ORACLE

Architectural Overview of the Oracle ZFS Storage Appliance

April, 2024, Version [1.0]

Public

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

ORACLE

Table of contents

Introduction **5**

Overview **5**

Architectural Principals and Design Goals **7**

Pooled Storage Model and Hierarchy **8**

ZFS Data Integrity **10**

The DRAM-Centric Hybrid Storage Pool **12**

Oracle ZFS – Database Aware Storage **15**

Oracle Enterprise Manager and Oracle VM Integration 18

ZFS Data Reduction **18**

ZFS Deduplication 18

Snapshot and Related Data Services **19**

Other Major Data Services **20**

File Protocols 20

Shadow Migration 21

Block Protocols 21

NDMP for Backup 21

Encryption 21

Management, Analytics, and Diagnostic Tools **22**

Conclusion **22**

Related Links **22**

List of figures

Figure 1. Identifies different configuration options for the Oracle ZFS Storage Appliance 5
Figure 2. Highlights of some key hardware features of an Oracle ZFS Storage Appliance 6
Figure 3. Graphical representation of ZFS pooled storage model 8
Figure 4. BUI Status screen shows current system status 9
Figure 5. Example of how ZFS self-healing architecture corrects a corrupted block. 10
Figure 6. Description of ZFS hierarchical checksums versus traditional RAID approach 11
Figure 7. Illustration of the relative latencies of reads served from different media types 13
Figure 8. Graphical depiction of the Hybrid Storage Pool architecture that illustrates dynamic storage tiering 13
Figure 9. BUI screen view for creating a database share 15
Figure 10. Architecture of a Direct NFS client 16
Figure 11. Overview of Oracle Intelligent Storage Protocol 16
Figure 12. Storage tiering with ADO 17
Figure 13. Oracle ZFS Storage Appliance deduplication components 19

List of tables

Table 1. A sampling of important data services available on the Oracle ZFS Storage Appliance 7

Introduction

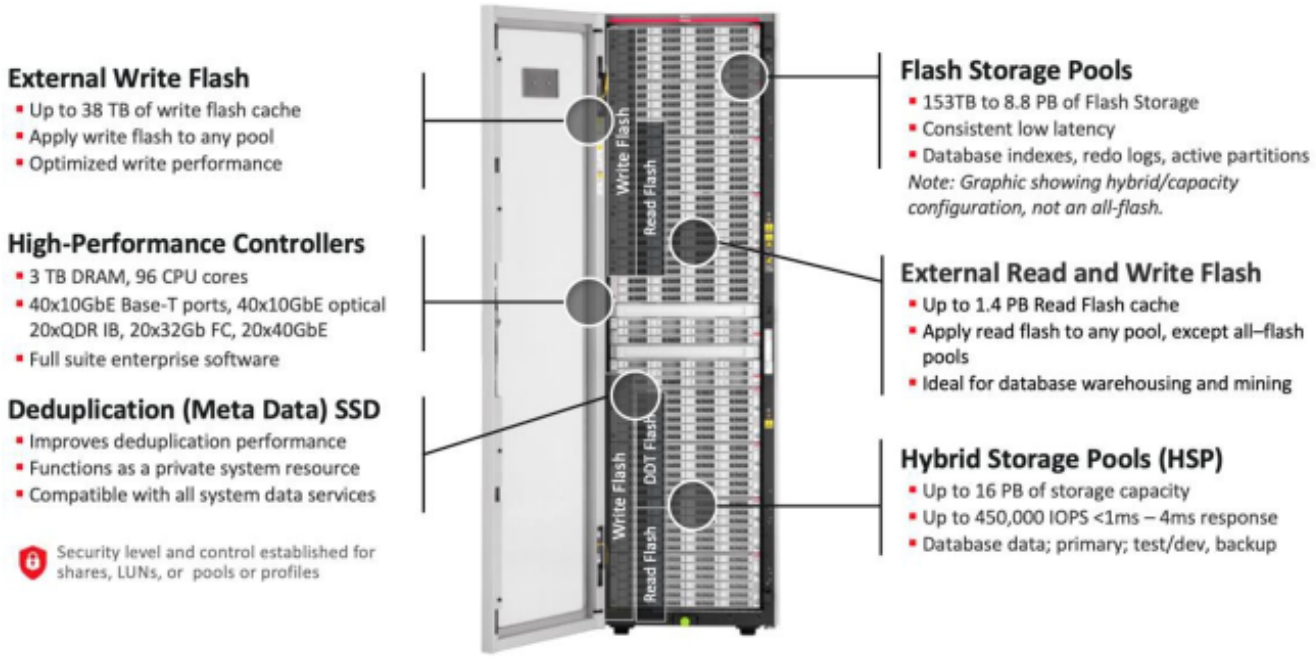
The Oracle ZFS Storage Appliance (Oracle ZFS Storage Appliance) is a best-of-breed, multiprotocol enterprise storage system designed to accelerate application performance, simplify management and increase storage efficiency in a budget-friendly manner. It is suitable for a wide variety of workloads in heterogeneous environments, whether deployed on premises or in the cloud. Collaborative co- engineering within Oracle enables the Oracle ZFS Storage Appliance to offer even more benefits when used in Oracle-on-Oracle environments. The purpose of this white paper is to explore the architectural details of the Oracle ZFS Storage Appliance, examine how it works from a high level, and explain why this unique enterprise storage product is able to drive extreme performance and efficiency at an affordable cost.

Overview

To deliver high performance and advanced data services, the Oracle ZFS Storage Appliance uses a combination of standard enterprise-grade hardware and a unique, storage-optimized operating system (OS) based on the Oracle Solaris kernel with Oracle's ZFS file system at its core.

The Oracle ZFS Storage Appliance supports a new configuration with a storage pool of solid state drives (SSD) to create an all flash configuration, or you can combine an SSD storage pool with a traditional hard disk drive (HDD) storage pool. SSD storage pools are supported with Oracle ZFS Storage ZS4-4, Oracle ZFS Storage ZS5, and Oracle ZFS Storage ZS7-2 models. An all-flash pool provides consistent low latency and is best suited for Oracle Database indexes, redo logs, and active partitions. Figure 1 identifies various Oracle ZFS Storage configurations and the intended workloads.

Figure 1. Identifies different configuration options for the Oracle ZFS Storage Appliance



The Oracle ZFS Storage Appliance storage controllers are based upon powerful Oracle x86 Servers that can deliver the exceptional compute power required to concurrently run multiple modern storage workloads along with advanced

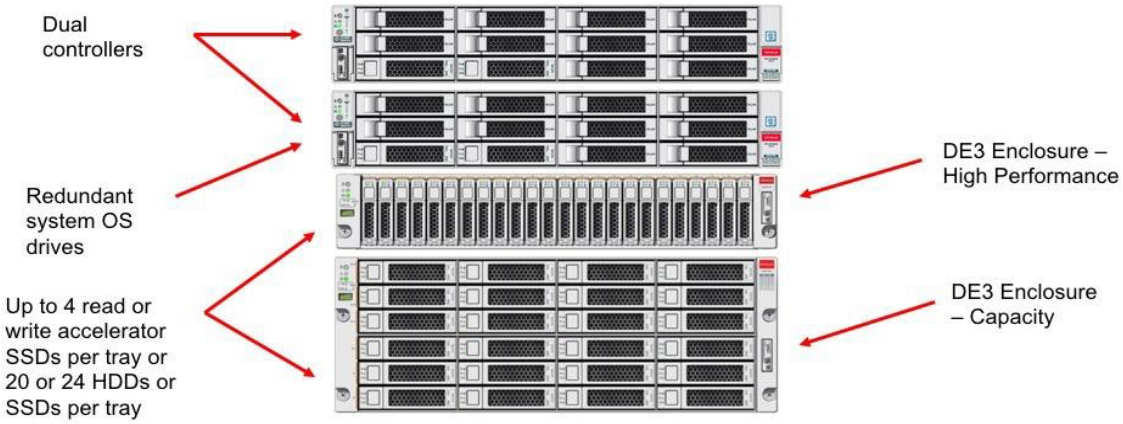
data services. Each Oracle ZFS Storage Appliance can be configured as a single controller or as a dual-controller system. In the case of a dual-controller system, two identical storage controllers work as a cluster, monitoring one another so that a single controller can take over the storage resources managed by the other controller in the event of a controller failure. Dual-controller systems are required when implementing a high availability (HA) environment.

