ORACLE
SOLARIS

An Oracle Technical White Paper
July 2014

Oracle Solaris ZFS Data Management

ORACLE

Introduction 1
Overview 2
ZFS Benefits 2
Oracle Solaris 11 2
ZFS Changes the Way Data Is Managed 3
Simplified Pooled Storage Model 3
Easy Administration 4
Data Integrity Model 4
Copy-on-Write 4
End-to-End Checksums 5
Self-Healing Data 5
Redundant Copies of Data 5
Scalability 6
ZFS and Solid-State Storage 6
ZFS Boot Environments 6
ZFS Data Services 7
ZFS Snapshots and Clones 7
ZFS Snapshot Capabilities 7
ZFS Security Features 8
ZFS Encryption 8
ZFS Data Reduction 8
ZFS Deduplication 9
ZFS Compression 9
Data Sharing 9
Oracle ZFS Storage Appliances 10
Conclusion 10
For More Information 11

Introduction

This paper discusses Oracle Solaris ZFS, a powerful file system that combines file system capabilities with storage features that are traditionally offered by a volume manager. ZFS is the default file system in Oracle Solaris 11 and includes integrated data services, such as compression, encryption, and snapshot and cloning.

This paper describes ZFS features and how ZFS adds value in today's data centers. Oracle Solaris and ZFS are the foundation of Oracle ZFS Storage Appliance, so that product line is also described.

Oracle Solaris 11 is a modern cloud infrastructure and ZFS bring many advantages to managing cloud storage and data. These features are described as well.

Overview

Industry experts agree that storage capacity requirements typically double every 18 months, creating storage management challenges in the data center. Using traditional products to manage storage can be complex and is often costly. Administrators are typically forced to configure multiple software layers consisting of volume managers, file systems, and access protocols. As a result, organizations are seeking better ways to cost-effectively manage enterprise data growth and data center complexity.

ZFS provides a simple management interface, which is a scalable and robust solution designed to reduce the total cost of ownership, simplify system and storage management, and provide high performance without the need for custom tuning of business applications. At the same time, it can help increase storage utilization through consolidation and by ensuring file system availability.

ZFS key differentiators are

- Replaces both traditional file system and volume manager products
- Provides end-to-end data integrity and data redundancy
- Integrates data services such as data reduction, encryption, and migration

ZFS Benefits

Managing a ZFS storage environment is greatly reduced compared to conventional file system management because ZFS does the following:

- Simplifies and reduces storage management tasks
- Increases storage agility and data protection
- Delivers superior performance and availability

Oracle Solaris 11

Oracle Solaris 11 is an enterprise-grade OS developed and hardened over two decades of data center use. It is a key component in the complete applications-to-disk stack that Oracle offers, providing system services and virtualization components as well.

Oracle Solaris is renowned for its robustness, reliability, security, and scalability on both the x86 and SPARC platforms. It also has unique technologies in the areas of fault management, system observability, virtualization, and data management (the topic of this paper).

Not surprisingly, much of the performance, scalability, observability, and reliability features in Oracle Solaris were developed in the context of the needs of relational databases with their unique patterns of I/O and the requirement for a platform on which to base a strongly consistent data model. The clustering abilities of Oracle Solaris combined with its virtualization technologies, such as Oracle Solaris Zones, make it a key platform for multi-instance database environments and database clustering.

ZFS Changes the Way Data Is Managed

The following are the primary design goals for ZFS:

- **Simple**: The goal is to remove the complexity of file system and volume manager administration—tasks that have traditionally been costly, error prone, and not scalable to storage systems with thousands of devices.
- **Easy administration**: A pooled storage design and a simple yet flexible command set make it easy to provision and manage the storage environment.
- **Data integrity**: In petabyte storage configurations for which ZFS is aimed, device failure and degradation are a given, so end-to-end data integrity and the ability to recover seamlessly and quickly from component failures are a must.
- **Ability to scale up or scale down**: ZFS is a 128-bit file system with no practical limits on the number of files, directories, and file systems, and no practical limits on the amount of physical storage that can be addressed. Performance can be scaled up as well by putting applications on faster solid-state drives (SSDs). Storage footprints can be scaled down 3X by using ZFS compression.

