ZFS: The Last Word in Filesystems?

Todd Underwood
Renesys
todd@renesys.com

Presentation Overview

- Context, Background and Prejudices
- Requirements
- ZFS overview
- Renesys solution case study
  - Hardware architecture
  - Performance
  - Cost
- Review of status OSS ports

renesys

Renesys / Babbledog & Data

- Renesys has two lines of business
  - Intelligence services from global routing (BGP) data
  - Realtime chat application in a browser sidebar
- O(GBs) raw data per day processed in near-realtime
- O(100sGB) cooked/indexed data per day
- O(10sTB) total data stored
- Computation spread among O(100) servers.

renesys

Requirements (for me)

- Reliable
- Shared access (NFS preferred)
- High (reasonable) performance
- Mangable: extensible, quotas, bandwidth guarantees, compression
- Cheap

Just for Context: Sun Sucks

- I hate Sun with a passion
- SunOS -> Solaris transition sucked
- Solaris antiquated and hard to use from a modern, free OS perspective
- Sun screwed up Java licensing for a full decade, preventing the language from achieving its full potential.
- Sun has no hardware, OS or software strategy: Many sequential quarters of losses (which may be good)
- Sun hates small customers. Refuses to ship product, refuses to support it, refuses to

Options Considered

- Linux NFS server
- Proprietary NFS Servers
  - Netapp
  - EMC
  - Agami
- Proprietary iSCSI servers
  - Lefthand
  - EqualLogic

Storage Architecture Review

- Block Layer
- Logical Volume Management Layer
- Filesystem Layer
- Network Filesystem Layer

History of the Hype

- ZFS announced Sept, 2004
- Integrated into Solaris Development Main trunk Oct, 2005
- Integrated into Opensolaris Nov, 2005
- Steady and growing attention throughout 2006 as people realized just how damned good it was
- Shipped in Solaris 10 update June, 2006
- Still not bootable in Solaris but grub support exists and work is underway

ZFS Rocks

- Most original work in storage management in over a decade. Maybe two. Seriously.
- First release appears to be astonishingly stable and full-featured. Completely production-ready. Development continues at an encouraging pace.
- Solves real problems with hardware RAID, software RAID
- Worth running Solaris to get. I deployed Solaris storage servers into a 100% Linux environment and don’t regret it (yet).

What's Better About ZFS?

- Merge block, logical volume management, filesystem, network filesytem layers
- **Not** a pointless, management-convenience-only, shiny new front end on the same crap merger
- Thorough, deep integration of the storage elements in order to fix problems (the RAID5 write hole, silent data loss), improve performance.
- Will make it hard to port to free operating systems (doesn't fit cleanly in the Linux VFS)

What's Better (2) ?

- Every read checksummed
- Dynamic pool allocation/increase
- IO performance guarantees/scheduling
- Trivial FS creation/management
- **Snapshots** and trivially replicated snapshots

•ZFS Architecture Overview

- Checksum per write, zero transparent data loss
- Copy on write for every write:
  - Most fs access is sequential
  - Checkpoint/snapshot trivial and zero-cost
- Fancy FS internals:
  - IO scheduling
  - Dynamic block size
  - Dynamic prefetch queues
  - Huge limits (128-bit data and 64-bit metadata)
  - Constant time directory operations

ZFS Architecture Basics

- No more partitions
- Pools have disks and a RAID strategy
- Filesystems are in pools
- Filesystems have other properties
  - Mountpoints
  - Exported or not
  - Compressed or not
  - Quotas
  - Bandwidth reservations

Renesys ZFS Architecture

- Dual core dual proc opteron, 16GB RAM, LSI PCIe SAS card
- Dell SAS – SATA 15 disk shelf
- 500GB SATA drives
- NFS exported to production servers
- (iSCSI considered for the future)

renesys

ZFS Basics

- Make a pool:
  - zpool create <poolname> <mirrortype> <devices....>
  - fs created automatically named as /<poolname>
- Create nested filesystems inside:
  - Zfs create /<poolname>/<fsname>
  - Share the global pool of space
  - Change montpoints to put them in the right place—automatically get umounted and remounted

ZFS Performance

- Disk speed
- Seriously, pretty much just disk speed (unless bounded by some bus or interface speed)
- But sometimes better (compression)

Iostat results

- Sunfire x4100 w/ PCI LSI SAS3442E-R to Dell Powervault MD w/ 15 500GB disks
- RAID10, 14 disks in 2 groups of 7 raidz
- No Compression:
  - Peak Read: 609589KB/s
  - Peak Write: 200661 KB/s
- With Compression:
  - Peak Read: 910876KB/s
  - Peak Write: 283547KB/s

ZFS on Linux?

- Userland via FUSE (slow, POC only)
- Kernel: serious incompatibilities between Sun CDDL and GPL
- ZFS support integrated into GPL grub, but insufficient for a full implementation
- Linus cranky about Sun and ZFS: patent issues and says sun are liars
- Interesting : Nexenta—Opensolaris kernel with Debian userland (http://www.gnusolaris.org/)

renesys

•OSS OS Ports (Continued)

- FBSD – in 7 current, some management issues
- Mac OSX in there now, but not default or boot

Renesys/Babbledog are Hiring! (sneaky plug)

- Software Engineers
- Web Designers
- Web Developers
- Work with smart, motivated people to build, extend and maintain software systems with large and complex network datasets, and interesting web applications and browser plugins.
- Required: Smart, gets things done
- Location: Manchester, NH, Hanover, NH or telecommute from any US location.
- jobs@renesys.com (pdf, txt, odt)

renesys

References

- http://www.sun.com/2004-0914/feature/
- http://www.opensolaris.org/os/community/z/
- http://blogs.sun.com/bonwick/
- http://blogs.sun.com/martin/entry/zfs_from_