Open ZFS @ BSDCan

May 2013
Matt Ahrens
mahrens@delphix.com
@mahrens1

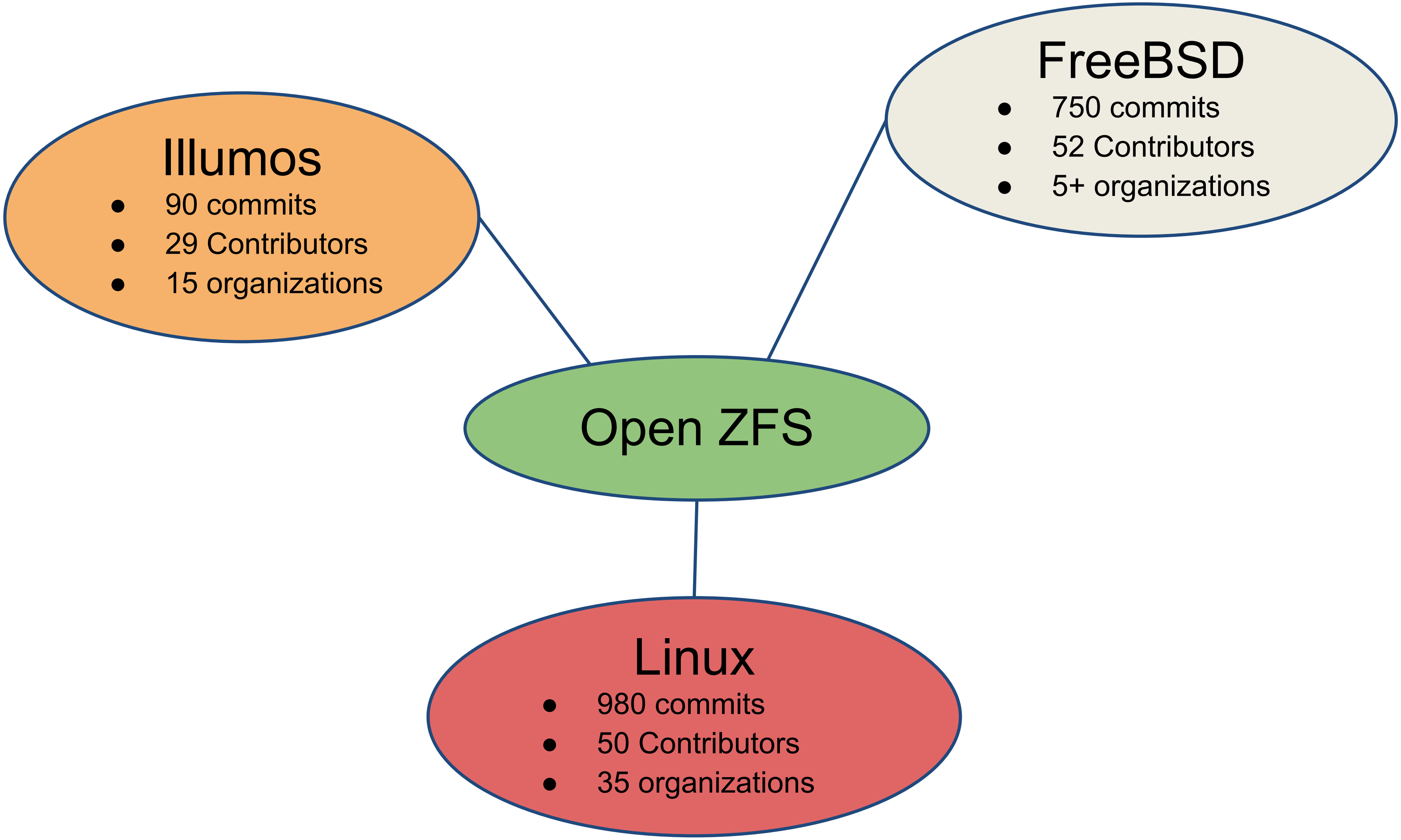
ZFS History

- 2001: development starts with 2 engineers
- 2005: ZFS source code released
- 2006: ZFS on FUSE for Linux started
- 2008: ZFS released in FreeBSD 7.0
- 2008: Sun’s 7000 series ZFS Storage Appliance ships
- 2010: Oracle stops contributing to Open ZFS
- 2012: ZFS on (native) Linux release candidate

ZFS Today

- Oracle is no longer contributing source
- illumos is the open-source continuation of Open Solaris
- Continuing development of Open ZFS on illumos, FreeBSD, and Linux
  - zfs send stream size estimation
  - zfs send progress reporting (Bill Pijewski)
  - single-copy ARC (George Wilson)
  - zfs destroy <snapspec> destroys many snaps at once
  - write throttle smoothes out write latency
  - background destroy of filesystems, >100x faster destroy of clones (Chris Siden)
  - performance for partial-block writes (Justin Gibbs & Will Andrews)
  - libzfs_core - simple, atomic API for programmatic zfs administration
  - SCSI UNMAP support for zvols (via comstar) (Dan McDonald)
  - TRIM support (in Linux and FreeBSD)

