The last word in file systems

Cracow, pkgsrcCon, 2016

ZFS

The Last Word In File Systems

Jeff Bonwick
Bill Moore

www.opensolaris.org/os/community/zfs

Mariusz Zaborski

<m.zaborski@wheelsystems.com>
<oshogbo@FreeBSD.org>

Outline

1. FreeBSD
2. ZFS
3. Lunch break

OpenZFS

FreeBSD®

Do you use FreeBSD?

Do you use ZFS?

FreeBSD... only “a few” words

FreeBSD - who are we, what do we do?

- modern, open-source operating systems
- 22 years old
- BSD license
- can’t use the Internet without using FreeBSD
- ~20,000 third party software ports
- A democratically run open source project

FreeBSD - who are we, what do we do?

- Complete integrated UNIX system
  - multi-processing, multi-threaded kernel
  - Intel/AMD 32/64-bit, ARM, MIPS, PPC sparc64, ia64
  - UNIX, POSIX, BSD programming interfaces
  - Multi-protocol network stack
    - IPv4, IPv6, IPsec, ATM, SCTP, 802.11, Bluetooth…

FreeBSD - who are we, what do we do?

- Complete integrated UNIX system
  - World
  - Kernel
  - Toolchain
  - Extensive documentation

Who uses FreeBSD? - **You!**

- WhatsApp
- Sony
- NetApp
- Apple
- Yahoo
- Netflix
- Juniper
- Wheel Systems
- Swisscom
- Microsoft Azure
- iXsystems
- Cisco
- Netgate
- NYI
- Verisign
- Dell KACE
- Dell/Compellent

Source: https://www.freebsdfoundation.org/freebsd/#whois

FreeBSD some features:

- UFS2
- ZFS
- DTrace
- Jails
- Bhyve
- LLVM/Clang/lldb
- Capsicm
- MAC Framework
- netmap
- Linuxulator

OpenZFS

ZFS

Are our hard disks invincible?

OpenZFS

Are our hard disks invincible?

Please write to offsets A, B, C, D

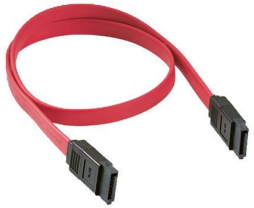
Please write to offsets A, B, D

Done, are you happy CPU?

https://www.youtube.com/watch?v=FaS2Svis3dA

Are our hard disks invincible?

Please write 0xFF 0x90 0xF9



Please write 0xFF 0x91 0xF9

Done, are you happy CPU?

Are our hard disks invincible?

OpenZFS

Thats joke, right?

OpenSSH CVE-2002-0083 - privilege escalation to root

OpenZFS

```
- if (id < 0 || id > channels_alloc) {
+ if (id < 0 || id >= channels_alloc) {
```

Source: Mike Perry, Seth Schoen

Thats joke, right?

OpenZFS

Assembly

```
cmpl $0x0,0x8(%ebp)          cmpl $0x0,0x8(%ebp)
js 16                        js 16
mov 0x4,%eax                 mov 0x4,%eax
cmp %eax,0x8(%ebp)           cmp %eax,0x8(%ebp)
jl 30                        jle 30
mov 0x8(%ebp),%eax           mov 0x8(%ebp),%eax
mov %eax,0x4(%esp)           mov %eax,0x4(%esp)
movl $0x4c,(%esp)            movl $0x4c,(%esp)
call 25                      call 25
```