Each controller ingests and sends the traffic from and to the storage clients via a high-performance network. Ethernet, Fibre Channel, and InfiniBand connectivity options are supported for this front-end traffic. The controller then handles the computations required to implement the selected data protection (mirror and RAIDZ), data reduction (inline compression and deduplication), and any other relevant data services (snapshots, encryption, and remote replication). The controllers also handle the caching of stored data in both DRAM and flash. A unique caching algorithm is key to the spectacular performance that can be obtained from an Oracle ZFS Storage Appliance. As it processes traffic, the storage controller then sends the data to or receives the data from the storage media. A SAS fabric is used for this back-end controller connectivity.

The disk/flash pools reside in enterprise-grade SAS drive enclosures with either all-flash storage (SSD), HDD storage, or with hybrid flash/disk combinations. Specific SAS high-endurance SSDs are used to stage synchronous writes so that they can be transferred sequentially to the spinning HDD disks, thus accelerating write performance. For all-flash storage configurations, read accelerators or deduplication meta devices are not required.

Both the controllers and drive enclosures have been configured with availability as the foremost thought. Redundancy is built into all systems, with features like dual power supplies, SAS loops, and redundant OS boot drives.

Figure 2. Highlights of some key hardware features of an Oracle ZFS Storage Appliance



Current hardware models are the Oracle ZFS Storage Appliance ZS7-2 mid range (for midrange enterprise storage workloads) and the Oracle ZFS Storage Appliance ZS7-2 high end (for high-end enterprise storage workloads). For specific details of the current Oracle ZFS Storage Appliance systems' hardware and configuration options, see the Oracle storage product documentation.

All Oracle ZFS Storage Appliance systems run the same enterprise storage OS. This storage OS offers multiple data protection layouts, end-to-end checksumming to prevent silent data corruption, and an advanced set of data services, including compression, snapshot and cloning, remote replication, and many others. Analytics is one of the most compelling and unique features of the Oracle ZFS Storage Appliance and provides a rich user interface to DTrace, a technology available in the Oracle Solaris OS. This Analytics feature can probe anywhere along the data pipeline, giving unique end-to- end visibility of the process with the ability to drill down on attributes of interest.

Table 1. A sampling of important data services available on the Oracle ZFS Storage Appliance

Data Protocols	Data Services	Management

• OISP
• Fiber Channel
• iSCSI
• Infiniband
  ○ NFS/RDMA
  ○ IPoIB
  ○ iSER
  ○ SRP
• Object API
• NFS v2, v3, v4 and v4.1
• SMB v1, v2, v3 and v3.1.1
• HTTP
• WebDAV
• FTP/SFTP/FTPS
• ZFS NDMP v4
	• Hybrid storage pools
	• Single, double, and triple- parity RAID
	• Mirroring and triple mirroring
	• End-to-end data integrity
	• Local and remote replication
	• Snapshots and clones
	• Quotas and reservations
	• Deduplication
	• Compression (5 levels, plus HCC)
	• Pool and share level encryption
	• Thin provisioning
	• Antivirus via ICAP protocol
	• Online data migration
	• Clustering
		• Complete REST API
		• Browser and CLI interface
		• Management dashboard
		• Hardware/component view
		• Role-based access control
		• Usage-oriented phone home
		• Storage Analytics
		• Scripting
		• Workflow automation
		• Advanced networking
		• Source-aware routing

Finally, data services are managed by an advanced management framework, available either through a command line interface (CLI) or a browser user interface (BUI). The BUI incorporates the advanced Analytics environment based on DTrace, which runs within the OS on the storage controller, offering unparalleled end-to-end visibility of key metrics.

For information on the current Oracle ZFS Storage Appliance hardware specifications and options, as well as the latest listing of data services, review the current datasheet.

Architectural Principals and Design Goals

An overarching development goal of the Oracle ZFS Storage Appliance is to provide maximum possible performance from standard enterprise hardware while providing robust end-to-end data protection and simplified management. To take maximum advantage of standard hardware, Oracle Solaris is used as the basis of the storage operating system. Oracle Solaris is a modern, symmetric multiprocessing (SMP) operating system that is able to take full advantage of modern Intel x86 multicore CPUs.

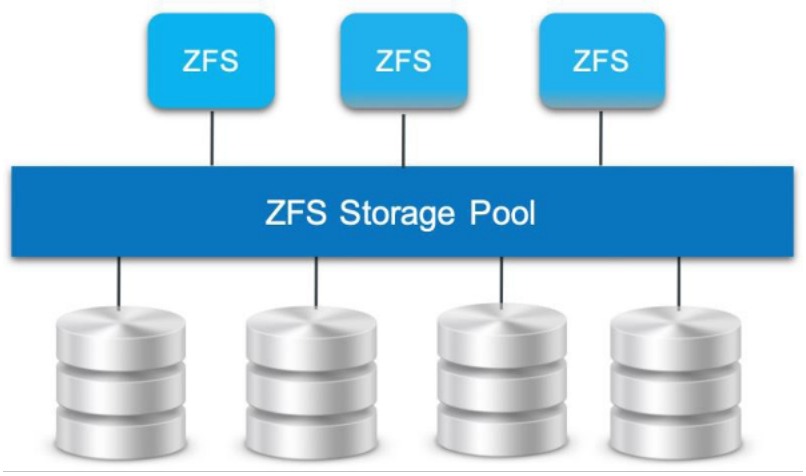
The current Oracle ZFS Storage Appliance OS is a 64-bit architecture to increase virtual memory address space. It also includes Address Space Layout Randomization (ASLR), a preventative design that randomizes memory addresses to help prevent exploit attacks. A 64-bit address space also reduces stress on process virtual memory, improves stability on heavily loaded systems, and provides more flexibility for future designs.

All of these features within the OS mean that the Oracle ZFS Storage Appliance can handle the computational needs to run data protection algorithms, checksumming, advanced data services (such as compression and deduplication), and manage the appliance's advanced automatic data tiering, all while simultaneously maintaining excellent throughput and transactional performance characteristics. This is also a reason the Oracle ZFS Storage Appliance delivers high performance in high burst, random I/O environments like virtualization, which means it can easily boot 1000s of VMs.

To use hardware most effectively and cost efficiently in transactional workloads, the ZFS file system is employed to manage traffic to and from clients in a way that isolates it from the latency penalties associated with spinning disks. This caching, or auto-tiering, approach is referred to as the Hybrid Storage Pool architecture. Hybrid Storage Pool is an exclusive feature of the Oracle ZFS Storage Appliance and is described in more detail starting on page13.

