

ZFS / OpenStorage

Lee Marzke

IT Consultant - 4AERO

ZFS / OpenStorage

- ***About the Zettabyte Filesystem***

- Developed by Sun Microsystems (now Oracle)
 - "The final word in Filesystems"
- License
- Quick Overview of Nexenta GUI / CLI
- New Terms / Definitions
- Features and Limitations
- Platforms and Products

- ***Demo***

ZFS / OpenStorage

- *Why ?*
 - Replace Expensive Proprietary Storage (SAN)
 - Second Tier storage
 - Primary NetApp / EMC / Equallogic / HP ?
 - Use off-the-shelf 64bit hardware
 - Proveable integrity (top to bottom checksums)
 - Transactional – always consistent on disk
 - Unlimited file size and numbers

ZFS / OpenStorage

- **ZFS**

- ZFS 1995: The last word in Filesystems (Sept 2008)
Jeff Bonwick and Bill Moore [15] (3 hour video)

- ***Open Storage***

- As a general term, open storage refers to storage systems built with an open architecture using industry-standard hardware and open-source software.
- Fishworks: Now it can be told: [12],[13,17] 2006/2008
Development of 1st NAS appliance on ZFS at Sun

ZFS / OpenStorage

- **ZFS License**
 - Open-Source CDDL (not compatible with GPL)
 - Linux userland (via FUSE module) implemented
 - Read performance 50%
 - Write performance 50% to 5% (random worse)
 - Linux kernel implementation not possible with current license. But this may change. See [1] (LLNL)

ZFS / OpenStorage

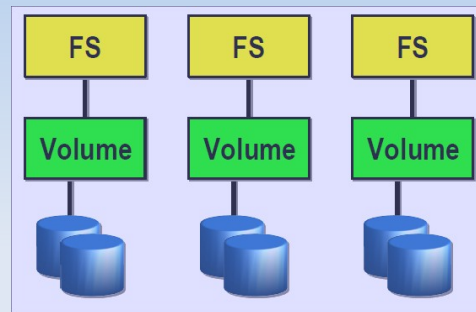
- ***Nexenta Overview***

- Nexenta Core (Solaris Kernel + Ubuntu userland)
 - Open source – free (CLI only)
- NexentaStor (Commercial GUI + support)
 - Build on top of Nexenta Core
 - Free for up to 18TB used storage
 - Commercial: Approx \$100/TB - \$200/TB

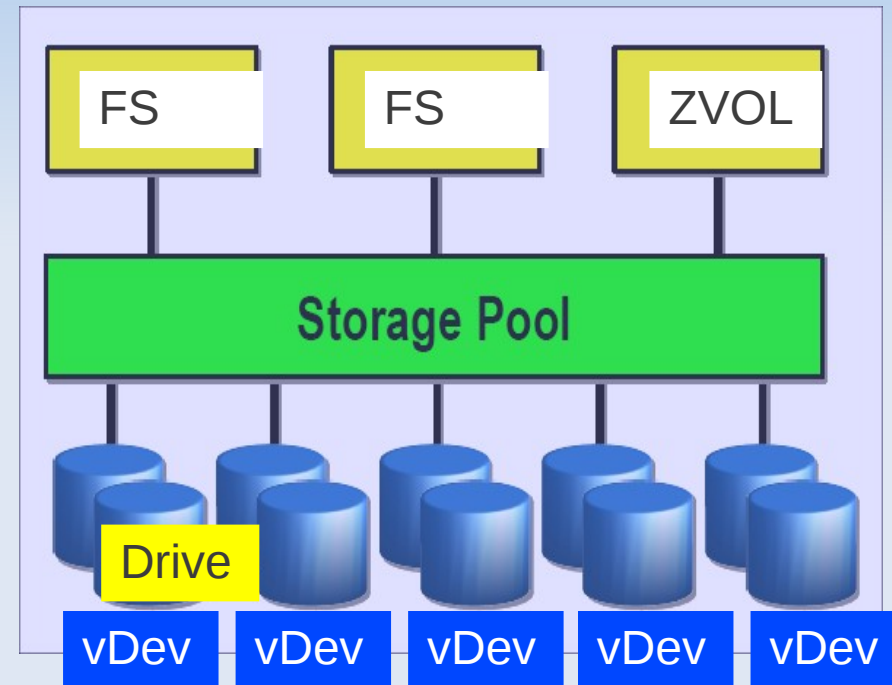
ZFS / OpenStorage

■ *Terms*

- No Volumes!
- Drive
- vDev
- Zpool
- DataSet
 - FileSystem
 - Clone / Snapshot
 - Zvol