Source: Mike Perry, Seth Schoen

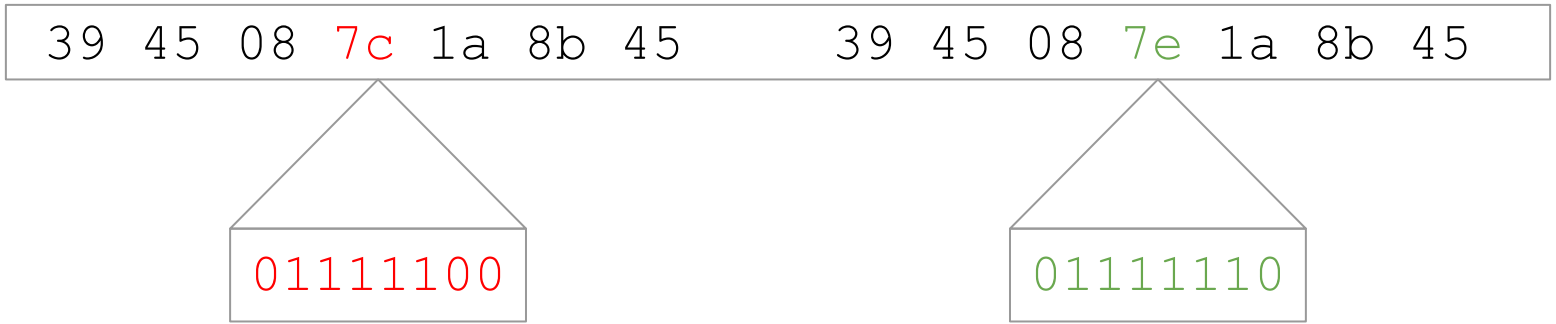
Thats joke, right?

Hex

Source: Mike Perry, Seth Schoen

Thats joke, right?

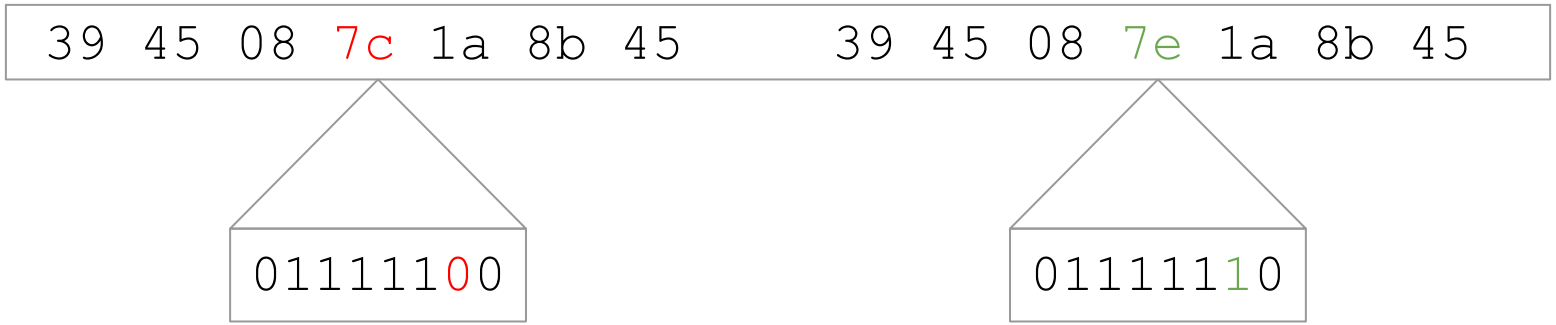
Binary



Source: Mike Perry, Seth Schoen

Thats joke, right?

Binary



Source: Mike Perry, Seth Schoen

ZFS - history

2001: Closed-source development of ZFS started with two engineers at Sun Microsystems.

2005: Source code was released as part of OpenSolaris.

2007: Apple started porting of ZFS to Mac OS X.

2008: A port to FreeBSD was released as part of FreeBSD 7.0.

2008: Development of a native ZFS Linux port started, known as ZFS on Linux.

ZFS - history

2010: OpenSolaris was discontinued, the last release was forked. Further development of ZFS on Solaris was no longer open source.

2010: illumos was founded as the truly open source successor to OpenSolaris.

2013: Official announcement of the OpenZFS project.

Why ZFS is ZFS?

1 kilobyte	1,000,000,000,000,000,000
1 megabyte	1,000,000,000,000,000,000
1 gigabyte	1,000,000,000,000,000,000
1 terabyte	1,000,000,000,000,000,000
1 petabyte	1,000,000,000,000,000,000
1 exabyte	1,000,000,000,000,000,000
1 zettabyte	1,000,000,000,000,000,000

Why ZFS is ZFS?