Platform Diversity on Open ZFS



FreeBSD

- 750 commits
- 52 Contributors
- 5+ organizations

Illumos

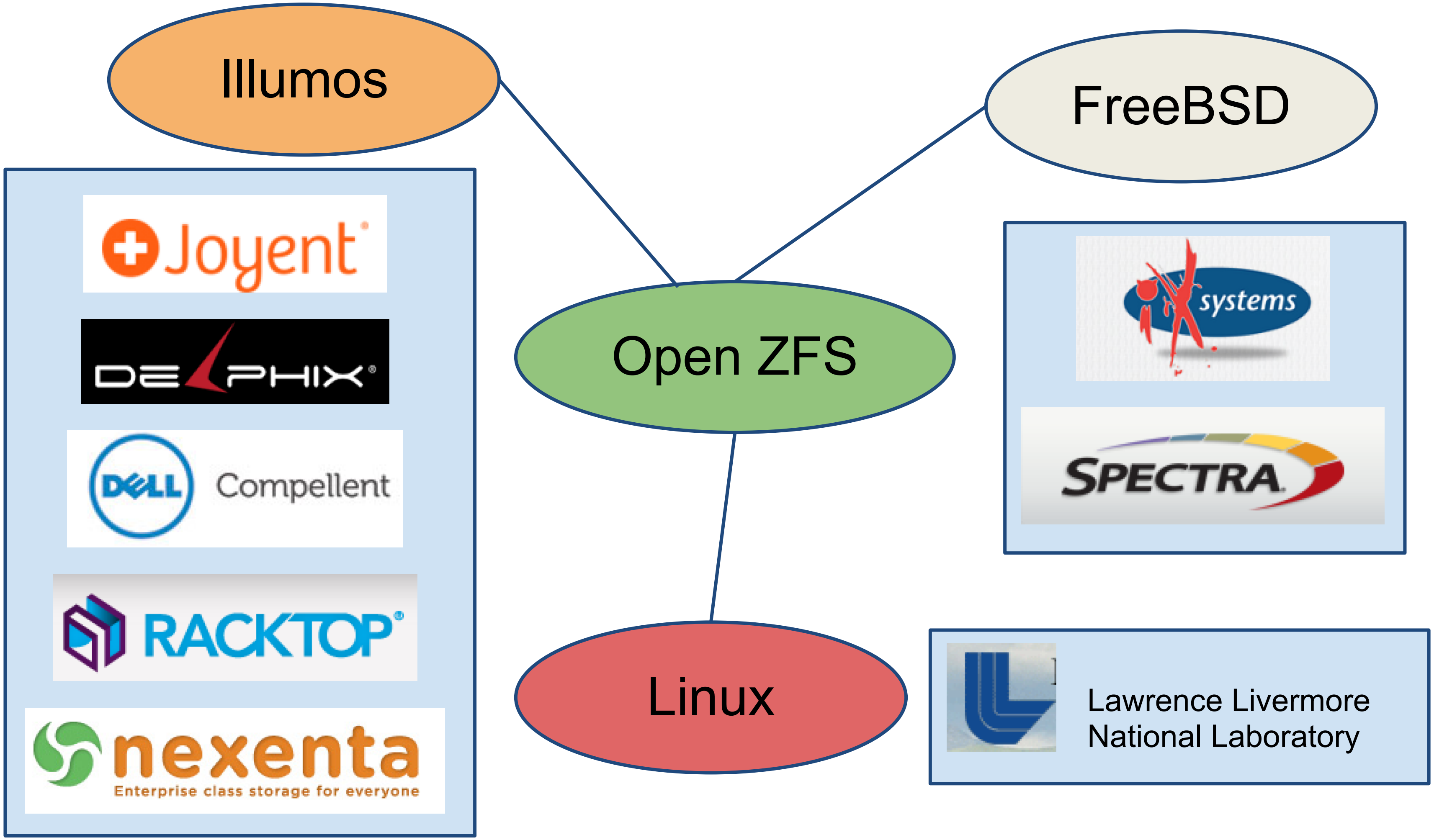
- 90 commits
- 29 Contributors
- 15 organizations

