By the Numbers: ZFS Performance Results from Six Operating Systems and Their Derivatives

@MichaelDexter
editor@callfortesting.org

vBSDcon 2019
STOP LOSING DATA
I ♥ ZFS

But it wasn’t love at first sight
We met at v15...

RaidZ planning was complex

Lose your SLOG and lose your pool

“You want 16GB more RAM than you currently have”
But we grew...

Iz4 compression simplified RaidZ planning

By v28 a SLOG would fail gracefully

Commodity hardware caught up with ZFS
Still Logical Conclusions

Rely on the CPU for data checksumming, compression, encryption...

Stripe vdevs to increase speed with capacity

CoW for snapshots, rollback, and replication
Continuous Data and Backup Validation

Validated on Read
Validated at Rest
Atomically-Identical Backups
You have a job to do
Fewer and fewer single-platform organizations

Ecosystems, not monocultures

Diversity by choice or even mandate
Putting the “v” in vBSDcon...
OpenZFS Platforms

Illumos Distributions
FreeBSD and Derivatives
GNU/Linux Distributions
macOS
Windows/Server/Hyper-V
NetBSD
ZFS vs. OpenZFS
Oracle Solaris
OpenZFS is becoming...
The TCP/IP | POSIX | OpenSSH
of File Systems
$DAYJOB
OpenZFS Crisis Line
FreeNAS/FreeBSD/Others

Zevo ZFS/OpenZFS on Mac since 2013
First ZFS on Windows on Hardware
We cannot have second-class OpenZFS Platforms
The Lab

HP Z220 X 10
Xeon E3-1225v2
8GB RAM

Overdrive 1000
AMD A1120 (A57 64-bit)
8GB RAM
The Lab 1.0

500GB 7200RPM Drives
Cygwin rather than PowerShell
Tribblix rather than OmniOSCE
Methodology 1.0

Equidistant OpenSSH touch, dd, mkdir, tree, realpath, zfs create dataset, native/Cygwin /bin/sh

How’s that portable BSD userland coming?
The Lab 2.0

Crucial BX500 120GB 3D NAND

Come for the silence, stay for the reporting

247 Host_Program_Page_Count 0x0032 ... 6846185
Methodology 2.0

Equidistant OpenSSH
FIO, the Flexible I/O Tester
(Where supported)
SmartMonTools/smart(8)
/bin/sh and PowerShell
The Lab 2.0

macOS Mojave
Windows 10
NetBSD 9.99
ZoL on FreeBSD 12
Hyper-V Core 2019
FreeBSD 12.0 ARM64

OmniOSCE
CentOS 7
Debian 10
Oracle Solaris
FreeNAS 11.2
The Lab 3.0 Hypervisors

HP Z420 Systems
FreeBSD/bhyve with OpenZFS
Hyper-V with OpenZFS
ESXi without OpenZFS
XCP-ng with OpenZFS
Proxmox with *root on ZFS!*
The Lab 3.0

SmartOS
FreeBSD 11-stable
FreeBSD 12-stable
FreeBSD Current
FreeBSD/Xen 11-stable...

Ubuntu
NetBSD/Xen
GNU/Linux ARM64
Windows Server
Benchmarks vs. Stress Testing

Mature platforms warrant benchmarking

New platforms warrant stress testing/fuzz testing
Unchartered Territory

“There is no way touch(1) could fail.”
“That RAM upgrade couldn’t possibly brick this motherboard.”
“Ported, portable software is portable.”

Be Consistent • Assume Nothing
Unchartered Territory

Always the tip of the iceberg
zpool create -f \ 
-0 casesensitivity=insensitive \ 
-0 mountpoint=/tank \ 
-0 compression=off \ 
-0 atime=off \ 
-0 sync=standard|always
FIO Strategy

fio --thread --randrepeat=1 \ 
--direct=1 --gtod_reduce=1 \ 
--name=test --bs=128k \ 
--filename=/tank/fio.tmp \ 
--iodepth=64 --size=8G \ 
--readwrite=randrw (--sync=1)
FIO Strategy