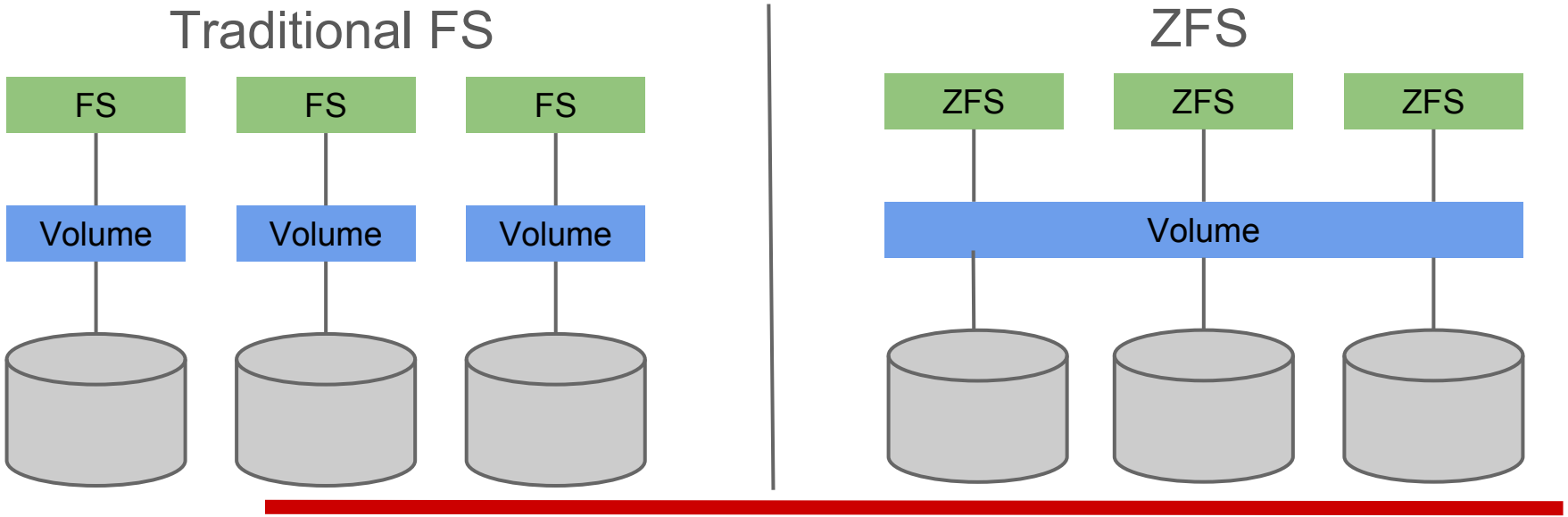
1 kilobyte	1,000,000,000,000,000,000
1 megabyte	1,000,000,000,000,000,000
1 gigabyte	1,000,000,000,000,000,000
1 terabyte	1,000,000,000,000,000,000
1 petabyte	1,000,000,000,000,000,000
1 exabyte	1,000,000,000,000,000,000
1 zettabyte	1,000,000,000,000,000,000

OpenZFS

160Gb ~ 275$

ZFS - overview

- Storage pool, integrated volume manager.
- Dynamic stripping.



