAN backup strategy. The graph demonstrates maximum sustainable throughput for a variety of Oracle ZFS Storage ZS3-2 configurations ranging from two to eight disk shelves.

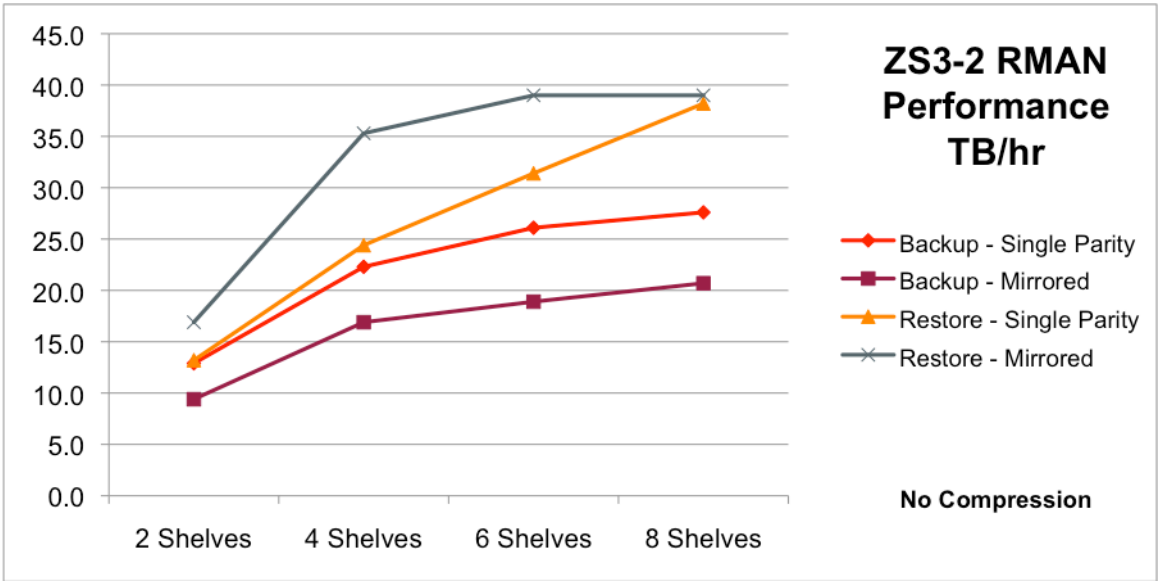

Figure 12: Oracle ZFS Storage ZS3-2 RMAN throughput

These are complete real-world results using Oracle Database 12c and a large OLTP database that is populated with sample customer data in a sales order entry schema. Advanced row compression is used at the database level since this is the best practices recommendation for customers that are running OLTP workloads. This is a fully functional database with the ability to run live transactional workloads before, during, or after an RMAN backup operation. These throughput rates were not obtained using a database or I/O generator test tool which can be misleading and only indirectly applicable to real-world use cases. They were also not projected based on low-level system benchmarks.

In all scenarios, sustainable throughput is determined by measuring physical I/O at the network layer. An average is collected over an extended period of time. These graphs demonstrate the maximum Oracle Database backup and restore rates that the Oracle ZFS Storage ZS3-2 has been proven capable of sustaining in a real-world Exadata backup environment. Maximum sustainable rates of 39 TB/hr restore and 28 TB/hr backup were demonstrated. All of these rates can be achieved with an Exadata Database Machine X5-2 full rack that uses normal ASM redundancy for the DATA diskgroup. Performance was collected using an Exadata and Oracle ZFS Storage ZS3-2 that were both otherwise idle during the Oracle Recovery Manager operation. Test environments were configured following the best practices presented in this white paper.

RMAN performance sizing efforts in the previous graph were conducted on an Oracle ZFS Storage ZS3-2 with two SAS HBAs installed in each controller. However, some ZFS Storage ZS3-2s are deployed with just a single SAS HBA in each controller. In these environments, SAS bandwidth can become a limiting factor for the backup and restore throughput. The following graphs demonstrate the impact of SAS constrained configurations.