thread pthread_create, not fork
randrepeat predictable across runs
direct=1 non-buffered I/O
gtod_reduce=1 reduce gettimeofday() calls
name, bs=128k, filename, size
iodepth=64 number of I/O units in flight
readwrite=randrw (--sync=1)
<table>
<thead>
<tr>
<th>System</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyper-V ZFS</td>
<td>24.5MiB/s</td>
<td>24.6MiB/s</td>
</tr>
<tr>
<td>Hyper-V NTFS</td>
<td>127MiB/s</td>
<td>127MiB/s</td>
</tr>
<tr>
<td>CentOS ZFS</td>
<td>33.1MiB/s</td>
<td>33.2MiB/s</td>
</tr>
<tr>
<td>FreeNAS ZFS</td>
<td>33.1MiB/s</td>
<td>33.2MiB/s</td>
</tr>
<tr>
<td>CentOS Native</td>
<td>66.6MiB/s</td>
<td>66.7MiB/s</td>
</tr>
<tr>
<td>macOS ZFS</td>
<td>37.2MiB/s</td>
<td>37.3MiB/s</td>
</tr>
<tr>
<td>macOS HFS</td>
<td>83.6MiB/s</td>
<td>83.7MiB/s</td>
</tr>
<tr>
<td>System</td>
<td>Read</td>
<td>Write</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>Windows 10 ZFS</td>
<td>44.9MiB/s</td>
<td>44.9MiB/s</td>
</tr>
<tr>
<td>Windows 10 NTFS</td>
<td>128MiB/s</td>
<td>128MiB/s</td>
</tr>
<tr>
<td>NetBSD ZFS</td>
<td>46.9MiB/s</td>
<td>47.9MiB/s</td>
</tr>
<tr>
<td>NetBSD UFS</td>
<td>63.3MiB/s</td>
<td>63.4MiB/s</td>
</tr>
<tr>
<td>FreeBSD ARM64</td>
<td>69.2MiB/s</td>
<td>69.2MiB/s</td>
</tr>
<tr>
<td>Debian 10 Native</td>
<td>71.6MiB/s</td>
<td>71.7MiB/s</td>
</tr>
<tr>
<td>ZoLoF ZFS</td>
<td>83.7MiB/s</td>
<td>83.9MiB/s</td>
</tr>
</tbody>
</table>

OmniOSCE ZFS, Solaris ZFS, Debian ZFS TBD
<table>
<thead>
<tr>
<th>Operating System</th>
<th>Filesystem</th>
<th>Read (MiB/s)</th>
<th>Write (MiB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetBSD</td>
<td>ZFS</td>
<td>69.3</td>
<td>69.6</td>
</tr>
<tr>
<td>macOS</td>
<td>ZFS</td>
<td>79.5</td>
<td>79.6</td>
</tr>
<tr>
<td>macOS</td>
<td>HFS</td>
<td>88.9</td>
<td>89.1</td>
</tr>
<tr>
<td>Windows 10</td>
<td>ZFS</td>
<td>87.1</td>
<td>87.3</td>
</tr>
<tr>
<td>Windows 10</td>
<td>NTFS</td>
<td>127</td>
<td>127</td>
</tr>
<tr>
<td>Hyper-V</td>
<td>ZFS</td>
<td>90.9</td>
<td>91.0</td>
</tr>
<tr>
<td>Hyper-V</td>
<td>NTFS</td>
<td>127</td>
<td>127</td>
</tr>
<tr>
<td>Operating System</td>
<td>Read Speed</td>
<td>Write Speed</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>FreeNAS ZFS</td>
<td>100MiB/s</td>
<td>100MiB/s</td>
<td></td>
</tr>
<tr>
<td>CentOS ZFS</td>
<td>106MiB/s</td>
<td>106MiB/s</td>
<td></td>
</tr>
<tr>
<td>CentOS Native</td>
<td>116MiB/s</td>
<td>116MiB/s</td>
<td></td>
</tr>
<tr>
<td>FreeBSD ARM64</td>
<td>113MiB/s</td>
<td>113MiB/s</td>
<td></td>
</tr>
<tr>
<td>ZoLoF ZFS</td>
<td>143MiB/s</td>
<td>143MiB/s</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NetBSD UFS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OmniOSCE ZFS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solaris ZFS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Debian ZFS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TBD
## 512M Results async/async

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeBSD ARM64</td>
<td>473MiB/s</td>
<td>500MiB/s</td>
</tr>
<tr>
<td>macOS ZFS</td>
<td>1016MiB/s</td>
<td>1073MiB/s</td>
</tr>
<tr>
<td>CentOS ZFS</td>
<td>1456MiB/s</td>
<td>1538MiB/s</td>
</tr>
<tr>
<td>NetBSD ZFS</td>
<td>1504MiB/s</td>
<td>1490MiB/s</td>
</tr>
<tr>
<td>FreeNAS ZFS</td>
<td>1976MiB/s</td>
<td>2087MiB/s</td>
</tr>
<tr>
<td>ZoLoF ZFS</td>
<td>1915MiB/s</td>
<td>2023MiB/s</td>
</tr>
</tbody>
</table>
Known Issues

ZFS on Windows is still Alpha
ZFS on Windows ‘zfs send’ needs help
macOS long-term performance
Are we there yet?

Universal root on ZFS
Solaris • Illumos • FreeBSD • Proxmox
Ubuntu is Promised
Are we there yet?

Add-on ZFS Platforms
Windows, CentOS, Debian, Ubuntu, XCP-ng, Zential, and more!
Are we there yet?

Boot environments across OS’s
Hypervisor 9pfs support
Are we there yet?

Baton Pass!