Simplified Pooled Storage Model

Central to the design of ZFS is the storage pool. Traditional volume managers aggregate physical disks into volumes that provide the basic building blocks for file systems. There is a one-to-one mapping between a volume and the traditional file system built on top of it.

Modifying a traditional file system, changing its size or read/write characteristics, adding or removing underlying storage hardware, or reprovisioning in this scenario requires the data to be backed up to some other medium, and the file system to be torn down and rebuilt after hardware modifications are made. Data is then restored after recreating the volume and file system.

Current and estimated trends of data ownership make this data management model unfeasible.

Simplicity decreases the likelihood of administrative errors. With ZFS, only two simple commands create a pool and the file systems within it. The physical storage is shared between all file systems within the pool. New storage devices can be added on the fly while file systems are online. The simple design provides simple, error-free management.

Easy Administration

ZFS provides a compact but expressively simple command set for creating ZFS storage pools and file systems and for dealing with all the ZFS data services—such as compression, encryption, and quotas and reservations—for example:

```
# zpool create pond mirror c1t0d0 c2t0d0 mirror c1t1d0 c2t1d0 spare c3t0d0
# zfs create pond/amy
# zfs create pond/rory
# zfs set –o quota=20g pond/rory
# zfs create –o compression=on pond/storage
```

In addition, ZFS is the underlying technology for the following Oracle Solaris 11 features:

- Boot environment updates and management
- System migration and recovery archives
- Virtual machine (VM) data security and migration

The ability to trivially create snapshots of boot environments enables ZFS features, such as rolling upgrades, with the ability to easily roll back to the previous boot environment, if necessary. This is a key feature when rapidly deploying cloud development, test, and production environments.

Data Integrity Model

ZFS guarantees the integrity of data from applications to disk. It achieves this goal in a number of ways. Its copy-on-write model (described in the next section) ensures that modified data is never written over the data it modifies. This methodology allows for transactional semantics. Either the old data is correct on disk or the new data is still valid. There is no intermediate or unknown state, so recovery and consistency checks are not necessary.

Copy-on-Write

ZFS uses copy-on-write semantics so that data is never overwritten in place. Figure 1 shows the creation of new data blocks with modified data in them rather than the overwriting of existing data blocks that the new data modifies. ZFS then creates new pointers to the new data and a new master block (uber-block) that points to the modified data tree. Only then does it move to using the new uberblock and tree. In addition to providing data integrity, having new and previous versions of the data on disk allows for services, such as snapshots to be implemented very efficiently.

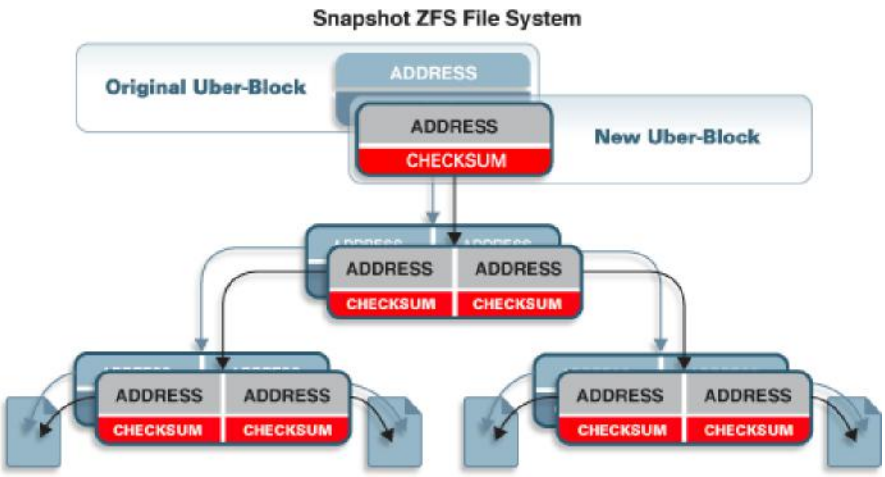

Figure 1. ZFS copy-on-write methodology

End-to-End Checksums

All data and metadata blocks in the ZFS file system are checksummed. The block checksum is in the parent block, a significant difference from the usual practice where checksums are stored with the data. ZFS detects if the data has been corrupted and repairs the data on the fly, as shown in Figure 1.