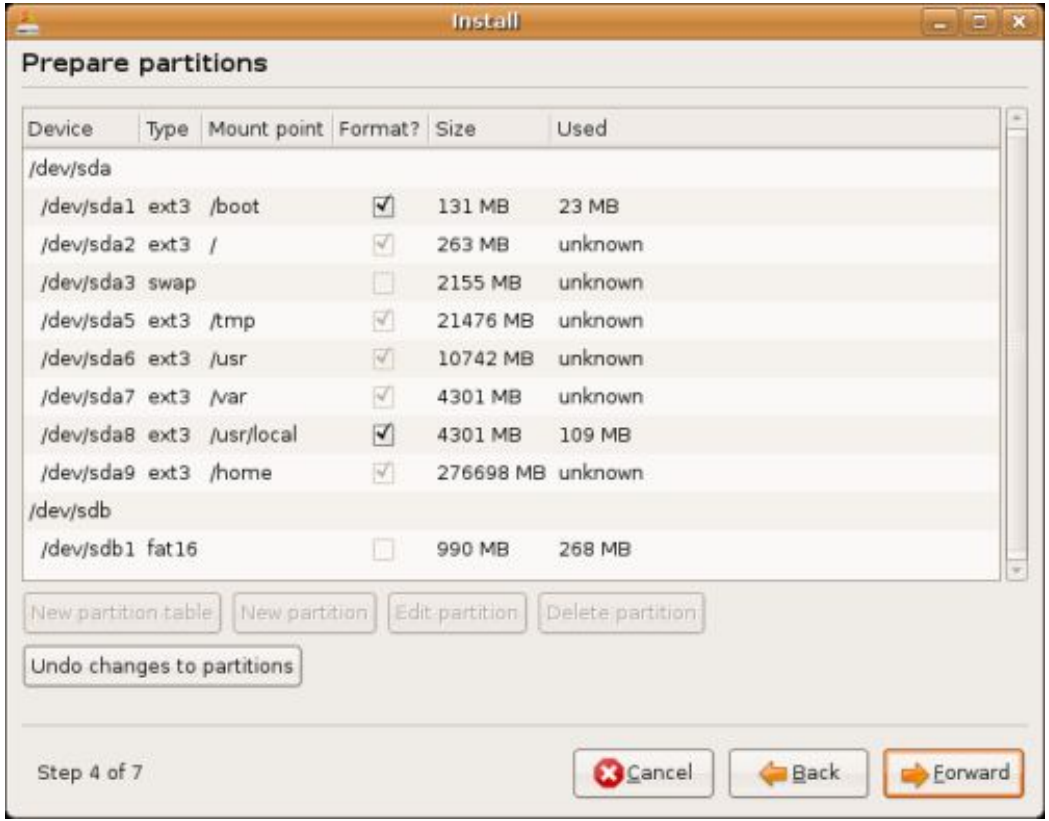
ZFS - overview

OpenZFS

- Storage pool, integrated volume manager.

```
$ df -h
Filesystem                Size    Used   Avail Capacity  Mounted on
zroot                     155G    526M    155G     0%    /
devfs                     1.0K    1.0K      0B   100%    /dev
zroot/tmp                 155G    261M    155G     0%    /tmp
zroot/usr                 172G     17G    155G    10%    /usr
zroot/usr/home            186G     31G    155G    17%    /usr/home
zroot/usr/ports           156G    1.2G    155G     1%    /usr/ports
zroot/var                 159G    4.0G    155G     3%    /var
zroot/var/crash           155G     96K    155G     0%    /var/crash
zroot/var/db              155G    216M    155G     0%    /var/db
zroot/var/log             155G    512K    155G     0%    /var/log
zroot/var/mail            155G    188K    155G     0%    /var/mail
zroot/var/tmp             155G    1.2M    155G     0%    /var/tmp
```

ZFS


Install
Prepare partitions
Device Type Mount point Format? Size Used
/dev/sda
/dev/sda1 ext3 /boot 131 MB 23 MB
/dev/sda2 ext3 / 263 MB unknown
/dev/sda3 swap 2155 MB unknown
/dev/sda5 ext3 /tmp 21476 MB unknown
/dev/sda6 ext3 /usr 10742 MB unknown
/dev/sda7 ext3 /var 4301 MB unknown
/dev/sda8 ext3 /usr/local 4301 MB 109 MB
/dev/sda9 ext3 /home 276698 MB unknown
/dev/sdb
/dev/sdb1 fat16 990 MB 268 MB
New partition table
New partition
Edit partition
Delete partition
Undo changes to partitions
Step 4 of 7
Cancel
Back
Forward

