ZFS: Advanced Integration

Allan Jude -- allanjude@freebsd.org @allanjude

Introduction: Allan Jude

- 16 Years as FreeBSD Server Admin
- FreeBSD src/doc committer (ZFS, installer, boot loader, GELI, bhyve, libucl, libxo)
- FreeBSD Core Team (July 2016 - 2018)
- Co-Author of "*FreeBSD Mastery: ZFS*" and "*FreeBSD Mastery: Advanced ZFS*" with Michael W. Lucas
- Architect of the ScaleEngine CDN (HTTP and Video)
- Host of weekly BSDNow.tv Podcast
- Personally Responsible for 1000 TB of ZFS Storage

ZFS: What Is It?

- ZFS is a filesystem with a built in volume manager (combine multiple disks into a pool)
- Space from the pool is thin-provisioned to multiple filesystems or block volumes (zvols)
- All data and metadata is checksummed
- Optional transparent compression
- Copy-on-Write with snapshots and clones
- Each filesystem is tunable with properties

Snapshots and Clones

- Copy-on-Write means snapshots are instant
- Blocks referenced by a snapshot kept when they are removed from the live filesystem
- Snapshots allows you to access the file-system as it was when snapshot was taken
- No performance impact on reads/writes
- Take no additional space until blocks change
- Clones allow you to “fork” a filesystem

Boot Environments

- If the root filesystem is on ZFS, you can snapshot before an upgrade, then clone it
- You now have 2 filesystems, one before the upgrade, and one after. Only takes the space of blocks that changed
- FreeBSD boot loader allows you to select which one to use from a menu
- Upgrade without fear, can always rollback

Boot Environment Tooling

- Existing tool: sysadmin/beadm
- GSoC 2017: be(8) and libbe(3)
- New tool supports better management of filesystem properties for boot integration
- New tool will support “deep” boot environments. Child dataset management.
- Library allows better integration with other tools like pkg(8) and GUIs

What Boot Environments Looks Like

NAME	USED	REFER	WRITTEN
z	455M	1M	1M
z/ROOT	453M	1M	1M
z/ROOT/default	452M	451M	307K
z/ROOT/default@11.1p0	1.75M	410M	410M
z/ROOT/default@11.1p2	211K	450M	41.9M
z/ROOT/11.1p0	12.8K	410M	12.8K
z/ROOT/11.1p2	12.8K	450M	12.8K

The Rest of the Pool

z	The root of the pool
z/tmp	/tmp
z/usr	not mounted, parent
z/usr/home	preserved across BEs
z/var	not mounted, parent
z/var/audit	audit files not versioned
z/var/crash	hopefully empty ;)
z/var/log	Keep logs across BEs
z/var/mail	Don’t lose mail, atime
z/var/tmp	/var/tmp

Example: Laptop

- This laptop uses Boot Environments
- If an OS or pkg upgrade goes sideways the day before my presentation and I don’t notice until I can’t output to the projector, I just reboot to last weeks working dataset
- My home directory (with the last minute update to the slides) is preserved even when I rollback the operating system

Example: Deep Boot Environments

- Some users and developers have more complex needs and preferences
- src and obj should be datasets with extra properties for increased performance
- The src and obj mounted with each dataset should match the running OS in the BE
- Ensure the correct child datasets are mounted based to match selected BE

Deep BEs

```
NAME                                USED
zroot/ROOT/newest                   1.34G
zroot/ROOT/newest/usr/obj           88K
zroot/ROOT/newest/usr/src           1.34G
zroot/ROOT/cloned                   220K
zroot/ROOT/cloned/usr/obj           0
zroot/ROOT/cloned/usr/src           0
```

BEs as Golden Images

- At ScaleEngine we use boot environments on all of our servers
- We started with just stock FreeBSD with security patches applied
- zfs send | xz > 11_1p2.xz
- fetch -o - https://imgsvr/11_1p2.xz | zfs recv
- Temporary mount to /mnt
- Copy select configuration files over

Persist Config Across Firmwares

- We have since enhanced this process
- New /cfg dataset holds persistent configs
- Images have those files symlinked from /etc
- zfs recv updated image
- zfsbootcfg (enhanced ZFS nextboot)
- If the new image doesn’t work, reboot to old
- If new image passes then zfs set bootfs
- Upgrades (minor or major) take seconds

Replace NanoBSD

- Replace nanobsd in your appliance with ZFS
- FreeNAS and pfSense have already done so
- Keep as many old images as you have space
- Still get firmware style updates
- Added reliability of ZFS
- Enhanced nextboot: 3 consecutive boot failures or uptimes less than 5 minutes automatically boots rescue system to allow intervention of headless appliances or AWS instances

Encryption Option #1: GELI

- AES-XTS or AES-CBC
- Full block device is encrypted (key per disk)
- Support for booting from encrypted dataset with only unencrypted gptzfsboot since 2016
- EFI support for booting encrypted pools expected before end of 2017
- Requires console access to enter passphrase
- No keyfile support in boot loader

Encryption Option #2: ZFS Native

- AES-GCM or AES-ICM
- Not all metadata is encrypted, and optionally not all datasets, but allows datasets to be unmounted and keys unloaded, so data is protected as it is actually “at rest”
- Scrub and Resilver without keys loaded
- Different keys for different datasets
- Expected in Spring 2018

GELI Enhancements

- BSDCan and BSDCam GELI working groups produced new metadata structures to enhance GELI to support many user keys and more advanced options. Expected 2018Q2
- Support for loading key material from USB devices or similar is planned for 2018Q3
- For laptops: support for second passphrase that boots alternate partition

Appliances: Channel Programs

- Until just a few months ago performance many ZFS administrative operations consecutively was not atomic and often slow
- New ZCP feature allows you to create short LUA scripts to perform bulk or iterative operations with the right locks held
- Instruction count and memory limited to prevent runaway processes
- Integrated last month! More scripts coming.