Traditional



ZFS

ZFS / OpenStorage

- ***Nexenta Overview***
 - Demo appliance
 - (1) zPool - consisting of 8 devices
 - (3) mirrored vDevs
 - (1) cache SSD (read cache)
 - (1) log SSD (write log)
 - SSD are optional to speed up read/write

ZFS / OpenStorage

- ***Terms***

- RaidZ – Parity drives added to survive failure
Stripe width variable
 - RaidZ1 – one parity device
 - RaidZ2 – two parity
 - RaidZ3 – three parity
- Resilver – process of creating/recreating a mirror device (mirror resync) or failed data or parity device

ZFS / OpenStorage

■ *Terms*

- Cache Device – improves reads
 - ARC in RAM
 - L2ARC in SSD
- Log Device – copy of transaction, speeds writes
 - ZFS Intent Log (ZIL) on SSD
- MLC – multi-level cell SSD
cheaper, best for reading, 10,000 writes life
- SLC – single-level cell SSD
best write performance, 100,000 writes life

ZFS / OpenStorage

- ***Terms***

- Hybrid Storage Management (HSM) Pool
 - Slow Primary disk
 - Medium performanc SSD cache (L2ARC)
 - Fast RAM cache (ARC)
- ZFS Intent Log (ZIL)
 - ZFS writes are transactions
 - Write metadata and data to log
 - Commit log to main disk
 - For performance put ZIL on separate fast disk

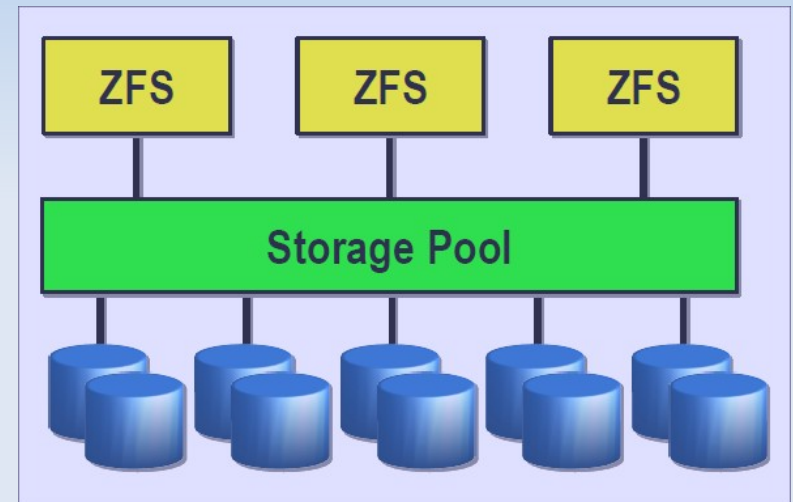
ZFS / OpenStorage

- ***Development of ZFS at Sun***
 - ZFS: The last word in Filesystems (Sept 2008)
Jeff Bonwick and Bill Moore [15] (3 hour video)
 - Fishworks: Now it can be told: [12]
Development of 1st NAS appliance at Sun
to make use of free ZFS stack on OpenSolaris

ZFS / OpenStorage

■ *WHY ZFS: Pooled Storage*

- No volumes to resize/manage
- Pools made up of 1 or more Vdev's
- Vdev's in pool are striped
- Vdev's can't be removed with present ZFS version



