OpenZFS
The Best FileSystem for Every OS
Summary & Introductions

1. Allan Jude
   FreeBSD Core Team
   OpenZFS Developer

2. Klara Inc.
   FreeBSD Professional
   Services and Support
Covered in this presentation

What is ZFS? Why all the excitement?
What is OpenZFS? How do I Join?
What new features are in OpenZFS?
What does the future hold for OpenZFS?
What is ZFS?

- ZFS is a filesystem with a built in volume manager (RAID)
- Space from the pool of disks is thin-provisioned to multiple filesystems or block volumes (zvols)
- All data and metadata is checksummed
- Optional transparent compression (LZ4, GZIP, soon: ZSTD)
- Copy-on-Write with snapshots and clones
- Each filesystem is tunable with properties
What Is Pooled Storage?

Regular Filesystem (Single Logical Disk)

- Filesystem
  - Volume
  - Disk

ZFS (Many Filesystems share all space)

- ZFS
  - ZFS
  - ZFS
  - Volume
  - Disk
  - Disk
  - Disk
What Is Copy-on-Write?

This is your disk:

- File Version 1

This is your disk on ZFS:

- File Version 1
What Is Copy-on-Write?

This is your disk:

File Version 2

This is your disk on ZFS:

File Version 1  File Version 2
What Is Copy-on-Write?

This is your disk when the power fails:

Partial File Version 3  Partial File Version 2

This is your disk on ZFS:

File Version 3  File Version 2
Transaction Updates

1. Consistent filesystem state
2. Modify a file
3. Update Metadata
4. Write the uberblock (commit)
Why All The Excitement?

- Copy-on-Write means snapshots are consistent and instant
- Blocks used in snapshot(s) kept when overwritten/deleted
- Snapshots allow access to filesystem at point-in-time
- No performance impact on reads/writes
- Take no additional space until blocks change
- Makes your storage ransomware-resistant
- Clones allow you to “fork” a filesystem
The Evolution of ZFS

- ZFS was originally developed at Sun Microsystems starting in 2001, and open sourced under the CDDL license in 2005
- Oracle bought Sun in 2010, and close sourced further work
- illumos, a fork of the last open source version of Solaris became the new upstream for work on ZFS
- ZFS was ported to many platforms, including FreeBSD in 2007 and Linux in 2008. The OpenZFS project was founded to coordinate development across platforms.
OpenZFS

- The original plan for OpenZFS was a single common repository where the OS independent code would live and be tested
- Each OS would sync with this repo and add their own glue
- However, the effort required to maintain a repo that would not be directly used by any of the consumers was not viable
- The “repo of record” became a fork of illumos
- FreeBSD tracked very closely
- Linux spent a great deal of effort getting caught up
Platforms

- OpenZFS is now available on almost every platform
  - illumos (OmniOS, OpenIndiana, SmartOS, DiOS, Tribblix)
  - FreeBSD (FreeNAS, XigmaNAS, TrueOS, pfSense, etc)
  - Linux (ZFS-on-Linux, Ubuntu, Gentoo, OviOS)
  - Mac OS X (ZFS-on-OSX, GreenBytes/ZEVO, Akitio, MacZFS)
  - Windows (https://openzfsonwindows.org/)
  - NetBSD
  - OSv
Divergence

● Each different platform’s version of ZFS started to diverge
● OpenZFS replaced the old “pool version number” with “Feature Flags”, since features would land in different orders
● Bugs were fixed in one repo and not necessarily upstreamed or communicated to other platform’s could apply the same fix
● Each camp did their development within their own community, and other communities might not be aware of duplicate efforts, etc.
The new OpenZFS project organized a conference in November 2013 to have developers from the various platforms share their work and future ideas and find solutions. Included a platform panel (Linux, Mac OS, IllumOS, FreeBSD) and vendor lightning talks. Attended by over 30 developers, since grown to over 100. Now includes a hackathon to work on prototypes while experts are in the room for advice / design discussions.
Leadership Meeting

- At the OpenZFS Developer Summit 2018 a discussion between the various platform leaders lead to the formation of a monthly video conference to discuss ongoing issues.
- Meeting once a month instead of once a year provides more information exchange and faster response times.
- Goal is to keep the platforms better in-sync and compatible.
- Open to anyone. Live streamed and recorded to Youtube.
Outcomes