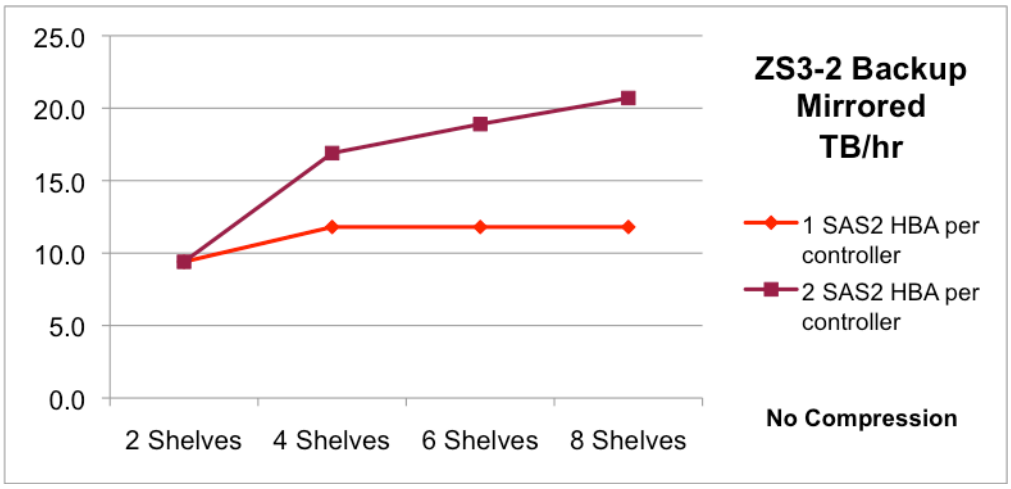

Figure 13: Oracle ZFS Storage ZS3-2 RMAN backup mirrored throughput comparison with multiple SAS HBAs

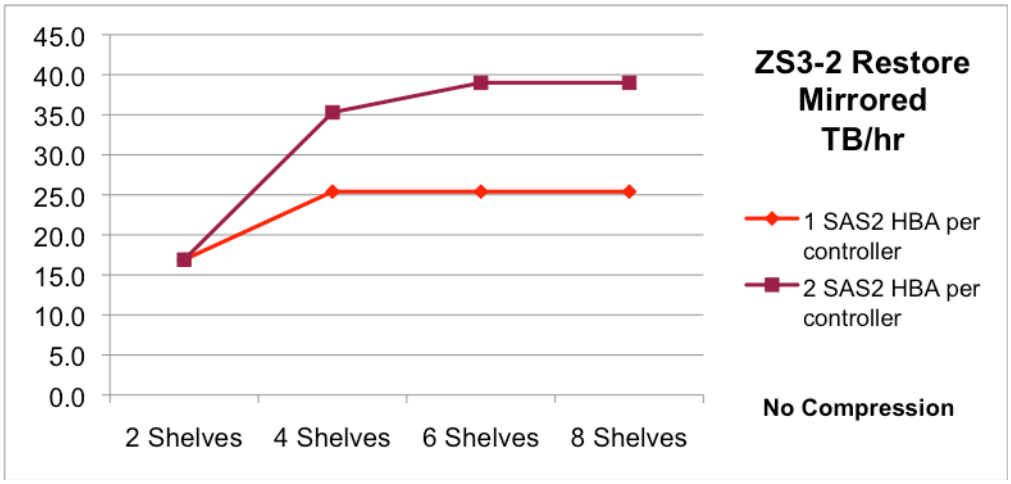

Figure 14: Oracle ZFS Storage ZS3-2 RMAN restore mirrored throughput comparison with multiple SAS HBAs

Backups to mirrored storage are the most taxing on SAS limitations since there is twice as much bandwidth going to back-end disk. Restore operations generate no bandwidth inflation but SAS can still be a limiting factor as shown in the previous graph. Single parity utilizes a narrow 3+1 stripe width which results in 33 percent bandwidth inflation on writes due to parity. It places less strain on SAS bandwidth than mirrored but single HBA configuration can still be easily saturated. It is recommended to use two

SAS HBAs in each controller if the clustered Oracle ZFS Storage ZS3-2 is configured with four or more disk shelves.

Performance sizing with RMAN workloads has focused on mirrored and single parity storage profiles since these are most commonly deployed. Double parity provides similar performance characteristics to single parity with RMAN workloads. Performance is slightly reduced but is typically within 5-10 percent of single parity.

Conclusion

Finding the right backup solution for an Oracle Exadata Database Machine is a challenging problem. Costly alternatives provide poor return on investment and cannot support high-performance environments. Competitive offerings are inflexible and do not address all of the customer's needs.

The Oracle ZFS Storage Appliance has proven to be an ideal solution for protecting the mission-critical data that resides on Oracle Exadata Database Machines. Powerful features combined with custom Oracle-on-Oracle integrations enable a wide range of Oracle Recovery Manager backup strategy options. These provide outstanding performance and flexibility that is unmatched by third party solutions.

Extreme restore throughput helps satisfy even the most stringent recovery time objectives. Archive log multiplexing delivers recovery points of 20 minutes or less. Oracle Intelligent Storage Protocol, Hybrid Columnar Compression, and Oracle Direct NFS provide unique advantages when protecting an Oracle Database. Native InfiniBand support seamlessly integrates with Exadata high throughput infrastructure.

In addition to the data protection benefits, an Exadata backup solution using the Oracle ZFS Storage Appliance provides many other advantages such as low-cost, high-performance storage for unstructured data that resides outside of the Oracle Database and an ideal snapshot cloning solution for provisioning development and test environments. It is easy to see why the Oracle ZFS Storage Appliance is a preferred solution for protecting an Exadata Database Machine.

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

ORACLE

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Integrated Cloud Applications & Platform Services

Protecting Exadata Database Machine with the Oracle ZFS Storage Appliance: Configuration Best Practices
July 2015
Author: Greg Drobish