ZFS / OpenStorage

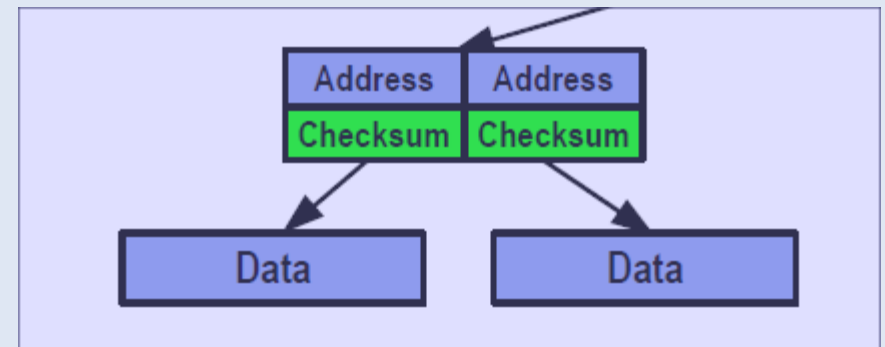
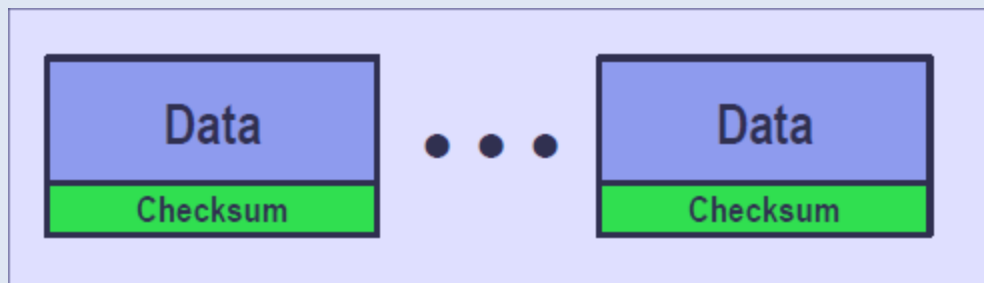
- ***Why ZFS: Transactions***
 - Transactions (meta-data and data)
 - Data first written to ZFS intent log (ZIL)
 - Then ZIL flushed to Disk
 - ZIL recommended to use dedicated SSD or Disk
 - Data on disk is ALWAYS consistent !
 - Except for TLER misfeature.
 - Disk sectors remapped – disk offline for seconds to possibly minutes

ZFS / OpenStorage

- ***Why ZFS: Transactions***
 - How 2 writes for each block can be faster:
 - Random writes are slowest
 - ZIL lives on a fast SSD device so random
 - writes to ZIL are fast
 - ZIL flushed each 5 seconds
 - ZIL flushed sequentially to disk

ZFS / OpenStorage

- ***Why ZFS: Checkums***
 - Traditional – checksum stored with block
 - ZFS – block checksum stored in meta-data
 - Each directory checksums contents below it (Merkle tree)



ZFS / OpenStorage

- ***Why ZFS***

- Online Scrub instead of fsck
- Scrub detects silent data corruption and repairs it.
 - Scrub reads all blocks, all parity, all meta-data.
 - Verify againsts 256 bit checksum / repair as needed
 - Runs at low priority in background.
- Recommended Scrub
 - Scrub 1/wk for Consumer drives
 - Scrub 1/month for Enterprise drives

ZFS / OpenStorage

- ***Why ZFS: Data Integrity***
 - Some blocks are more important (root meta-data)
 - ZFS adds Ditto Blocks [4]
 - Multiple copies on disk (not across stripes)
 - User data – one copy
 - FS meta data – two copies
 - Global meta data – three copies

ZFS / OpenStorage

- **Why ZFS: COW (Copy – on - Write)**

- For RaidZx - no edit in place
NO RAID5 write hole!
- New copy of Entire stripe (data + parity) written
- Variable width stripes

- Because of ZIL
everything written 2X
- This isn't BAD ! Why ?