● The leadership meetings have been very successful
● OpenZFS is working to standardize the command line interface where it has diverged across platforms
● New features are discussed during the design phase and platform specific issues are resolved early, with better results
● More effort into effective naming of tunables (ashift is an internal implementation detail, the user tunable should be called sectorsize and be expressed in bytes)
Deprecation Policy

- Creation of a deprecation policy. After 18 years it is time to remove a feature from ZFS: Deduplicated send (replication).
- Feature is not related to pool dedup, rarely used, complicated.
- This will require building a utility to convert old replication streams so they can be received by future versions of ZFS.
- Also slated for removal: dedupditto. Designed to write a 2nd copy of a block if it is deduped more than 100 times. Turns out it has never worked properly, not checked/fixed on scrub.

klarasystems.com
Cross Platform Compatibility

- Improved user interface for pool creation. Specify `compatible=openzfs-2019` or `compatible=freebsd-12`
- Enable only those feature flags compatible selected platforms
- The OpenZFS-YYYY macro will refer to what is available across all platforms as of January of that year
- Still identifying what best options are for other values
- Need to suppress the commended upgrade in `zpool status`
Compression Conundrum

- Another feature is slated for removal: the ability to disable the Compressed ARC feature. Interferes with updating compression algorithms and increases code complexity.

- I am working on adding ZStandard compression of ZFS:
  - ZStandard is under very active development, we do not want to be frozen to an older version for compatibility.
  - Need to support upgrading compression algorithms.
  - No guarantee same data will compress to same hash.
Features: All Platforms

- sequential scrub/resilver
- zpool scrub pause/resume
- device removal
- zpool checkpoint
- zpool initialize
- spacemap encoding v2
- Channel programs
- large dnode
Features: Some Platforms

- Encryption (incl. raw send/recv)
- multi-import protection (MMP)
- special devices for metadata (allocation classes)
- parallel ZFS mount
- zpool sync
- TRIM (new way)
- resilver restart
- xattr=sa
Features: Coming Soon

- fast clone deletion
- spacemap log
- remove dedupditto
- redacted send/recv
- ZSTD compression
- per-vdev properties
- Enable compression by default
Features: Future

- RAID-Z Expansion
- DRAID
- Persistent L2ARC
- Adaptive Compression (compress more when not busy)
- Smart Compression (file based heuristic)
- Platform specific ShareNFS property handling
Features: Wish List

- Improved dedup (dedup log)
- Offline Dedup
- File Cloning
- Per-dataset throttling/QoS (IOPS and BPS)
- SMR Support (Shingled Disks)
- Clustered Features
- Continuous Replication
The Linux 5.x Scare

- With the release of Linux kernel 5.0 it was announced that some kernel functions that OpenZFS was using were being removed in favour of newer GPL-only symbols.
- This broke compilation of OpenZFS and scared a lot of people.
- The functions that were removed were to do with SSE/SIMD.
- Were used to Vectorize (speed up) Checksumming.
- ZFS-on-Linux simply reverted to standard checksumming.
The Linux 5.x Scare

● Greg Kroah-Hartman followed up on the mailing list with:
  ○ "Sorry, no, we do not keep symbols exported for no in-kernel users."
  ○ "my tolerance for ZFS is pretty non-existant."

● Longtime Linux kernel developer Christoph Hellwig also suggested users switch to FreeBSD instead if they care about ZFS.
Get Involved

- The OpenZFS community is very active and very welcoming
- Watch some of the past “OpenZFS Leadership Meeting” conference calls on youtube to see for yourself
- The “repo of record” is transitioning to the ZFS-on-Linux repo as it has the most active development and the most code that still needs to be pulled into other platforms
- Github Issues and Pull requests
- Mailing Lists (Topic Box) for discussions
More Resources

● Want to know more about ZFS?
  ○ “FreeBSD Mastery: ZFS” & “FreeBSD Mastery: Advanced ZFS”
  ○ Not just for FreeBSD, DRM-Free ebooks ZFSBook.com

● BSDNow.tv - Weekly video podcast on BSD & ZFS

● @allanjude on twitter