Appliances: Checkpoints (2017Q4)

- Upgrade process is always more involved than just updating the underlying operating system and tools
- Creating a checkpoint preserves ALL data
- Can undo operations that a snapshot cannot, like destroying or rename datasets
- If upgrade process fails midway, Roll back to checkpoint, as if it never happened
- Preserve checkpoint until upgrade is confirmed good. Don’t keep long term, no blocks are freed

What Would Make ZFS Better For You?

- I just came from the ZFS Developers Summit
- The cross-platform community is very active and interested in features that benefit users
- We would love to hear your ideas
- FreeBSD Foundation and Delphix are partnering to bring the most often requested feature: RAID-Z vdev expansion
- What do you need?

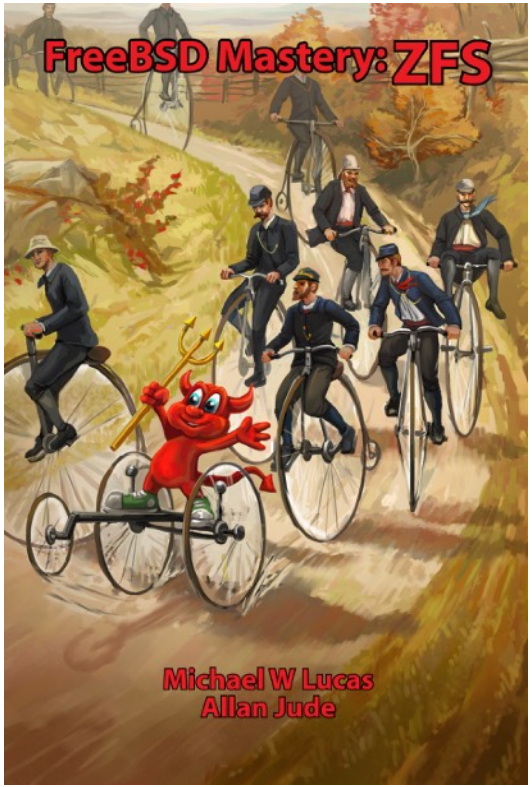
Near Future Features:

- ZSTD Compression
- Adaptive Compression
- Faster Resilver (sequential)
- Smarter Resilver (prefetch)
- ZIL performance enhancements
- MMP: Safe "zpool import" for Clusters
- Device Removal (Evacuation)

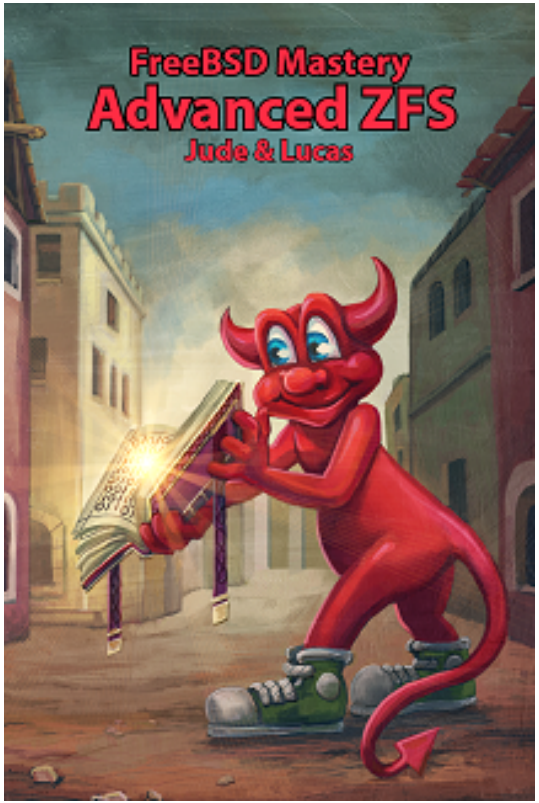
Further Future Features:

- ZIL performance enhancements
- Fast clone deletion
- Spacemap log (faster alloc/free)
- ashift policy. Replace 512b with 4Kn disks
- Distributed Parity (DRAID)
- VDEV Classes (metadata, block size)
- 1000x Dedup performance using dedup log

Get Your Copy at ZFSBook.com



Beginner and Advanced guides to ZFS for home and production. Ebook & Paperback from Amazon & others



BSDNow.tv

BSDNow.tv is a weekly video podcast featuring News, and Developer Interviews about the BSD family of Operating Systems. Hosted by Benedict Reuschling (VP FreeBSD Foundation) and Myself.

Always looking for people to Interview, email guests@bsdnow.tv to schedule yours.