Pooled Storage Model and Hierarchy

Figure 3. Graphical representation of ZFS pooled storage model



A ZFS storage pool is a collection of physical disks rather than the traditional model of a one-to-one connection between the disk or volume and the file system. HDDs and Flash SDDs, write flash-accelerating SSDs, and read flash-accelerating SSDs are physically grouped together in these physical pools of resources. Within a given pool, the storage devices all are subject to the same layout (for example, mirror or RAID) and are managed by the same assigned storage controller. Thus, in a dual-controller cluster, for an active/active setup, you must provision the physical devices in at least two distinct pools so that each active controller can manage at least one pool.

A ZFS *virtual device* (vdev) is an internal representation of the storage pool that describes the layout of physical storage and the storage pool's fault characteristics. As such, a *virtual device* represents the disk *devices* or files that are used to create the storage pool.

Each pool can contain multiple vdevs. For example, if a double-parity RAID layout is selected for the pool, based upon the built-in best practice, a 4+2 stripe width is used with each vdev containing nine disks. The Oracle ZFS Storage Appliance OS effectively masks the vdevs, and “hot spares” (unused disks that are preinstalled and ready for use as a replacement in case of failure of an active data disk) are provisioned automatically, based upon built-in best practices. Therefore, all you need to be concerned with in terms of physical components is the pool level.

Figure 4. BUI Status screen shows current system status

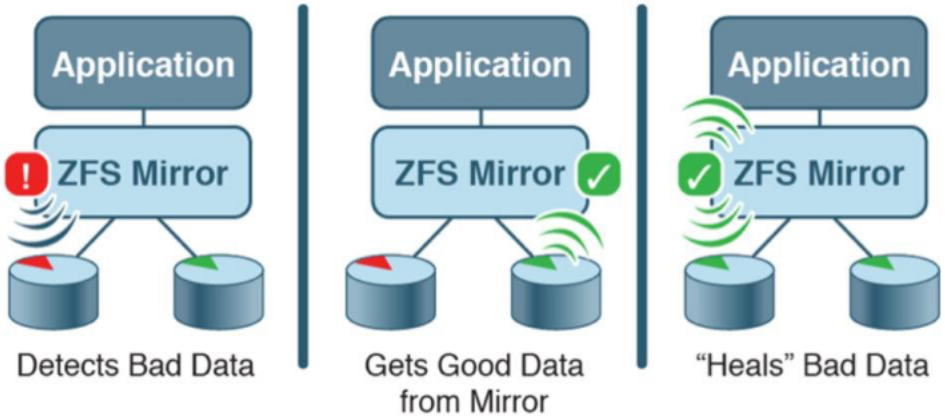
After a pool is created with the devices and layout you specify, the pool is the basic physical resource from which you and the system work. Each file system and LUN that is created within the pool has a maximum available capacity equal to the remaining formatted capacity of the whole pool. File systems and LUNs have various share setting options. A share quota can be established so that the capacity within that share cannot exceed a threshold and can therefore not consume more than its quota of the pool. A share reservation also can be set such that the other shares in the pool cannot occupy more of the pool than is possible without infringing on the reserved capacity.

After the pool, the next hierarchical data structure is known as the project. Think of a project as a template for creating shares. Shares have many settings that can be optimized for different use cases, and projects provide a way to make a template so that shares can be easily provisioned for the same or similar use cases, repeatedly over time, without undue effort. For example, a typical Oracle ZFS Storage Appliance might have three concurrently running workloads: user home directories in SMB shares, VM files in NFS shares, and some LUNs associated with different e-mail or collaboration software. A project could be created to group the shares separately for each of these use cases. As new VM servers are deployed, or as new user home directories are added, or as new e-mail or collaboration accounts are created, an admin might want to create a new share that is totally new and empty, but has the raw properties that the other similar shares utilize. To do so, you simply create a new share under the correct project. After a share has been created based upon a project, the settings of the individual share can be simply edited to be unique to that project, if needed, without changing the project. In other words, just because you started with the template does not imply that you have to conform to it later. If you change the settings at the share level, that share will have a setting that is distinct from the rest of the shares in that project. If, however, you change a setting at the project level, then all shares in that project inherit the new setting. Note that projects are completely thin provisioned in that they have no notion of size—it is simply a set of attributes that apply to their associated shares. Thus, if physical capacity is added to a pool, the projects in that pool are not impacted in any way. It simply increases the total available capacity for all files, all shares, and all projects associated with the pool.

ZFS Data Integrity

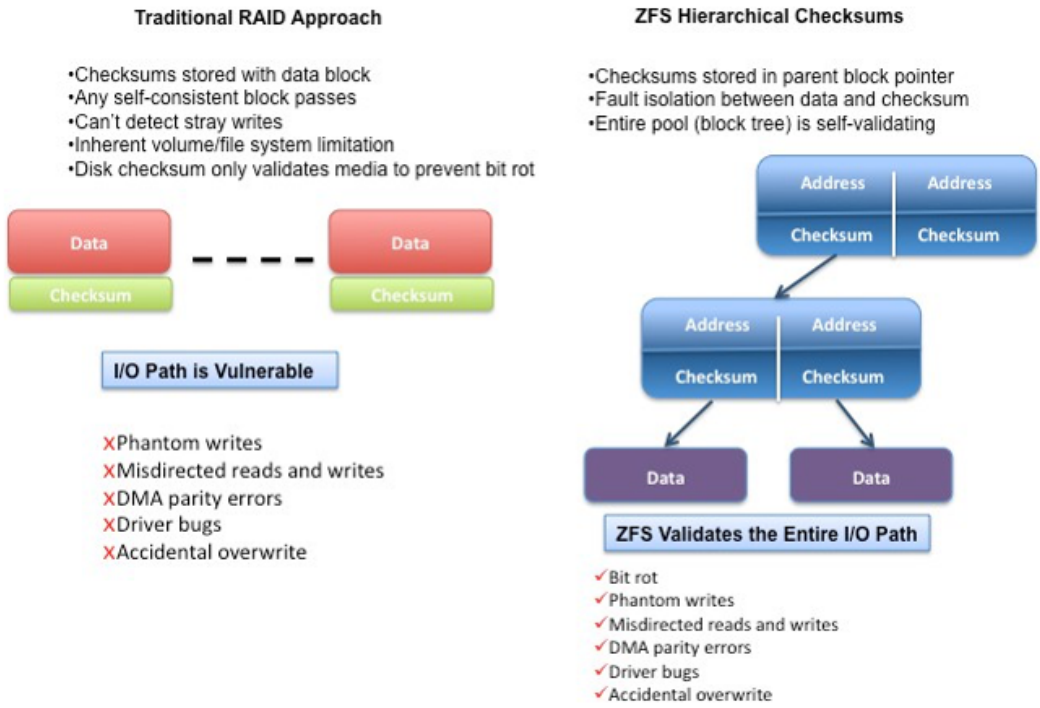
The Oracle ZFS Storage Appliance provides robust data protection at many levels, to help ensure data integrity and protection against silent data corruption while also protecting data from hardware failures. It is able to perform end-to-end checksumming throughout all controller-based cache components all the way down to the disk level. This is because the Oracle ZFS Storage Appliance uses no RAID controllers—rather the storage controller's operating system itself runs ZFS data protection. This architecture provides the file system full control of data movement and visibility of the blocks so that integrated end-to-end checksumming can be performed. ZFS checksumming is a more advanced type of hierarchical checksumming versus the traditional flat checksumming approach. Traditional checksumming can check the integrity of only one block in isolation of others, meaning that media bit rot can be successfully screened, but other types of data corruption across the I/O path cannot be identified. ZFS checksumming uses a Merkle tree, whereby data blocks are distinct from address block checksums and the entire I/O path can be protected from end to end.