Every data block's checksum is stored in its parent's block. The checksum for that parent is stored in its parent's block, all way up the tree. This provides a whole range of data integrity benefits. There is no scope for silent data corruption and no possibility of system failure due to corrupted metadata.

Self-Healing Data

When a data block is found to be bad, the bad block can be repaired on the fly from its mirrored block. This concept, of course, extends to other RAID types that ZFS supports.

Redundant Copies of Data

While the size of disk drives has increased, their performance has not. Repopulation times have significantly increased, opening a window of vulnerability where one drive is being repopulated while another fails.

ZFS supports mirroring and several other RAID implementations. In particular, triple-parity RAID, known as RAIDZ3, is a RAID 6 implementation. It extends double-parity RAIDZ, which allows for two drive failures, to three drives' worth of parity. Triple-parity RAID protects against up to three drive failures.

ZFS built-in redundancy means that you can build an OpenStack cloud solution without the sprawl of clustering systems together to provide data redundancy.

Scalability

ZFS is implemented using 128-bit addressing regardless of the operating system's address size. Oracle Solaris 11 is a 64-bit operating system. It is built for petabyte scale and content. There are no practical limits on its operation other than constraints imposed by the available hardware.

ZFS scalability is about more than just a size. It's about scaling up when you need to and scaling down when you need to. ZFS can scale up performance as well, by adding fast storage such as SSDs. In addition, you can scale down storage footprints by using ZFS compression.

ZFS and Solid-State Storage

The pooled storage model allows ZFS to incorporate SSDs in a very efficient manner, not simply as "faster disks." SSDs, used as read and write caches, are managed by intelligent prefetch and caching algorithms within the pool.

An SSD-backed write cache allows applications, such as NFS file systems and databases that require direct synchronous I/O, to have their writes initially flushed to very fast SSDs and later destaged to spinning disk. This provides very low write latency to the application and speeds up the response time.

On the read side of the data path, an SSD-backed adaptive replacement cache allows for intelligent prefetch and caching of data blocks.

SSDs can be added to the pool seamlessly in the same way as traditional disks, for example:

```
# zpool add pond log c4t1d0
```

ZFS Boot Environments

Changes to the Oracle Solaris boot and upgrade environment provide some of the most significant advantages provided in Oracle Solaris 11 in terms of increased availability and decreased risk.

The use of ZFS and its enabling technologies for the boot disk began in Oracle Solaris 10. ZFS combined with Oracle Solaris 10 Live Upgrade provided major availability and performance enhancements over the UFS version of Live Upgrade. ZFS dramatically decreases the time to create a boot environment, which is a copy of the environment that Oracle Solaris needs in order to run. Using boot environments minimizes the risk of making changes to system software and maximizes availability by reducing planned downtime.

Oracle Solaris 11 boot environments provide integration with the Oracle Solaris Image Packaging System and Oracle Solaris Zones, and Oracle Solaris also includes a new boot environment management tool. Given these advances—and the clear advantage of a file system that can make snapshots rather than be constrained to the add-on approach of Oracle Solaris Live Upgrade—there was no reason to continue to offer support for booting Oracle Solaris 11 from a UFS boot disk.

ZFS Data Services

ZFS provides a robust and integrated set of data services that are provided with the base Oracle Solaris OS.

ZFS Snapshots and Clones

A snapshot is a read-only, point-in-time copy of a file system. It is useable as part of a backup strategy and as a tool for migration and upgrade, providing a safety net when changes are being made to the system. The ZFS copy-on-write capability makes snapshots so space-efficient they are essentially "free" from a storage space point of view. Only changes to the data are tracked. The ability to create a snapshot can be delegated by administrators to the users of the data.