		Disk				
		A	B	C	D	E
LBA	0	P ₀	D ₀	D ₂	D ₄	D ₆
1	P ₁	D ₁	D ₃	D ₅	D ₇	
2	P ₀	D ₀	D ₁	D ₂	P ₀	
3	D ₀	D ₁	D ₂	P ₀	D ₀	
4	P ₀	D ₀	D ₁	D ₂	P ₀	

ZFS / OpenStorage

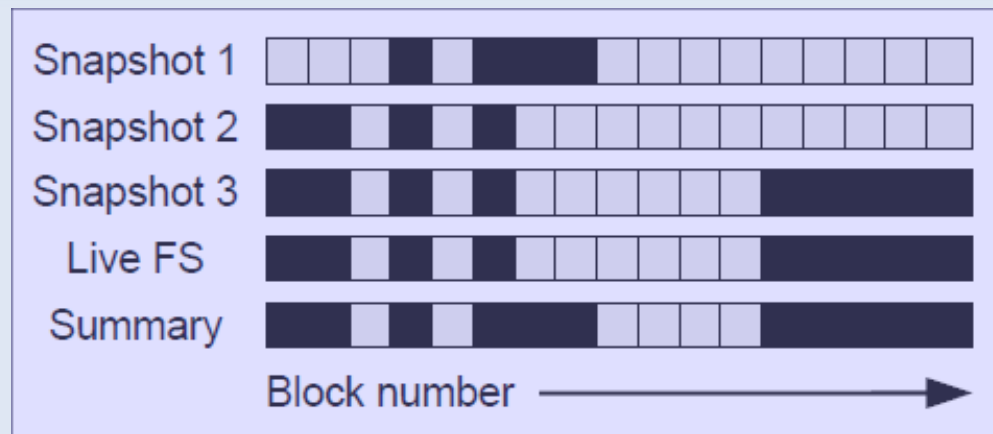
- **Why ZFS: COW (Copy – on - Write)**
 - Most SAN writes are highly random
 - COW writes to ZIL on very fast SSD
 - After 5 seconds ZIL flushed "sequentially" to main disk.
 - This turns random writes into sequential.

		Disk				
		A	B	C	D	E
LBA	0	P ₀	D ₀	D ₂	D ₄	D ₆
	1	P ₁	D ₁	D ₃	D ₅	D ₇
	2	P ₀	D ₀	D ₁	D ₂	P ₀
	3	D ₀	D ₁	D ₂	P ₀	D ₀
	4	P ₀	D ₀	D ₁	D ₂	P ₀

ZFS / OpenStorage

- **Why ZFS: Snapshots**

- Other FS: Bitmap per snapshot [15] p11
 - $O(N)$ space, $O(N)$ time (create, destroy)



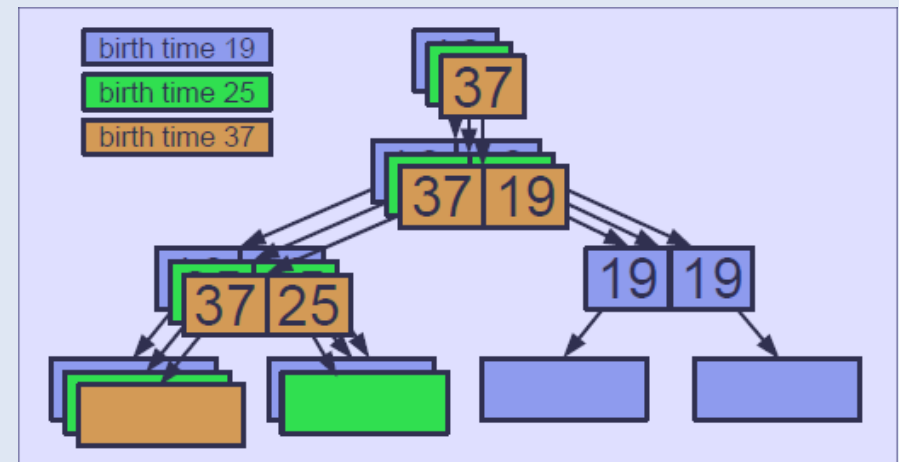
BITMAPS

Pointer + Birthtimes

ZFS / OpenStorage

- **Why ZFS: Snapshots**

- ZFS: Pointer with checksums, birthtimes (COW)
 - O(1) space, O(1) create time
 - O(Δ) delete (birthtime tree-walk)



