The last word in file systems
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
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
- Capsicum
- MAC Framework
- netmap
- Linuxulator
Are our hard disks invincible?
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?
Thats joke, right?

OpenSSH CVE-2002-0083 - privilege escalation to root

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

Source: Mike Perry, Seth Schoen
Thats joke, right?

Assembly

<table>
<thead>
<tr>
<th>cmpl $0x0,0x8(%ebp)</th>
<th>cmpl $0x0,0x8(%ebp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>js 16</td>
<td>js 16</td>
</tr>
<tr>
<td>mov 0x4,%eax</td>
<td>mov 0x4,%eax</td>
</tr>
<tr>
<td>cmp %eax,0x8(%ebp)</td>
<td>cmp %eax,0x8(%ebp)</td>
</tr>
<tr>
<td>jl 30</td>
<td>jle 30</td>
</tr>
<tr>
<td>mov 0x8(%ebp),%eax</td>
<td>mov 0x8(%ebp),%eax</td>
</tr>
<tr>
<td>mov %eax,0x4(%esp)</td>
<td>mov %eax,0x4(%esp)</td>
</tr>
<tr>
<td>movl $0x4c,(%esp)</td>
<td>movl $0x4c,(%esp)</td>
</tr>
<tr>
<td>call 25</td>
<td>call 25</td>
</tr>
</tbody>
</table>

Source: Mike Perry, Seth Schoen
Thats joke, right?
Hex

39 45 08 7c 1a 8b 45  
39 45 08 7e 1a 8b 45

Source: Mike Perry, Seth Schoen
Thats joke, right?

Binary

39 45 08 7c 1a 8b 45
39 45 08 7e 1a 8b 45

01111100
01111110

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?

<table>
<thead>
<tr>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 kilobyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1 megabyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1 gigabyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1 terabyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1 petabyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1 exabyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1 zettabyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
</tbody>
</table>
## Why ZFS is ZFS?

<table>
<thead>
<tr>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 kilobyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1 megabyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1 gigabyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1 terabyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1 petabyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1 exabyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1 zettabyte</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
</tbody>
</table>

160Gb ~ 275$
ZFS - overview

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

- Storage pool, integrated volume manager.

```bash
$ 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

### Prepare partitions

<table>
<thead>
<tr>
<th>Device</th>
<th>Type</th>
<th>Mount point</th>
<th>Format?</th>
<th>Size</th>
<th>Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/sda</td>
<td>ext3</td>
<td>/boot</td>
<td>☑️</td>
<td>131 MB</td>
<td>23 MB</td>
</tr>
<tr>
<td>/dev/sda2</td>
<td>ext3</td>
<td>/</td>
<td>☑️</td>
<td>263 MB</td>
<td>unknown</td>
</tr>
<tr>
<td>/dev/sda3</td>
<td>swap</td>
<td></td>
<td></td>
<td>2155 MB</td>
<td>unknown</td>
</tr>
<tr>
<td>/dev/sda5</td>
<td>ext3</td>
<td>/tmp</td>
<td>☑️</td>
<td>21476 MB</td>
<td>unknown</td>
</tr>
<tr>
<td>/dev/sda6</td>
<td>ext3</td>
<td>/usr</td>
<td>☑️</td>
<td>10742 MB</td>
<td>unknown</td>
</tr>
<tr>
<td>/dev/sda7</td>
<td>ext3</td>
<td>/var</td>
<td>☑️</td>
<td>4301 MB</td>
<td>unknown</td>
</tr>
<tr>
<td>/dev/sda8</td>
<td>ext3</td>
<td>/usr/local</td>
<td>☑️</td>
<td>4301 MB</td>
<td>109 MB</td>
</tr>
<tr>
<td>/dev/sda9</td>
<td>ext3</td>
<td>/home</td>
<td></td>
<td>276698 MB</td>
<td>unknown</td>
</tr>
<tr>
<td>/dev/sdb</td>
<td>fat16</td>
<td></td>
<td></td>
<td>990 MB</td>
<td>268 MB</td>
</tr>
</tbody>
</table>

[New partition table] [New partition] [Edit partition] [Delete partition]

[Undo changes to partitions]

Step 4 of 7

[Cancel] [Back] [Forward]
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

1. Initial block tree

2. COW some blocks

3. COW indirect blocks

4. Rewrite uberblock (atomic)
ZFS - overview

Traditional FS

Diagram:
- Application
- FS
- Volume
- X
- Y
ZFS - overview

Traditional FS
ZFS - overview

Traditional FS

- Application
- FS
- Volume
- x
- y
ZFS - overview

ZFS
ZFS - overview

ZFS self-healing

Diagram:
- Application
- ZFS
- Disks labeled X and Y
- Red X indicates failure or issue
- Green Y indicates healthy status
ZFS - overview

ZFS self-healing

![ZFS - overview diagram](image-url)
ZFS - overview

ZFS self-healing
ZFS redundancy

- Stripe
- Mirror (RAID-1)
- Raid-Z{1,2}
ZFS Raidz-Z\{1,2,3\}

<table>
<thead>
<tr>
<th>Raid</th>
<th>Minimal hard disk count</th>
<th>Recommended amount of hard disks</th>
<th>Parity hard disk</th>
<th>Data hard disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAIDZ1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>n - 1</td>
</tr>
<tr>
<td>RAIDZ2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>n - 2</td>
</tr>
<tr>
<td>RAIDZ3</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>n - 3</td>
</tr>
</tbody>
</table>
ZFS - other features

- Endian-independent
- Compression
  - LZ4
  - LZJB
  - GZIP
  - ZLE
ZFS - other features

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

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 }]
  ```

ZFS - when to use it?

- Cluster multi-master

<table>
<thead>
<tr>
<th>NAME</th>
<th>USED</th>
<th>AVAIL</th>
<th>REFER</th>
<th>MOUNTPOINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>data/data/10000013</td>
<td>40M</td>
<td>6.48G</td>
<td>33K</td>
<td>/data/10000013</td>
</tr>
<tr>
<td>data/data/10000013/data</td>
<td>29M</td>
<td>6.48G</td>
<td>29.3M</td>
<td>/data/10000013/data</td>
</tr>
<tr>
<td>data/data/10000013/pdfs</td>
<td>11M</td>
<td>6.48G</td>
<td>11.6M</td>
<td>/data/10000013/pdfs</td>
</tr>
<tr>
<td>data/data/local</td>
<td>4.74G</td>
<td>6.48G</td>
<td>4.57G</td>
<td>/data/local</td>
</tr>
<tr>
<td>data/data/local/data</td>
<td>178M</td>
<td>6.48G</td>
<td>178M</td>
<td>/data/local/data</td>
</tr>
<tr>
<td>data/data/local/pdfs</td>
<td>436K</td>
<td>6.48G</td>
<td>402K</td>
<td>/data/local/pdfs</td>
</tr>
</tbody>
</table>
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
Thank you for your attention!

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