Clones are simply writeable snapshots turned back into file systems. Clones are ideal for sharing many private copies of data and have wide-ranging uses in areas such as software installation and upgrade, development workspaces, and diskless clients. Clones are a huge convenience to cloud administrators and developers for cloning development, test, and production environments.

ZFS Snapshot Capabilities

ZFS allows you to take a snapshot and use standard UNIX commands or APIs to send the ZFS snapshot to a local pool or over a network to a pool on another system. This feature is used extensively as a means of asynchronous cross-site or cross-system replication. The technology is so efficient that it is used for both remote replication and by users with automatic snapshot recovery.

The following commands create a recursive snapshot of the `pond` pool and then the `zfs send –rnv` command displays the individual snapshots in the recursive snapshot and the estimated stream size. The second `zfs send` command includes the `pv` command to identify the send stream status during the send operation and receives the recursive snapshot on a remote system.

```
# zfs snapshot –r pond@snap1
# zfs send -rnv pond@snap1
sending full stream to pond@snap1
sending full stream to pond/river@snap1
sending full stream to pond/storage@snap1
sending full stream to pond/rory@snap1
sending full stream to pond/amy@snap1
estimated stream size: 16.0G
# zfs send pond@snap1 | pv | ssh root@tardis zfs recv
bpool/pond
Password: xxxxxx
# zfs send -rv pond@snap1 | pv | zfs recv tank/pond
sending full stream to pond@snap1
sending full stream to pond/river@snap1
sending full stream to pond/storage@snap1
sending full stream to pond/rory@snap1
sending full stream to pond/amy@snap1
estimated stream size: 16.0G
1.27GB 0:00:16 [90.2MB/s] [              <=>
```

ZFS Security Features

ZFS provides standard UNIX permissions. It also implements access control lists (ACLs) and the framework around ACLs. It is tightly integrated into the Oracle Solaris Zones virtualization environment providing further security. ZFS also provides delegated administration where you can provide ZFS-related permissions to your administrative staff or to users without having to provide superuser access to the system. For example, you could allow tenant administrators to snapshot and clone virtual environments while restricting their access to the larger multitenant environment.

ZFS Encryption

ZFS provides on-disk encryption in which data is encoded on disk for privacy. Creating an encrypted ZFS dataset is as simple as setting a property at creation time.

The data owner uses a key to access the encoded data. The encryption property is enabled when a ZFS file system is created. You are prompted for a passphrase by default, as shown below. A wrapping key is either taken from a file (in raw or hexadecimal format) or it is derived from the passphrase.

```
# zfs create –o encryption=on pond/river
Enter passphrase for 'pond/river': xxxxxxxx
Enter again: xxxxxxxx
```

The wrapping key encrypts the data encryption keys. It is passed from the `zfs` command to the kernel when the encrypted file system is created. The default encryption algorithm is AES-128-CCM. Others are available and different encryption methods are possible with each file system in a storage pool. In addition, ZFS data encryption can leverage the Oracle Solaris cryptographic framework and take advantage of cryptographic hardware support for best encryption performance.

ZFS encryption is a key feature for keeping data safe in a multitenant cloud environment. In addition, VM tenant file systems can be locked down with read-only ZFS file systems.

ZFS Data Reduction

Management of data and the infrastructure required to store it has become a top priority in today's data center. On average, customers need to store 15 copies of primary data for various uses, such as backup, data recovery, test, development, and other uses. Compliance legislation is also driving customers to store data for longer periods of time. Energy and power costs are now one of the top operational expenses for enterprise data centers. To address these critical issues, ZFS offers enabling technologies that help customers better manage their data as well as reduce their overall storage footprint.

Data reduction technologies are provided in ZFS is to help customers achieve greater efficiency in terms of reducing the amount of physical storage consumed and increasing data throughput—the fastest data transfers being those that need not occur at all.

ZFS Deduplication

ZFS is the only general-purpose file system with built-in deduplication capabilities that can greatly reduce the amount of data that is stored on physical disk.

ZFS deduplication is performed at the block level, meaning that only unique blocks are stored even if they are shared by two or more separate files. The process is performed inline, meaning that data is deduplicated as it is written to the system. This methodology contrasts with post-process deduplication, which is common in competitor implementations. Post-process deduplication causes the entire dataset to be written to disk initially and then the dataset is deduplicated at a later, configurable time.