OpenZFS

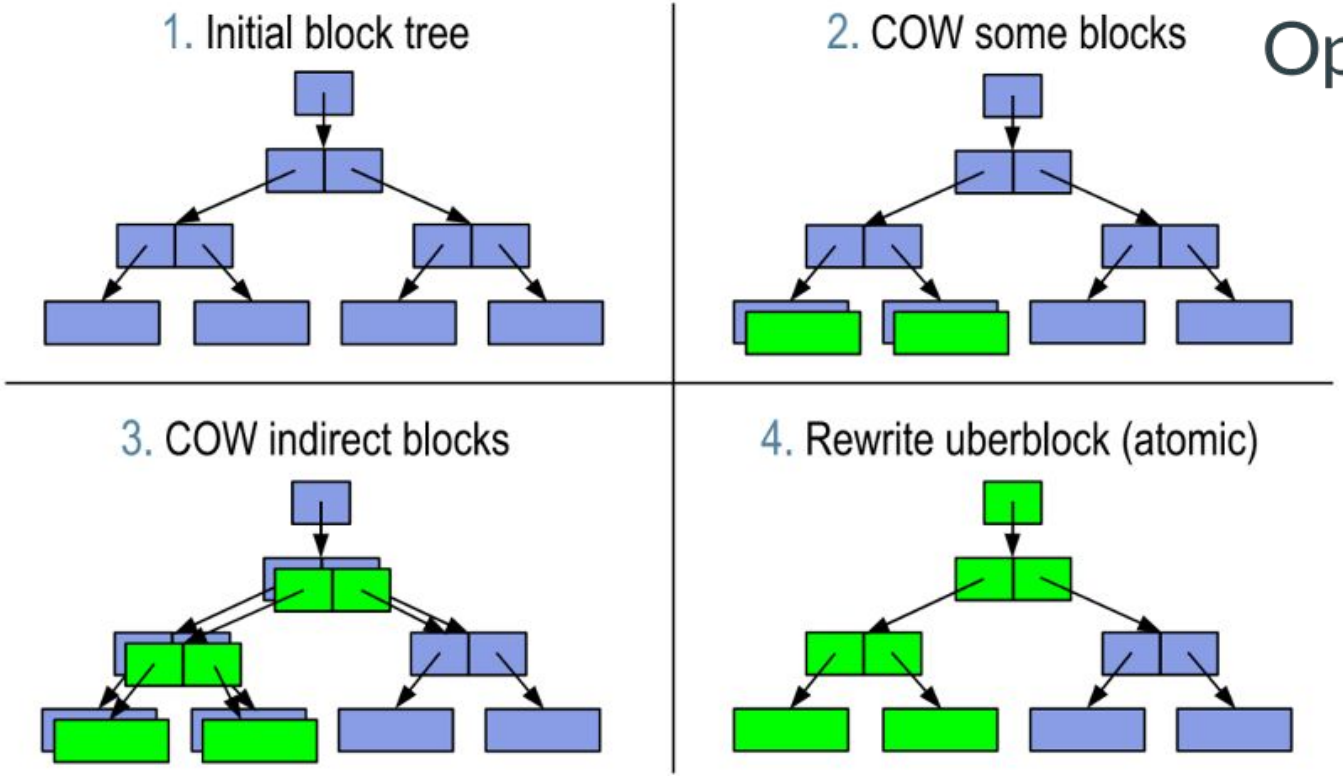
ZFS - overview

- End-to-end checksumming
  - SHA256, fletcher2, fletcher4
- Copy-on-write
- Transactional operations
  - Always consistent on disk
  - No fsck, no journaling

What is journaling?