BITMAPS

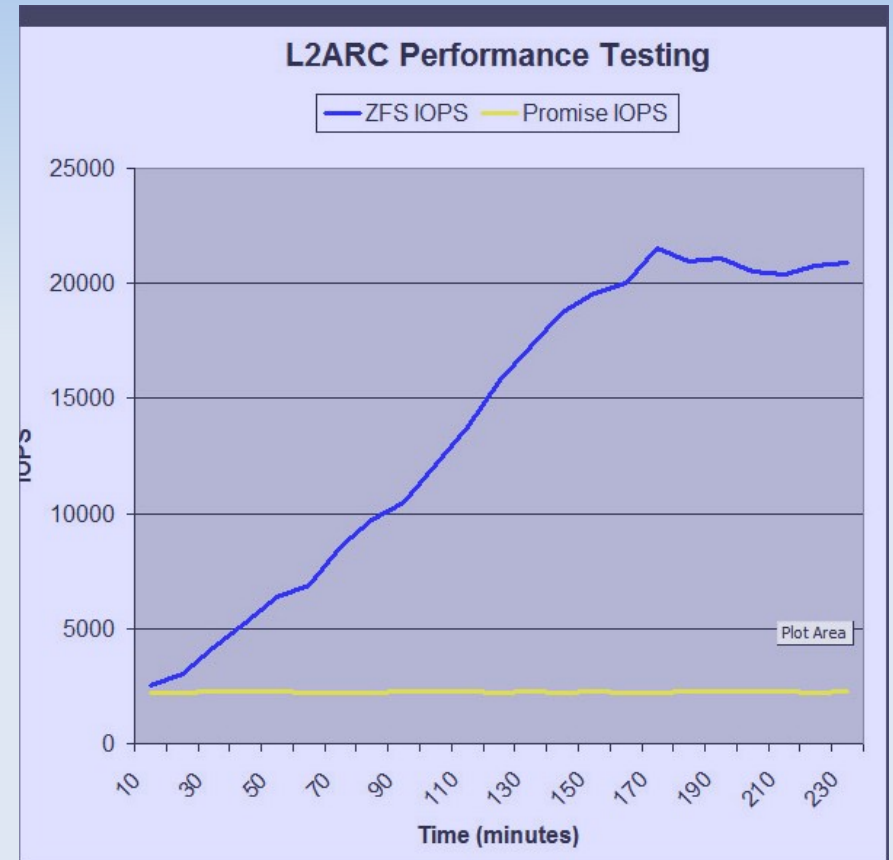
Pointer + Birthtimes

ZFS / OpenStorage

- ***Pooled HSM Storage (with cache+log)***
 - Typical Cache Setup for 10TB pool
 - 32GB RAM – 4GB for OS leaves 28GB ARC
 - (1) 500GB MLC SSD
 - Total of 0.5TB of cache
 - Use `arc_summary` for info
 - Typical ZIL Setup
 - Need to store (2) transaction groups of 5 seconds of write data (10 sec x 170Mb/s ~ 2GB required)
ZIL Mirror recommended.
 - Use `zilstat` to monitor performance

ZFS / OpenStorage

- ***L2ARC test [10]***
 - Promise 610i
16 x SATA
-vs- ZFS
 - 100% random read
 - Cache population rate limited to slow wear on SSD's



ZFS / OpenStorage

- ***Compression***

- Built in LZJB and GZIP
- Encryption TBD

- ***Dedup***

- Deduplication recently added
- Dedup lookup table should fit in RAM
 - Requires lots of RAM and ZIL on SSD

- ***Send / Recieve***

- ZFS snapshot Δ send/recieve (for replication)

ZFS / OpenStorage

- ***Limitations***

- Can't remove vDev's from pool
- Lots of RAM, and SSD's needed for good performance
- ZFS capacity: 256 quadrillion ZB (1ZB = 1 billion TB)

ZFS / OpenStorage

■ *Sample Commands*

- `zpool create tank mirror ctd0 c3d0`
- `zfs create tank/home`
- `zfs create tank/home/lee (fs)`
- `zfs set mountpoint=/export/home/lee tank/home/lee`
no need to manage /etc/exports, etc.
- `zfs snapshot tank/home/lee@tuesday`
- `Zfs clone tank/app1 tank/testing/fixapp1`

