ZFS: The Last Word in Filesystems?

Maurice Castro
maurice@castro.aus.net

Agenda

- What is ZFS?
- Why do we need it?
- Comparison?
- Competitors
- Extra features
- How is it used?

Agenda

- Platforms
- A Word on Performance

What is ZFS?

- ZFS (Zettabyte File System)
  - Sun Microsystems
    - Jeff Bonwick - Sept 2004
- Combines
  - Volume management
  - File system

Why do we need ZFS?

- Disk sizes are growing
- Data stored is growing
- Disk reliability is not growing fast enough
- ZFS performs end-to-end checksums on stored data

Why do we need ZFS?

- Where does storage fail?
  - Spindles
    - Surfaces / Cells
    - Controllers
- Errors / Noise can occur at any point

Why do we need ZFS?

- ZFS performs end-to-end checksums on stored data at the block level

Comparison

- Traditional Filesystems
  - LVM and Ext3
  - Geom and FFS
- FSCK only checks metadata not data

Competitors

- BTRFS
- Linux
- Ready for prime time?

Comparison

- Mirrors
  - Good - Bit level comparison
  - Bad - Failed sector returns working copy
- RAID
  - Parity bits - more efficient space usage
- Neither case controller failure not handled

Extra features

- Copy-on-write transactional model
- Snapshots and clones
- Variable block sizes
- Adaptive endianness
- Deduplication
- Encryption

Extra features

- Built in versioning
- User quotas in later versions

How is ZFS used?

- Filer / Storage server
  - Managed with 2 commands:
    - zpool & zfs
  - Pool types: mirror, raidz, raidz2, raidz3, hot spare
  - Copies of data on disk

Platforms

- Solaris / OpenSolaris
- FreeBSD
- NetBSD
- FUSE - performance issues
- Linux - 2 native ports (delayed by licensing)

A Word on Performance

- Good performance requires hardware
- ZFS is really a multilevel cache
  - Checksums needs CPU
  - Async throughput needs Mem & Ctrl BW
  - Sync throughput needs non-volatile cache
  - Deduplication requires CPU and Mem