- Stores metadata
- Recovery mechanism
- Used in:
  - Ext3
  - UFS with SU+J

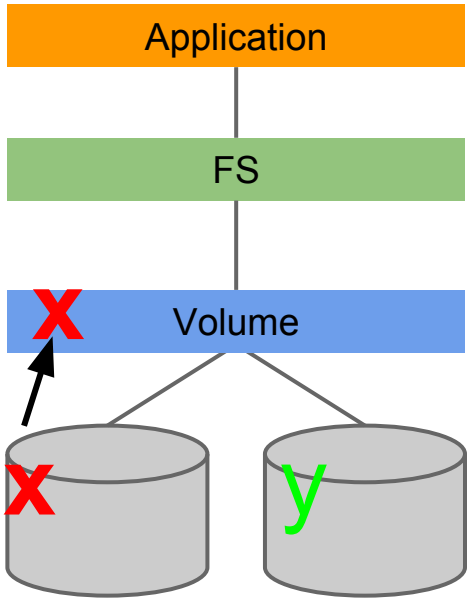
ZFS - CoW



http://wiki.illumos.org/download/attachments/1146951/zfs_last.pdf

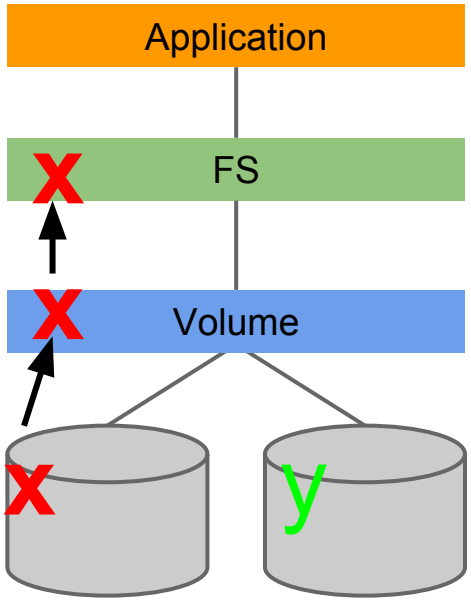
ZFS - overview

Traditional FS



ZFS - overview

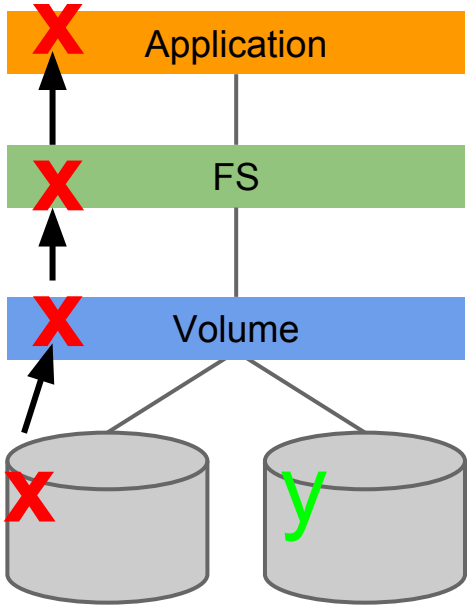
Traditional FS



OpenZFS

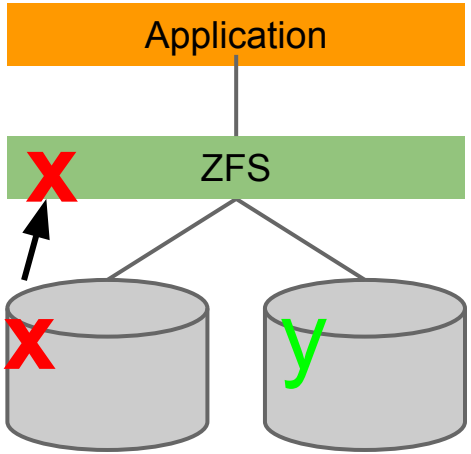
ZFS - overview

Traditional FS



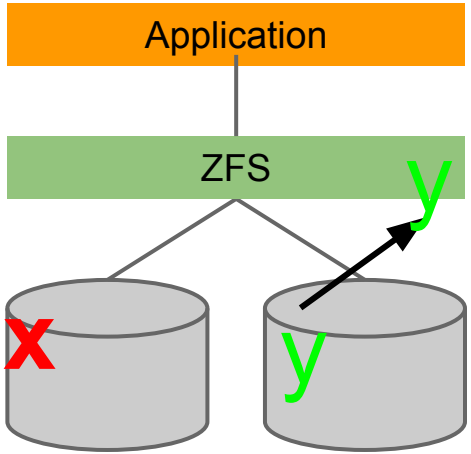
ZFS - overview

ZFS