Figure 5. Example of how ZFS self-healing architecture corrects a corrupted block.



Another advantage of ZFS checksumming is that it enables a self-healing architecture. In some traditional checksum approaches, wherever data blocks get replicated from one location to another there is an opportunity to propagate data corruption. This is because, with traditional checksum approaches, the newest data block simply gets replicated. With ZFS checksums, each block replica pair has a checksum calculated independently. If one is corrupted, the healthy block is then used as a reference to repair the unhealthy block.

Figure 6. Description of ZFS hierarchical checksums versus traditional RAID approach



Oracle ZFS Storage Appliance hierarchical checksums are able to detect data inconsistencies that traditional RAID products cannot.

The most typical hardware failure in enterprise storage is, of course, disk failures. The Oracle ZFS Storage Appliance provides robust protection from hardware failures, offering multiple options. Each storage pool layout defines how data could be protected. Available layout options are:

- Stripe (no media failure protection), which is not recommended for production workloads
- Mirror (single disk failure of a paired-set protection) or triple-mirror (dual disk failure of a triple-set protection)
- RAIDZ-1 single-parity raid (single disk failure protection within a four-disk set)
- RAIDZ-2 dual-parity raid (dual disk failure protection within a 9, 10, or 12-disk set, depending on pool drive count)
- RAIDZ-3 (triple-disk failure protection within a multiple disk set, where stripe width varies depending on pool disk count)

Mirroring tends to offer the highest performance for latency-sensitive transactional workloads, with triple mirroring being a good transactional performance option as well when a higher protection level is desired. Single-parity RAID tends to offer excellent throughput performance in streaming workloads, with dual-parity RAID being a reasonable throughput option when higher protection is desired, with some performance expense. Write SSDs can be either striped or mirrored, while read SSDs are always striped. Oracle ZFS Storage ZS4-4, ZS5 and ZS7 platforms running deduplication workloads require *metadevice* SSDs. For more information, see ZFS Data Reduction.

Oracle has established best practices for balancing media protection requirements with performance requirements for a variety of use cases. Oracle Sales Consultants can advise you about their particular environment or users can review publicly available documents on the Oracle Technology Network.

Data stored in a ZFS storage pool can be routinely verified by running a pool scrub. Any data inconsistencies found can be resolved during the scrubbing process and logged internally so that you can obtain quick overview of all known errors within the pool. A recommended best practice is to scrub your pools once per quarter. In the current release, a pool scrub is automated by default and scheduled every 30 days.

The DRAM-Centric Hybrid Storage Pool

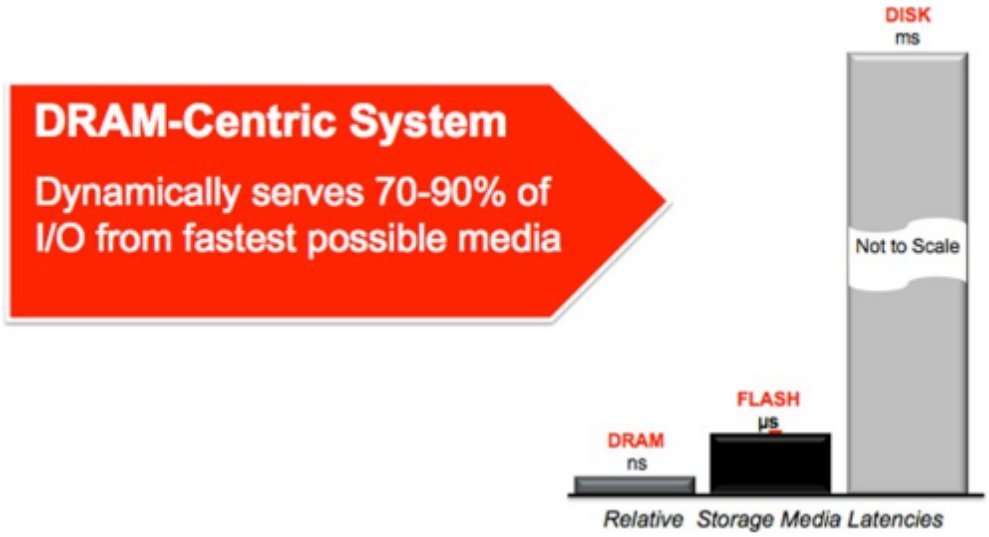
The Hybrid Storage Pool architecture is the core technology that enables the Oracle ZFS Storage Appliance's superior performance. Using an intelligent and adaptive set of algorithms to manage I/O, the Oracle ZFS Storage Appliance is able to make most efficient use of modern hardware resources: automatic placement of data on dynamic random-access memory (DRAM), read and write optimized flash-based SSD, and SAS disk for optimal performance efficiency. Read and write paths are each handled in a distinct manner to address the unique performance and data integrity needs of each.

In the Hybrid Storage Pool architecture, DRAM is treated as a shared resource. As in most storage operating systems, DRAM serves as a resource for controller operations overhead for the operating system. But uniquely, and crucially for performance, DRAM is also used as a primary cache to accelerate reads. Because DRAM is a much faster media type than either disk or flash for transactional workloads, having a high proportion of read operations served out of DRAM radically accelerates overall system performance. The portion of DRAM used to serve as a read cache is known as the Adaptive Replacement Cache (ARC). DRAM allocation to ARC is managed by the operating system on a dynamic basis to maximize overall system performance. (The ARC is shared across all pools—the Hybrid Storage Pool read-tiering algorithm is global and not associated to any particular pool for the ARC, but the L2ARC is assigned per pool.)

In the ARC, blocks are classified as most recently used (MRU), most frequently used (MFU), least recently used (LRU), or least frequently used (LFU). The idea is to keep the "hottest" portion of the overall data set in DRAM. As the ARC becomes saturated and hotter data needs to replace cooler data in the ARC, the Hybrid Storage Pool evicts the coolest data in DRAM to a read flash cache device. This is known as the Level 2 ARC (L2ARC), for which the Oracle ZFS Storage Appliance uses SSDs. Read requests for data that had not been judged hot enough to be placed in either ARC or L2ARC must be served from stable storage, resulting in a higher latency on those reads.

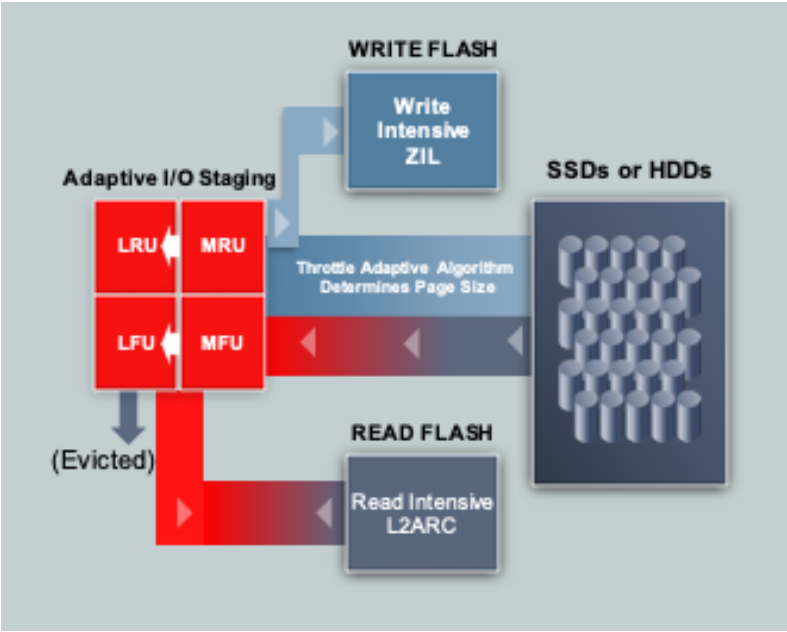
However, in practice, it is common to have ARC hit rates in excess of 80-90 percent across a wide sampling of installed base systems. This means that in the vast majority of workloads, performance tends toward accelerated, cached, DRAM speeds. DRAM is orders of magnitude faster than flash, which is orders of magnitude faster than spinning disk—which is why it is important that the Oracle ZFS Storage Appliance features this DRAM-centric architecture for serving reads, unless an all-flash pool is serving the read workload. This acceleration is beneficial for performance in any latency-sensitive workload. For example, in database workloads, where the hot portion of data is automatically assigned so that reads are issued from DRAM, or in server virtualization use cases, where OS images can be automatically cached in DRAM, this DRAM-centric architecture can dramatically reduce read latency, accelerating application host performance.