ZFS / OpenStorage

- Practical Implementations
 - OpenSolaris (discontinued by Oracle)
 - Nexenta OS (OpenSolaris kernel , Ubuntu userland)
 - NexentaStor
 - NexentaOS + Commercial GUI + Support
 - Free for first 18TB
 - Licensed by TB of storage thereafter
 - OpenIndiana
 - IllumOS Kernel - Fork of OpenSolaris
 - Many vendors migrating here, not in production

ZFS / OpenStorage

- Demo
 - NexentaStor (Free version for 18TB)
 - Running on OpenSource NexentaOS
Unlimited use – CL only

ZFS / OpenStorage

```
*****
*                               *
*       Nexenta Management Console                               *
*                               *
*       Version 3.1.0 (r9348)                                     *
*                               *
*     press  TAB-TAB to list and complete available options    *
*                               *
*     type   help    for help                                     *
*           exit    to exit local NMC, remote NMC, or group mode *
*           q[uit] or Ctrl-C  exit NMC dialogs                  *
*           q[uit] or Ctrl-C  exit NMC text viewer              *
*                               *
*           option -h      help on NMC options                  *
*           <any command> -h  help on any command               *
*           <any command> ?   brief summary                     *
*           help keyword [-q] locate NMC commands              *
*           help -k [-q]      same as above                     *
*           setup usage       combined 'setup' man pages        *
*           show usage        combined 'show' man pages         *
*                               *
*     type   help    and press TAB-TAB                          *
*                               *
*     Management GUI: https://172.16.236.128:2000/ *
*                               *
*****
```

ZFS / OpenStorage

The screenshot displays the Nexenta OpenStorage web interface. At the top, there is a navigation bar with links for 'About', 'Support', 'Register', 'Convert To Enterprise', and 'Help'. On the right, it says 'Welcome Administrator | Logout'. Below this is a main menu with 'Status', 'Settings', 'Data Management', and 'Analytics'. A secondary menu shows 'General', 'Storage', and 'Network'. On the far right, there are icons for 'Console', 'View log', and 'Jobs'.

The main content area is divided into two columns. The left column contains a sidebar with sections: 'Overall Status Information' (with links for Data Volumes, Replication services, Network services, and Fault Management services), 'Appliances' (listing 'nexenta' as 'This appliance'), and 'Appliance Groups' (with a 'Create' link).

The right column is titled 'GENERAL STATUS AND DETAILS: NEXENTA' and features a 'CPU and I/O Monitor' section. This section contains three gauges: 'CPU Utilization' (showing 30%), 'Network I/O, KB/Sec' (with 'IN' and 'OUT' gauges both at 11), and 'Disk I/O, KB/Sec' (with 'READ' at 0 and 'WRITE' at 240). Below the gauges, a note states: 'Currently gauges update every 5 seconds. You can change this interval [here](#)'.

Below the monitor section is a 'General Appliance Information' table:

Property	Value
Server Time	Thu Oct 6 06:54:20 2011
Time Zone	US/Eastern
Last System Boot	Wed Oct 5 20:02:06 2011
Load Average	0.33, 0.31, 0.38
NMS Version	3.1.0 (r9356)
NMC Version	3.1.0 (r9348)
NMV Version	3.1.0 (r9371)
OS Version	3.1.1
Total Memory	2999MB
Free Memory	1859MB

ZFS / OpenStorage

The screenshot displays the Nexenta OpenStorage web interface. The top navigation bar includes links for About, Support, Register, Convert To Enterprise, and Help, along with a user greeting 'Welcome Administrator' and a Logout link. The main navigation menu features tabs for Status, Settings, Data Management, and Analytics. Below this, there are icons for Data Sets, Shares, SCSI Target, Auto Services, and Runners, along with links for Console, View log, and Jobs.

The interface is divided into a left sidebar and a main content area. The sidebar contains three sections: Volumes, Folders, and Snapshots, each with 'Show' and 'Create' options. The main content area is titled 'VOLUME: TANK' and shows the following configuration details:

- Disks:** A table listing the disks in the volume pool.
- Hot Spares (only in available state):** A table listing available hot spare disks.
- Configurable Properties:** A section for setting various volume properties.