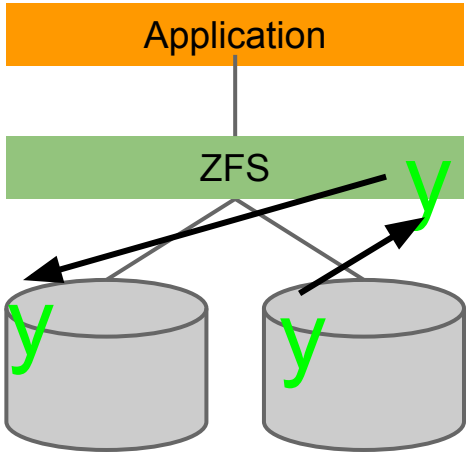
ZFS - overview

ZFS self-healing



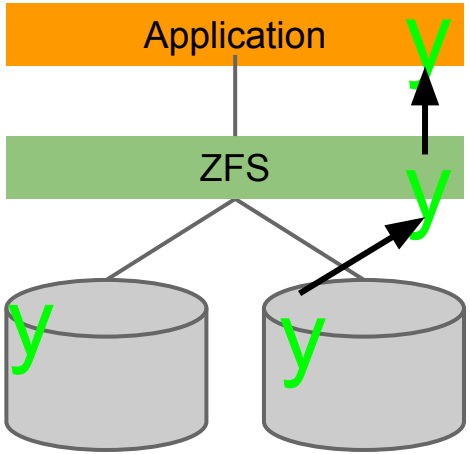
ZFS - overview

ZFS self-healing



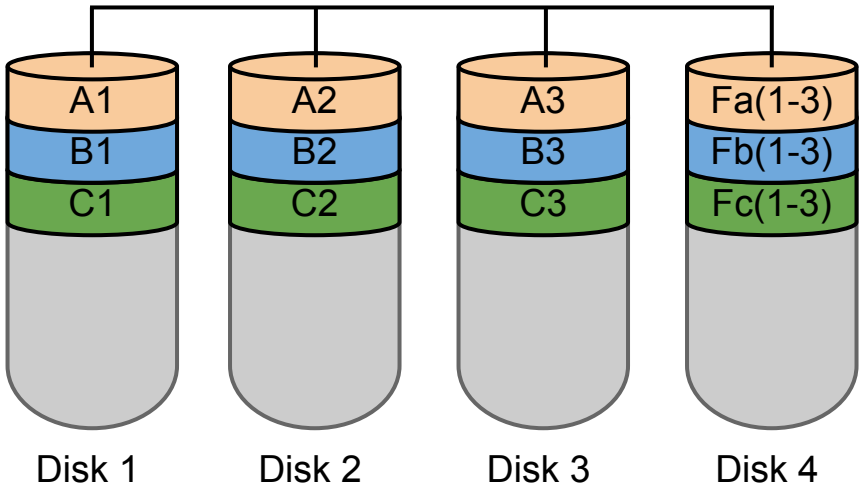
ZFS - overview

ZFS self-healing



ZFS redundancy

- Stripe
- Mirror (RAID-1)
- Raid-Z{1,2}



ZFS Raidz-Z{1,2,3}

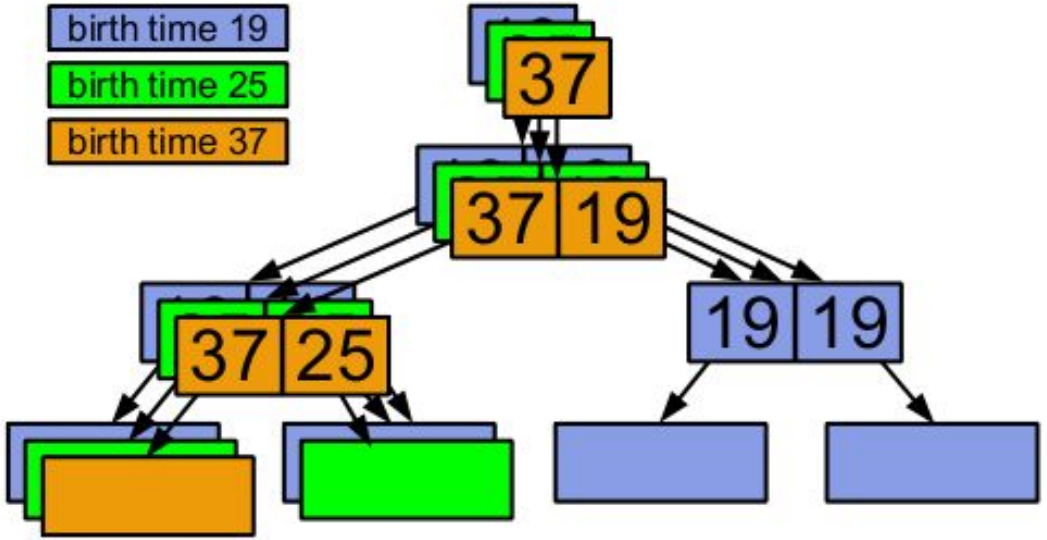
Raid	Minimal hard disk count	Recommended amount of hard disks	Parity hard disk	Data hard disk
RAIDZ1	2	3	1	n - 1
RAIDZ2	3	4	2	n - 2
RAIDZ3	4	6	3	n - 3