Figure 7. Illustration of the relative latencies of reads served from different media types



It is important to note that, for the read path, *all* data resides on spinning disk, whether cached or uncached. The automated caching done by the Hybrid Storage Pool also writes a duplicate sets of blocks in the ARC or L2ARC into the main storage pool. This is important because, in the event of a controller failure, all data is protected because it is persistently stored on spinning disk. In a dual-controller system, upon failure of a pool's primary controller, the second controller can take over the pool's disk resources, have access to all data, and serve the reads just as the first controller would have done. Any cached reads are checksummed against the persistent storage and any changed blocks are updated before serving the read to the client.

Figure 8. Graphical depiction of the Hybrid Storage Pool architecture that illustrates dynamic storage tiering



The write path is handled differently. Incoming writes to the appliance initially land in DRAM. Clients can potentially issue writes to storage either synchronously or asynchronously, although for most enterprise client OSes, hypervisors,

and applications, writes are generally requested synchronously for data protection and consistency reasons. Asynchronous writes are acknowledged as complete to the client immediately upon landing in DRAM.

Synchronous writes to the Oracle ZFS Storage Appliance are *not* acknowledged immediately upon landing in DRAM. Instead, they are acknowledged after they are persistently stored on disk or flash. There is a mechanism in place to replay any writes that might have been interrupted on their way to stable storage due to a system crash or a power failure, called the ZFS Intent Log (ZIL).

The ZIL is stored on a low-latency, high-endurance and write-optimized SSD and has several critical purposes:

- Keeps a linked list of records to be replayed if a power outage or system crash occurs before the contents are written to stable storage
- Provides in-memory state of every POSIX operation that needs to modify data
- Reduces synchronous write latency, such as those for Oracle Database operations

The logbias property can be set as either latency or throughput depending on the particular workload. For latency workloads, such as redo log shares for transactional databases and certain VM environments, the logbias

= latency share setting is selected so that the ZIL is enabled and writes are immediately copied from the system DRAM buffer into the high-endurance SSD. For throughput or streaming workloads, such as query-intensive database workloads or media streaming, latency is not as critical as data transfer rates. In these workloads, the logbias = throughput share setting should be used so that write log blocks are allocated from the main pool, thus skipping the SSD and going straight from the controller DRAM buffer to spinning disk. (Contrary to common misperception, groups of spinning disks can be faster than smaller number of SSDs for throughput workloads.) In this case, once stored persistently on spinning disk, the write-complete acknowledgement is delivered to the client. Additionally, the ZIL should be mirrored for additional data protection.

Inside the specialized write-optimized SSD are three primary components:

- NAND flash, used to persistently store the ZIL data
- DRAM buffer, to stage data entering the SSD before transferring to the NAND flash
- Tantalum capacitor, designed to provide power to allow flushing of the DRAM buffer to the NAND flash in the event of power supply loss to the SSD, before the buffer is cleared to flash.

The SSD contains the embedded circuitry required for persistent storage of the acknowledged data and to handle any flash error corrections needed to complete a restore upon power resumption. In this manner, the SSD serves as a mechanism to persistently and safely stage writes and accelerate write-complete acknowledgement, dramatically reducing write latency without risking the integrity of the data set.

With this architecture, reads are mostly cached in DRAM for optimal read performance, and writes are handled either to optimize latency performance or throughput performance, all while providing data integrity and persistency. The performance benefits of the Oracle ZFS Storage Appliance are well documented and independently verified.

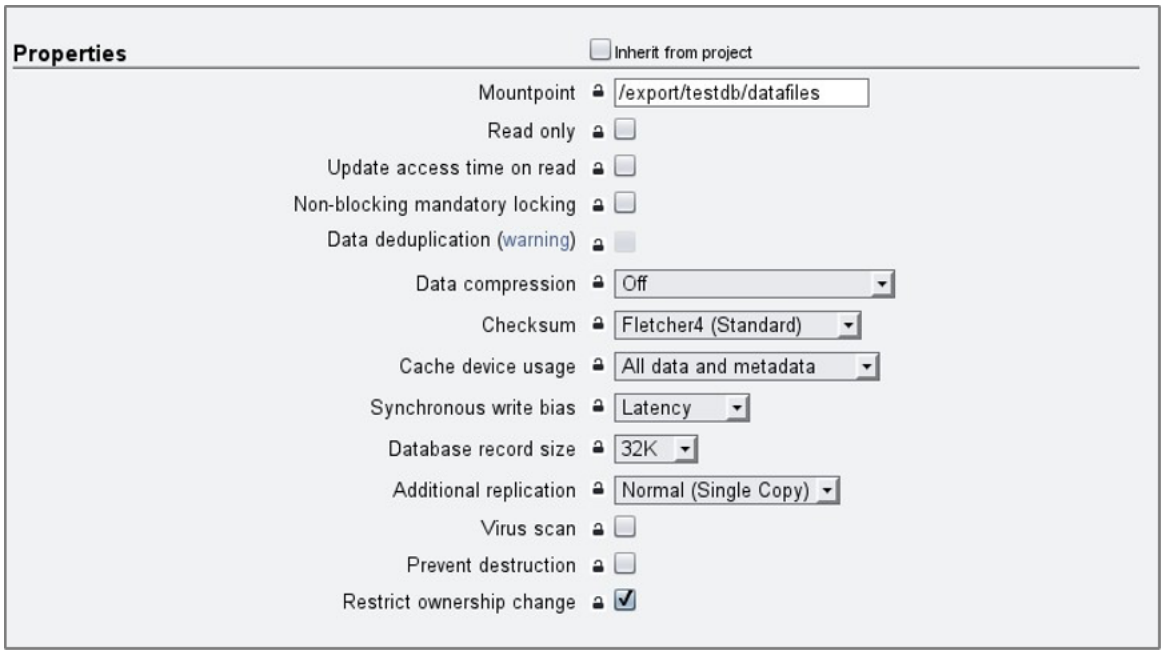
Oracle periodically publishes Storage Performance Council's SPC-1 and SPC-2 benchmark results, as well as Standard Performance Evaluation Corporation's SPEC SFS benchmark results to demonstrate performance results for the Oracle ZFS Storage Appliance. Visit Standard Performance Evaluation Corporation's website (www.spec.org) and the Storage Performance Council's website (www.storageperformance.org) for the latest independently audited, standardized storage benchmark results for the Oracle ZFS Storage Appliance and for results of many competitors.