Using ZFS deduplication is recommended only when you have ensured that your data is dedupable and your system has the memory capacity to support deduping data.

ZFS Compression

ZFS data compression reduces the amount of disk space used by employing one of four compression algorithms. This allows tradeoffs between the amount of CPU required for compression and the resulting compression efficiency. Compression also provides increased throughput and performance within the system due to the fact that less data is being written to and read from disk, resulting in fewer I/O operations.

If your data is compressable, typical disk space savings by enabling ZFS compression are in the 3X range.

Data Sharing

ZFS provides a variety of data sharing protocols.

- Data can be shared over NFS in the data center over fast lower-layer data center transports, such as gigabit Ethernet.
- Where tight integration with Microsoft client infrastructure is needed, Oracle Solaris has a fully compliant Server Message Block (SMB) stack in the kernel.

For block storage, you can create a ZFS volume (an object that looks like a storage LUN) and the volume can be shared over iSCSI, Fibre Channel, and other storage transports as a block device for applications and virtualization environments that require block storage. Block-device provisioning is also useful in data centers that want to leverage their existing SAN infrastructure for cloud storage.

These protocols are fully available in the base Oracle Solaris operating system. Higher-level web protocols (for example, HTTP, REST, SOAP, and Webdav) can be layered on top of ZFS within Oracle Solaris.

Oracle ZFS Storage Appliances

The data center can take advantage of ZFS in two ways:

- ZFS is the base file system and data management technology of Oracle Solaris 11 as well as Oracle Solaris 10.
- The Oracle ZFS Storage Appliance product line, which consists of integrated storage systems, is built on Oracle Solaris and ZFS technology. There are multiple models in the product line, scaling from entry-level/workgroup systems to highly available, enterprise-class storage systems. All models support various block and file protocols and ZFS data services, including those discussed previously.

Oracle ZFS Storage Appliances fully support Oracle's key operating platforms and business applications, such as Oracle Database, Oracle Fusion Middleware, and Oracle Applications. Also supported are critical business applications, such as VMware, Microsoft applications, and others.

Two key capabilities of Oracle ZFS Storage Appliances are

- A browser-based user interface (BUI), which provides an easy-to-use interface for managing your pools and file systems.
- A diagnostic framework based on the Oracle Solaris DTrace analysis tool. DTrace on Oracle ZFS Storage Appliances provides a comprehensive, real-time graphical view of workload and performance information presented through an intuitive user interface. Oracle ZFS Storage Appliances are the only storage system on the market with this unique and simple-to-use functionality.

Some ZFS data services, supplied at no extra cost, are separately licensed in Oracle ZFS Storage Appliances.

Conclusion

Oracle Solaris 11 customers can now benefit from a single, integrated data management solution across the data center and beyond. Oracle Solaris offers a complete cloud infrastructure platform, built-in virtualization, one set of installation and configuration tools, security and compliance reporting tools, and a single vendor for support and troubleshooting. These capabilities greatly simplify storage administration and eliminate the need for third-party volume managers and file systems.

ZFS technology is tightly integrated with the Oracle Solaris 11 Image Packaging System, Oracle Solaris Zones for virtualization, the operating system's data sharing protocols and its fault management architecture, and other Oracle technologies. It presents a comprehensive and well-tested solution to help organizations deploy their applications, databases, and storage in the enterprise cloud environment.

And last but certainly not least, there is no additional license or cost for ZFS or any of the data services it offers.

For More Information

- Oracle Solaris 11 download: http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html
- Oracle Solaris 11 how-to guides: http://www.oracle.com/technetwork/server-storage/solaris11/documentation/how-to-517481.html
- Oracle Solaris 11 resources: http://www.oracle.com/technetwork/server-storage/solaris11/overview/index.html

Oracle Solaris ZFS Data Management
July 2014, Version 2.0
Author: Dominic Kay
Updated by: Cindy Swearingen

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

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

Hardware and Software, Engineered to Work Together