ZFS - other features

- Endian-independent
- Compression
  - LZ4
  - LZJB
  - GZIP
  - ZLE

ZFS - other features

- Snapshots
  - Very very cheap - O(1)
  - Birth time



http://wiki.illumos.org/download/attachments/1146951/zfs_last.pdf

ZFS - other features

- Snapshots

```
zfs snapshot tank/home/oshogbo@201606

zfs rollback tank/home/oshogbo@201512

cd ~/oshogbo/.zfs/snapshot/201512/
```

ZFS - other features

- Incremental snapshots
- Clones
- ZVOL
- Deduplication
  - verify
  - sha256

ZFS - other features

- Quotas
- Reservations
- NFS
- Resilvering

ZFS - when to use it?

- Backups

```
zfs send -i 201605 tank@201606 | ssh mbackup zfs recv -F obackup
```

- Bhyve/jails

```
# zfs get -o property,value all zroot/vm0001 bhyve:cpus 4
bhyve:memory 2048
bhyve:console /dev/nmdm0001A
bhyve:networks [{type:virtio-net,name:tap7]
bhyve:disks [ { type:ahci-hd, path:/dev/zvol/zroot/disk0001 },
              { type:ahci-hd, path:/dev/zvol/zroot/disk0002 }]
```

Bhyve example: http://www.allanjude.com/talks/vBSDCon2015_-_Interesting_ZFS.pdf

ZFS - when to use it?

- Cluster multi-master

```
NAME                        USED  AVAIL  REFER  MOUNTPOINT
data/data/10000013          40M   6.48G    33K  /data/10000013
data/data/10000013/data     29M   6.48G  29.3M  /data/10000013/data
data/data/10000013/pdfs     11M   6.48G  11.6M  /data/10000013/pdfs
data/data/local             4.74G  6.48G  4.57G /data/local
data/data/local/data        178M  6.48G   178M  /data/local/data
data/data/local/pdfs        436K  6.48G   402K  /data/local/pdfs
```

ZFS - ongoing work

- Resumable send/recv (FreeBSD 11)
- FS encryption
- Sending compressed blocks

ZFS downsides

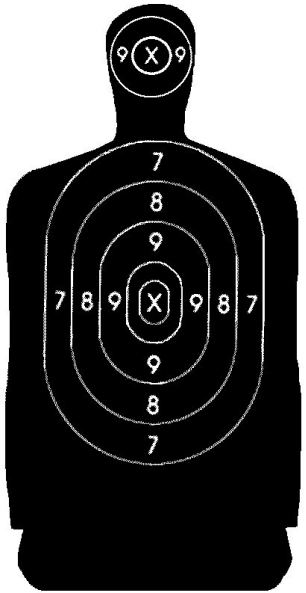
- Requires a lot of RAM.
- An insane amount of RAM (1GB RAM per 1TB of storage).
- Not so useful for embedded systems?
- Defragmentation
- Really slow on small capacity

FreeBSD

OS X

FreeNAS

Thank you for your attention!



<m.zaborski@wheelsystems.com>
<oshogbo@FreeBSD.org>