Oracle ZFS – Database Aware Storage

It is very easy to set up a database on Oracle ZFS Storage Appliance. The steps in summary are as follows:

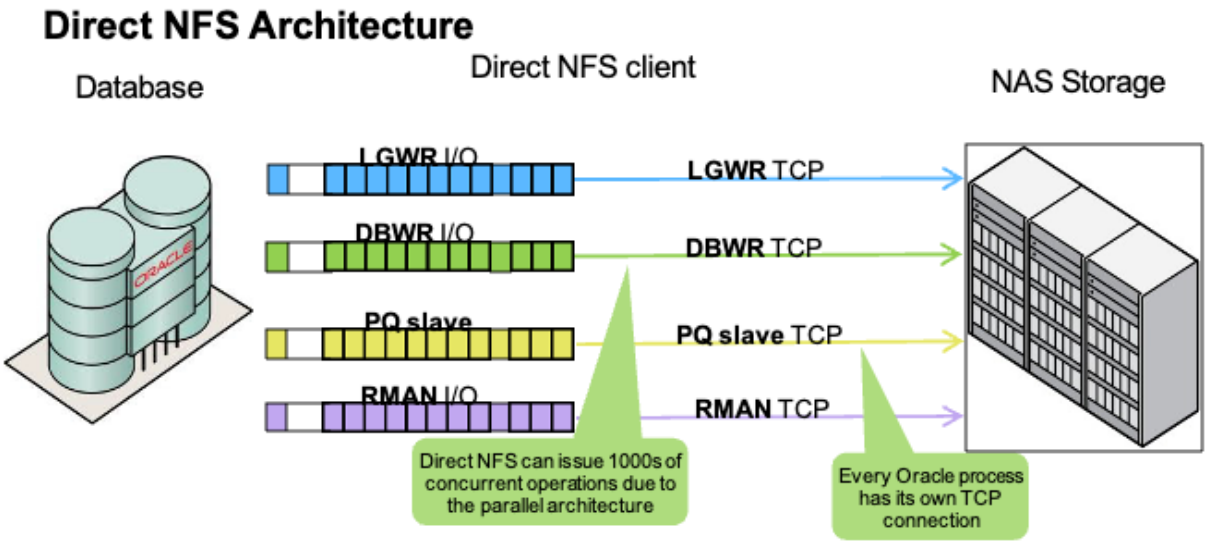
1. Create a storage pool with required capacity and configuration (HDDs or SSDs) and desired protection mode (mirror, RAIDZ1, RAIDZ2).
2. Create a project for the database.
3. Create shares (file systems or LUNs) to be associated with the project based on the client access type (file or block) and with recommended settings for record size and write bias (based on database workload).
4. Mount the shares on the client using OS kernel mount options and change directory ownership to oracle user.
5. Create the database using the mounted shares.

Figure 9. BUI screen view for creating a database share



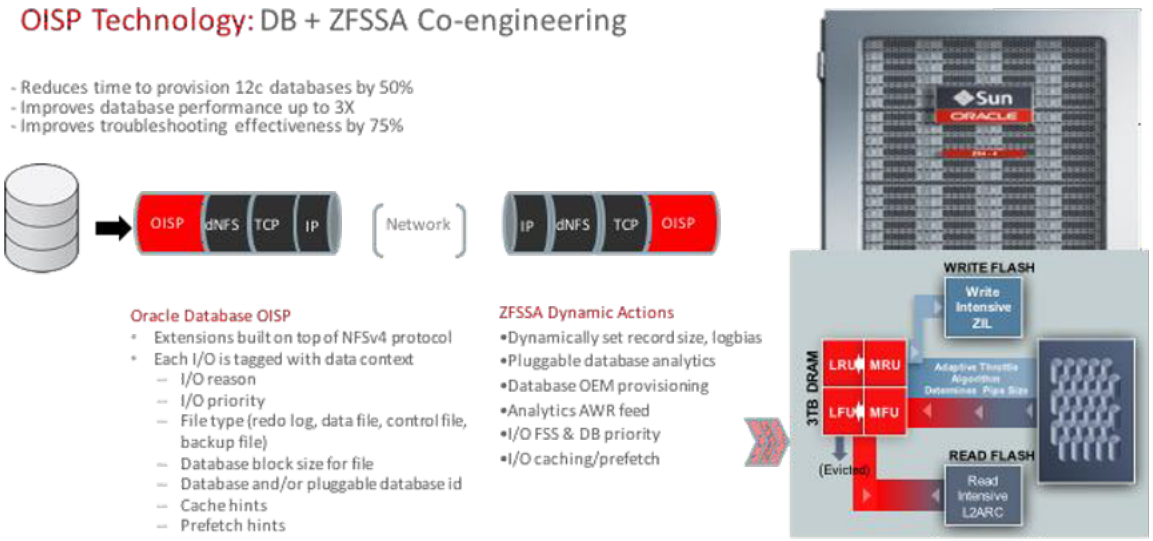
The Hybrid Storage Pool architecture of Oracle ZFS Storage Appliance enables optimal database performance by retaining the most recent and frequently accessed blocks in DRAM, which is the fastest medium on the appliance.

Figure 10. Architecture of a Direct NFS client



When deploying Oracle databases on the Oracle ZFS Storage Appliance, it is strongly recommended to implement the Direct NFS client that comes standard with Oracle Database 11g and later releases. The integrated Direct NFS client within the database stack bypasses many kernel OS limitations and can read data blocks directly into SGA from the storage appliance. The Direct NFS client can optimize mount options based on the workload and network characteristics and provides support for multiple network interfaces between the Oracle Database and Oracle ZFS Storage Appliance. This provides more bandwidth for I/O operations and failover options across the multiple interfaces. When this architecture is coupled with 40GbE, 10GbE or InfiniBand networks, it provides an extremely fast pipe between database and storage appliance, leading to very fast response times that benefit both transactional workloads like OLTP and streaming workloads like OLAP or backups/restores.

Figure 11. Overview of Oracle Intelligent Storage Protocol



Another aspect of co-engineering that makes Oracle ZFS Storage Appliance unique for hosting Oracle Database workloads is a protocol called Oracle Intelligent Storage Protocol. This protocol allows Oracle databases, Release 12c

and later, to send dynamic hints to the Oracle ZFS Storage Appliance. These hints indicate the type of workload running on the database server and the nature and priority of that workload. This kind of intelligence lets the appliance make dynamic choices for record size of database files and logbias settings of the shares. This feature provides consistent database performance because the critical database operations (logbuffer writes, checkpoint writes, voting disk writes) are recognized by the appliance and are configured to have their own pool of threads to service them. This feature also prevents a large block streaming workload, like a database backup, to take up a huge chunk of resources (DRAM/CPU) on the appliance, starving synchronous database operations of resources and potentially degrading database response times. Another advantage of this protocol is that it allows negative cache hints to be passed with large block streaming workloads. The Oracle ZFS Storage Appliance can evict those blocks from DRAM, effectively freeing up that DRAM for staging critical database blocks. Analytics is also enhanced with this protocol, as information that is passed with each I/O operation can be displayed through various charts on the BUI. Deep drilldowns into file-level operations are possible to get a more granular view of database operations. This granular view allows DBAs to compare AWR reports with what storage administrators can view on their BUI, leading to a far more effective troubleshooting experience in complex multitenant database environments.