Disk	Size	Status	Errors (R/W/C)
mirror-1 (1 device)			
mirror-0 (1 device)			
mirror-0 (1 device)			
mirror-1 (1 device)			
logs (1 device)			
cache (1 device)			

Disk	Size	Volume
<input checked="" type="checkbox"/> c1t5d0	2.00 GB	tank

Configurable Properties:

- Description:** Demo pool (Optional volume description. Maximum length is 255 characters.)
- Deduplication:** off (Controls the deduplication option for the volume. If enabled, it will optimize use of duplicate copies of data. Default is off.)
- Compression:** on (Controls the compression algorithm used for this dataset. Default is "on". Setting compression to "on" uses the lzjb compression algorithm. The lzjb compression algorithm is optimized for performance while providing decent data compression. Currently, "gzip" is equivalent to "gzip-6".)
- Autoexpand:** off (Controls automatic pool expansion when the underlying LUN is grown.)
- Sync:** standard (Controls synchronous requests (standard - ensure all synchronous requests are written to stable storage; always - every file system transaction will be written and flushed to stable storage by system call return; disabled - synchronous requests are disabled). Default is standard.)

ZFS / OpenStorage

The screenshot displays the Nexenta OpenStorage web interface. At the top, the Nexenta logo is on the left, and navigation links for 'About', 'Support', 'Register', 'Convert To Enterprise', and 'Help' are on the right. A secondary navigation bar includes 'Status', 'Settings', 'Data Management', and 'Analytics'. Below this, a toolbar contains icons for 'Data Sets', 'Shares', 'SCSI Target', 'Auto Services', and 'Runners', along with 'Console', 'View log', and 'Jobs' options.

The main content area is titled 'SUMMARY INFORMATION : FOLDERS'. On the left, a sidebar menu shows 'Folders' (expanded) with options for 'Show Summary Information' and 'Create New Folder', and 'CIFS Server' (status: online) with options for 'Configure', 'Identity Mapping', 'Join AD/DNS Server', 'Join Workgroup', 'View Log', and 'Status'.

The central table lists folder information:

<input type="checkbox"/> Folder	Refer	Used	Avail	CIFS	NFS	FTP	RSYNC	WebDAV	Index	Delete
<input type="checkbox"/> tank/lee	31.00 KB	31.00 KB	2.87 GB	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Below the table, there is a search input field, 'Filter' and 'Delete selected' buttons, and a status indicator 'Results 1 - 1 (all)'.

ZFS / OpenStorage

1. http://www.phoronix.com/scan.php?page=article&item=zfs_fuse_performance
2. http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide
3. <http://zfs-on-fuse.blogspot.com/2007/04/zfs-in-linux-kernel.html>
4. http://blogs.oracle.com/bill/entry/ditto_blocks_the_amazing_tape
5. <http://hub.opensolaris.org/bin/view/Community+Group+zfs/basics>
6. http://www.opensolaris.org/os/community/zfs/docs/zfs_last.pdf
7. http://blogs.oracle.com/roch/entry/when_to_and_not_to
8. <http://constantin.glez.de/blog/2010/06/closer-look-zfs-vdevs-and-performance#vdevs>
9. <http://www.zfsbuild.com/2010/04/15/explanation-of-arc-and-l2arc/>
10. <http://www.zfsbuild.com/2010/07/30/testing-the-l2arc/>

ZFS / OpenStorage

- 11.** http://blogs.oracle.com/openstorage/entry/fishworks_virtualbox_tutorial
- 12.** http://blogs.oracle.com/bmc/entry/fishworks_now_it_can_be
- 13.** <http://blogs.oracle.com/bmc/resource/fishy-redacted.pdf>
- 14.** http://blogs.oracle.com/bonwick/entry/zfs_dedup
- 15.** http://blogs.oracle.com/video/entry/zfs_the_last_word_in
- 16.** **ZFS Admin:** <http://download.oracle.com/docs/cd/E19082-01/817-2271/>
- 17.** <http://www.informationweek.com/news/storage/systems/212001591>