Actual Output
## Sending from FreeNAS to ZoLoF

### Listing snapshots on ZoLoF

<table>
<thead>
<tr>
<th>NAME</th>
<th>USED</th>
<th>AVAIL</th>
<th>REFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>tank/baton@travel</td>
<td>0B</td>
<td>-</td>
<td>1.00G</td>
</tr>
</tbody>
</table>

### Sending from FreeNAS to ZoLoF

Listing snapshots on ZoLoF

<table>
<thead>
<tr>
<th>NAME</th>
<th>USED</th>
<th>AVAIL</th>
<th>REFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>tank/baton@travel</td>
<td>0B</td>
<td>-</td>
<td>1.00G</td>
</tr>
</tbody>
</table>

### Sending from ZoLoF to FreeBSD ARM64

Listing snapshots on FreeBSD ARM64

<table>
<thead>
<tr>
<th>NAME</th>
<th>USED</th>
<th>AVAIL</th>
<th>REFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>tank/baton@travel</td>
<td>0</td>
<td>-</td>
<td>1.00G</td>
</tr>
</tbody>
</table>
Sending from FreeBSD ARM64 to OmniOSCE

Listing snapshots on OmniOSCE

<table>
<thead>
<tr>
<th>NAME</th>
<th>USED</th>
<th>AVAIL</th>
<th>REFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>rpool/ROOT/omnios...</td>
<td>82.0M</td>
<td>-</td>
<td>622M</td>
</tr>
<tr>
<td>tank/baton@travel</td>
<td>0</td>
<td>-</td>
<td>1.00G</td>
</tr>
</tbody>
</table>
Sending from FreeBSD ARM64 to OmniOSCE

Listing snapshots on OmniOSCE

<table>
<thead>
<tr>
<th>NAME</th>
<th>USED</th>
<th>AVAIL</th>
<th>REFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>rpool/ROOT/omnios...</td>
<td>82.0M</td>
<td>-</td>
<td>622M</td>
</tr>
<tr>
<td>tank/baton@travel</td>
<td>0</td>
<td>-</td>
<td>1.00G</td>
</tr>
</tbody>
</table>

Sending from OmniOSCE to CentOS

Listing snapshots on CentOS

<table>
<thead>
<tr>
<th>NAME</th>
<th>USED</th>
<th>AVAIL</th>
<th>REFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>tank/baton@travel</td>
<td>0B</td>
<td>-</td>
<td>1.00G</td>
</tr>
</tbody>
</table>
### Sending from CentOS to Debian

Listing snapshots on Debian

<table>
<thead>
<tr>
<th>NAME</th>
<th>USED</th>
<th>AVAIL</th>
<th>REFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>tank/baton@travel</td>
<td>0B</td>
<td>-</td>
<td>1.00G</td>
</tr>
</tbody>
</table>
Sending from CentOS to Debian

Listing snapshots on Debian

<table>
<thead>
<tr>
<th>NAME</th>
<th>USED</th>
<th>AVAIL</th>
<th>REFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>tank/baton@travel</td>
<td>0B</td>
<td>-</td>
<td>1.00G</td>
</tr>
</tbody>
</table>

Sending from Debian to NetBSD

Listing snapshots on NetBSD

ssh 10.13 /sbin/zfs list -t snapshot
**Sending from NetBSD to macOS**

**Listing snapshots on macOS**

<table>
<thead>
<tr>
<th>NAME</th>
<th>USED</th>
<th>AVAIL</th>
<th>REFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>tank/baton@travel</td>
<td>23K</td>
<td>-</td>
<td>1.00G</td>
</tr>
<tr>
<td>/tank/baton/.zfs/snapshot/travel</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

macos:baton root# shasum -a 512 baton.img
ce6210e6a71eb408517b994853af409...

macos:baton root# cat baton.img.sha512
SHA512 (/tank/baton/baton.img) =
ce6210e6a71eb408517b994853af409
Wish List

OpenZFS Lite/Embedded
Permissively-licensed
Single-disk/NVMe only, no RaidZ
Minimal caching
You have a job to do
Wish List cont.

About that Tweet...

Looking for a few 100K to port #OpenZFS to #VMware and #OpenBSD. Has to happen, even if neither will be official. #SeenThings
About that...
Hyper-V Core
+
OpenZFS
+
Windows Admin Center
+
Chrome...
**Overview**

**Computer name**: hyperlab

**Domain**: -

**Operating system**: Microsoft Hyper-V Server

**Version**: 10.0.17763

**Installed memory (RAM)**: 8 GB

**Disk space (Free / Total)**: 95.12 GB / 111.19 GB

**Processors**: Intel(R) Xeon(R) CPU E3-1225 V2 @ 3.20GHz

**Manufacturer**: Hewlett-Packard

**Model**: HP Z220 SFF Workstation

**Logical processors**: 4

**Windows Defender Real-time protection**: On

**Up time**: 0:6:51

**NIC(s)**: 1

**Logged in users**: 1

**Azure Backup status**: Not protected
Hyper-V Tab
IPMI Tab
pfSense Tab
FreeNAS Tab
ESXi Tab
Proxmox Tab
Zential Tab...
Increasingly on OpenZFS
See you there!

November 4th - 5th
San Francisco, CA

Jörgen Lundman and I will talk about OpenZFS Portability
Special Thanks

Jörgen Lundman for ZFS on macOS/Windows

Conor Beh and Jason Barbier for Windows assistance

Verisign, Dan, and Team for vBSDcon
Thank you!

Questions?

@MichaelDexter
editor@callfortesttesting.org