Oracle ZFS Storage Appliance also supports hybrid columnar compression (HCC). A feature of Oracle Database 11g and later releases, HCC is only supported on Oracle products, such as the Oracle ZFS Storage Appliance. HCC is well-matched for use with data warehousing applications as it provides a far higher compression level (up to 50X for archive data) and speeds up query response times. To enable HCC on Oracle ZFS Storage Appliance, the SNMP service must be activated. Support for thin-cloning technologies at the OS layer makes it extremely fast to spin up clones of databases hosted on Oracle ZFS Storage Appliance, without taking up additional storage in the storage pools. Storage is consumed only as data blocks are updated or added in the clones. Oracle ZFS Storage Appliance can be used as part of Oracle ADO (active data optimization) storage-tiering policy to automatically compress and move older data to shares on slower spinning disks, on premises or to the cloud. This makes Oracle ZFS Storage Appliance ideally suited for implementing data lifecycle management.

Figure 12. Storage tiering with ADO

Oracle Enterprise Manager and Oracle VM Integration

An Oracle Enterprise Manager Plug-in is available for Oracle ZFS Storage Appliance for end-to-end management visibility with monitoring and provisioning at the share, LUN, or project level for all appliance models. Oracle VM Storage Connect Plug-in for Oracle ZFS Storage Appliance enables Oracle VM to provision and manage the appliance for a streamlined virtualization implementation.

ZFS Data Reduction

The Oracle ZFS Storage Appliance has several options for data reduction, offering five levels of compression plus a deduplication option, primarily recommended for backup workloads. These are share options that can be set at either the share or project level and can be applied to file systems or LUNs equally. All of these data reduction mechanisms work inline at the ZFS block level. Although the options can be altered on the fly after initial share creation, only new data is subjected to the new policy. The at-rest data is unaffected by the new setting, unless and until those ZFS blocks at rest are accessed and changed.

Typically, the following compression options offer the best combination of data reduction and performance, based upon the desired backup workload:

- LZ4 – Lower overhead compression algorithm
- LZJB – Low overhead compression algorithm
- GZIP – Higher compression algorithm but incurs compute overhead
- GZIP 2, 9 – Increasingly higher compression algorithm but can incur significant compute overhead
- Hybrid Columnar Compression (HCC) – Integrated Oracle Database compression feature that yields compression ratios in the 10X-50X range

Both LZ4 and LZJB are lightweight compression algorithms that yield compression ratios in the 2X-5X range. In fact, using either LZ4 or LZJB can improve performance because these algorithms use so little compute overhead, and because transmitting compressed data means that the traffic across the SAS fabric between the controller and the disks is reduced. The net effect is increased bandwidth utilization and, often, a net increase in performance versus using no compression at all (though this is not always the case).

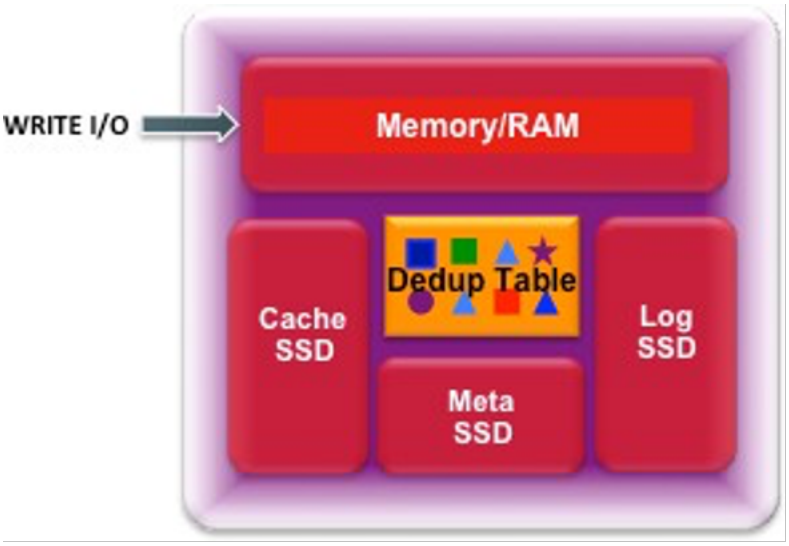
GZIP-2, GZIP, and GZIP-9 are standard compression algorithms that are commonly used industry-wide in a variety of IT implementations (www.gzip.org provides more information from the developers of these algorithms). These options can provide higher compression than LZ4 and LZJB, but typically with greater compute resource utilization. In systems that have significant excess compute capacity relative to the workload, or in environments where data reduction is more important than maximum performance, using GZIP compression might be an option.

ZFS Deduplication

ZFS deduplication is another data reduction option available on the Oracle ZFS Storage Appliance. The implementation is inline, block-level deduplication, which means that redundant data can be reduced when it is written to disk. The way it works is that a hash is calculated for each block written to storage, which is then compared to entries in a deduplication table (DDT) of previously written blocks. When an incoming block matches the DDT entry, it is not written to storage. Instead, a reference pointer to the existing block is used, resulting in reduced storage space for large amounts of backup data.

The deduplication design provides that the DDT table is available in memory and on DDT solid-state drives (SSDs) called metadevices, leveraging the existing Hybrid Storage Pool model to scale performance.

Figure 13. Oracle ZFS Storage Appliance deduplication components



Overall deduplication benefits include the following:

- Provides excellent data reduction yields for full backup use cases
- Leverages powerful Hybrid Storage Pool for scalable deduplication performance
- No additional licensing costs for deduplication or compression

ZFS deduplication is recommended for backup use cases with the Oracle ZFS Storage Appliance as the backup target, where both deduplication and LZ4 compression are used. Full or level 0 backups yield the highest rate of deduplicated data, and some customers have reported combined data reduction ratios in the 15X-20X range. The following backup products are supported with ZFS data reduction features:

- Oracle Recovery Manager (Oracle RMAN) for Oracle Database backups
- Microsoft SQL Server dumps
- Veritas NetBackup (NBU) Open Storage Technology (OST)
- VEEAM v. 9 for VM backups

Excellent data reduction rates can be achieved along with optimized performance. Oracle has many documented solutions and best practices for a variety of environments, particularly Oracle Database environments, to optimize overall system performance and data reduction. Additional information is available on Oracle Technology Network , the Oracle Optimized Solutions page, or from an Oracle Sales Consultant.

Snapshot and Related Data Services

The Oracle ZFS Storage Appliance features a snapshot data service, as well as several other data services built using this snapshot capability as a foundation. Snapshots serve as point-in-time, read-only copies of data. They can be a restore point for a data set, should it be desired, to roll the state of the data set back to a previous point in time. Note

that while snapshots create a logical restore point, they do not provide a physical backup. (Data should always be appropriately backed up physically in accordance with criticality appropriate data redundancy policies.) Snapshots also can be referenced as read-only shares from clients (unless you disable this option) by mounting the root of the file system and changing the directory to `.zfs/snapshot`. The maximum number of snapshots is virtually unlimited. They take up almost no space, and take very little time to create, so many customers use them liberally. Because of the Merkle tree and copy-on-write technology used in ZFS, snapshots are effectively penalty- free. Snapshots can be scheduled or taken manually, depending on usage and policies.

The cloning feature of Oracle ZFS Storage Appliance is one data service based upon the snapshot technology. Clones are essentially zero-copy read/write shares based on a snapshot. Clones are created from existing snapshots, turning that snapshot into an object that behaves as any regular share to the user. But in the background, the Oracle ZFS Storage Appliance is tracking and storing only changes to blocks and referencing the original data set, thus avoiding any requirement to fully duplicate the entire data set to have a second, distinct working copy. Clones provide a simple, fast way to provision test/dev/QA environments with minimal incremental storage consumption over the data set upon which the snapshot is based, for example. (Again, note that clones do not provide a physical backup, although mounting a clone and copying it to a distinct location on physically distinct storage could.)

The replication feature of Oracle ZFS Storage Appliance is another available data service, and is also based on the snapshot technology. Replication can be used to replicate a data set from one pool to another pool on the same appliance, for example. Replication also can be used externally to replicate a data set from one Oracle ZFS Storage Appliance to another appliance. This is useful for facilitating backups or for DR purposes, or simply to move a data set from one physical location to another for some other purpose. Oracle ZFS Storage Appliance Replication is asynchronous, meaning that WAN latency does not impact acknowledgment of client requests. (Note that for environments requiring synchronous replication, a variety of host-based solutions are available to accomplish this goal in a higher availability, higher performance manner than storage-based synchronous replication typically would. Additional information is available on Oracle Technology Network, or from an Oracle Sales Consultant.) Replication can be handled manually, can be scheduled, or can be automated with scripting. Another option is continuous replication, which starts the next replication run as soon as the previous is completed. This helps ensure the minimum possible gap/data-loss window. Because it is snapshot-based, replication is incremental in nature, meaning that the initial replication will move the entire data set but subsequent replication of the same data set will move only changed blocks, resulting in shorter subsequent replication times. Replication can be invoked at either the project or share level on the source appliance.

Other Major Data Services

File Protocols

The Oracle ZFS Storage Appliance supports many common protocols, both file and block. NFS versions 2, 3, 4, and 4.1 are supported. Configuring NFS with Kerberos is also supported. SMB versions 2, 3, and 3.1.1 are supported, along with key SMB features such as encryption, opportunistic locks, SMB signing, and Active Directory. (Note that the Oracle ZFS - Appliance only supports membership in one Active Directory domain or one Kerberos Realm. Multiple domain/realm membership is not supported.)

SMB 3.1.1 encryption supports both AES-128-CC and AES-128 GCM algorithms. When SMB encryption is enabled on the appliance, the SMB protocol automatically negotiates the best encryption algorithm so no manual selection is required by the Windows client. On the appliance, SMB 3.1.1 encryption can be enabled globally, which means data is encrypted for all new sessions or enabled for all new sessions on a specific share.

A single share can be accessed via both NFS and SMB simultaneously. An identity management service is built in to facilitate sharing of identities between Windows (SMB) and UNIX (NFS) systems so that mixed clients can access the same share simultaneously. FTP is supported (along with SFTP and TFTP) so that the Oracle ZFS - Appliance can be used as an FTP server. Access to shares is also possible via HTTP or HTTPS as the appliance implements the WebDAV extension. NIS and LDAP authentication are supported for NFS, HTTP, and FTP access. A virus scanning option is built-in and, when invoked, performs inline scanning whenever a file is accessed via any of these file protocols and will quarantine as necessary. DNS, Dynamic Routing, IPMP, LACP, and NIS are all supported.

Shadow Migration

Another feature of the Oracle ZFS - Appliance is Shadow Migration. This service uses the NFS protocol to migrate data from any legacy NFS system to the Oracle ZFS Storage Appliance in a low downtime manner. Clients are disconnected from the legacy filers and then reconnected to the Oracle ZFS Storage Appliance, and then the Oracle ZFS Storage Appliance is connected as a client to the legacy filer. The directory structure of the legacy filer is scanned and data begins migrating to the Oracle ZFS Storage Appliance. The system is online, so if a client requests a file not yet present on the Oracle ZFS Storage Appliance, it will be retrieved from the legacy filer and copied to the Oracle ZFS Storage Appliance and passed on to the client. Other files migrate as a background activity. While not intended as a fast migration method, it does provide for extremely low downtime migration.

Block Protocols

Block protocol support is also present in the Oracle ZFS Storage Appliance. LUNs can be exported via iSCSI or FC. SRP and iSER (via 40 Gb InfiniBand) are also supported. For iSCSI, both RADIUS and iSNS discovery are supported. The Oracle ZFS Storage Appliance can serve as a FC target or an FC initiator to facilitate backups.

NDMP for Backup

NDMP is also supported for backups in DMA environments. NDMP backups can be produced in dump, tar, or zfs formats. Note that, while zfs NDMP format may provide a performance benefit, it does not support DAR, so direct file access is not possible with zfs NDMP format specifically. The dump or tar format must be used if DAR support is required.

Encryption

The Oracle ZFS Storage Appliance supports Encryption on the following models: ZS3-4, ZS4-4, ZS5-2, ZS5-4, and ZS7-2. (Contact your sales representative for the latest information). Encryption happens as a fully inline process upon ingest, so all data-at-rest is encrypted. The technology used is a highly secure AES 128/192/256-bit algorithm with a two-tier architecture. The first level encrypts the data volume, the second level then encrypts that with another 256-bit encryption key. Encryption keys can be stored either locally within the ZFS key manager or centrally within the Oracle Key Management System (OKM). This provides robust privacy protection against security breaches and can help data centers to meet security requirements. Encryption can be enabled when a project or share/LUN is created for granularity in implementation and efficiency in administration or, more comprehensively, when a pool is created. Many competitive options also require expensive, specialized self-encrypting drives whereas Oracle ZFS Storage Appliance's encryption is controller-based, drive-independent, and very flexible, yet easy to use.

Management, Analytics, and Diagnostic Tools

The Oracle ZFS Storage Appliance includes an advanced command line interface (CLI) and browser user interface (BUI). These interfaces contain the same management options and are designed to mask the complexity of the underlying OS while still offering a powerful and deep command set. Most common administrative tasks can be accomplished quickly and easily with just a few commands or mouse clicks. The BUI also offers a built-in visual, industry-leading analytics package based on DTrace that runs in the storage controller OS. This analytics package gives unparalleled visibility into the entire storage stack—all the way down to disk and all the way up to client network interfaces, including cache statistics, CPU metrics, and many other parameters. This analytics package is extremely useful to identify any bottlenecks and tune the overall system for optimal performance. It is also very helpful in troubleshooting situations, along with a client's system administration staff, as the information can help the storage administrator to distinguish clearly upstream issues. This is particularly useful, for example, in large-scale virtualization environments where one VM out of thousands has or is causing a performance issue. For further information on analytics, please see the Analytics Guide on the Oracle ZFS Storage Appliance Product Documentation website.

The Oracle ZFS Storage Appliance also offers a rich scripting environment based on ECMAScript, which initially was based on JavaScript. Workflows can be used to store scripts in the appliance, and unlike many competitive frameworks that run on external machines, the Oracle ZFS Storage Appliance can take inputs from users or other workflows. Workflows can be invoked via either the BUI or CLI, or by system alerts and timers. Scripting is a powerful way for you to automate complex but repetitive tasks allowing for customization. The Oracle ZFS Storage Appliance also offers an advanced RESTful management API, so that administrative functions can be integrated into existing custom management tools by making standard REST calls to the storage.

Conclusion

The Oracle ZFS Storage Appliance is designed to extract maximum storage performance from standard enterprise-grade hardware while providing robust data protection, management simplicity, and compelling economics. The unique architecture, based upon the Hybrid Storage Pool model and wide variety of advanced data services make the Oracle ZFS Storage Appliance an excellent choice for a wide variety of enterprise storage workloads that demand high performance.

Related Links

Oracle ZFS Storage Appliance website: http://www.oracle.com/zfsstorage

Oracle Technology Network, Oracle ZFS Storage Appliance page: http://www.oracle.com/technetwork/server-storage/sun-unified-storage/overview/index.html

ZS9-2 Product Data Sheet: https://www.oracle.com/a/ocom/docs/storage/oracle-zfs-storage-appliance-datasheet.pdf

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

ORACLE

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

March 2024