Open ZFS

Linux

- 980 commits
- 50 Contributors
- 35 organizations

Platform Diversity on Open ZFS



Illumos

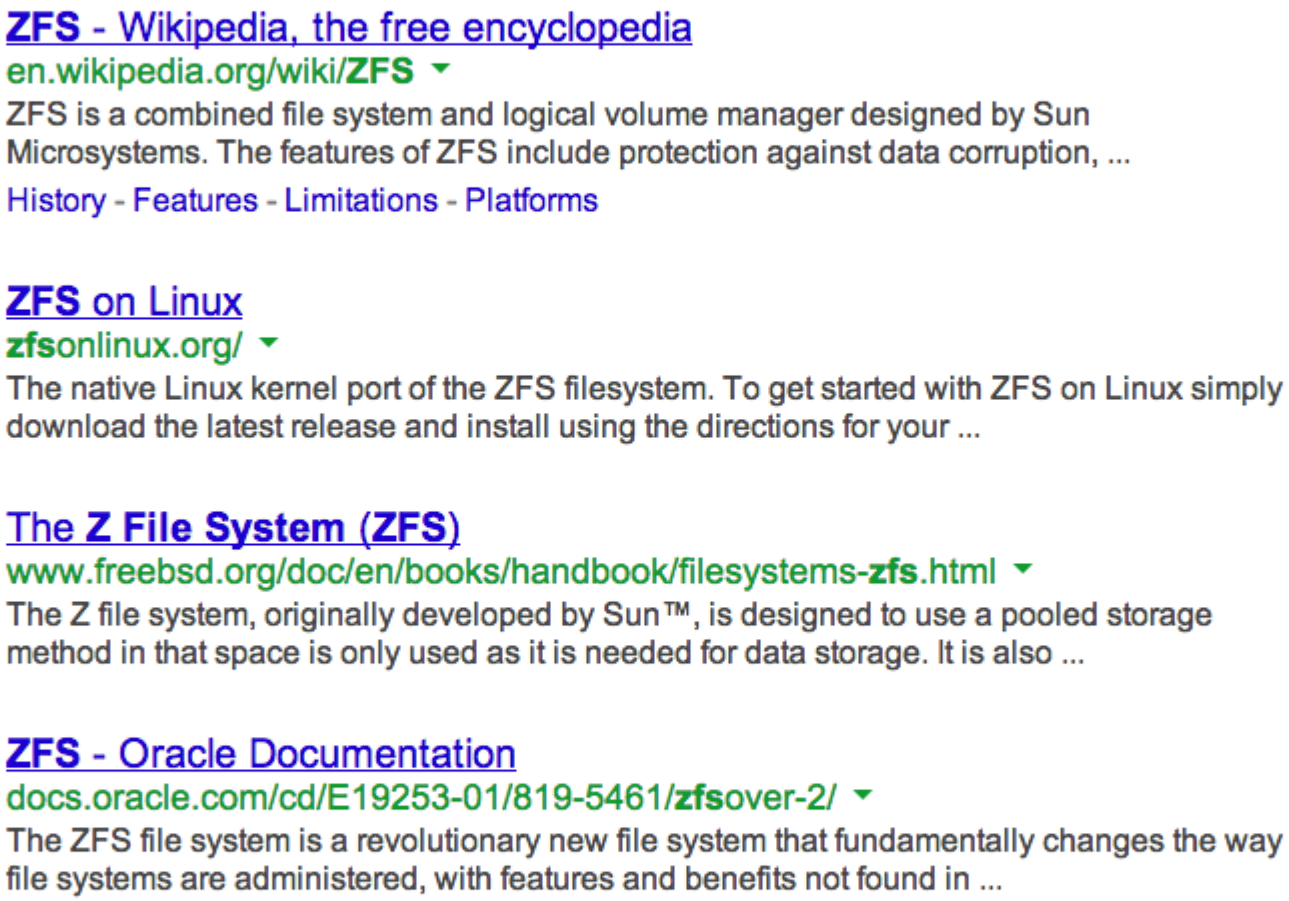
FreeBSD

Open ZFS

Linux

Lawrence Livermore National Laboratory

What is ZFS?



ZFS - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/ZFS
ZFS is a combined file system and logical volume manager designed by Sun Microsystems. The features of ZFS include protection against data corruption, ...
History - Features - Limitations - Platforms

ZFS on Linux
zfsonlinux.org/
The native Linux kernel port of the ZFS filesystem. To get started with ZFS on Linux simply download the latest release and install using the directions for your ...

The Z File System (ZFS)
www.freebsd.org/doc/en/books/handbook/filesystems-zfs.html
The Z file system, originally developed by Sun™, is designed to use a pooled storage method in that space is only used as it is needed for data storage. It is also ...

ZFS - Oracle Documentation
docs.oracle.com/cd/E19253-01/819-5461/zfsover-2/
The ZFS file system is a revolutionary new file system that fundamentally changes the way file systems are administered, with features and benefits not found in ...

What is ZFS?

- ZFS is not (just) an Oracle technology
- ZFS is not (just) an illumos technology
- ZFS is our technology
    - shared development on several platforms
        - illumos, FreeBSD, Linux
    - "Open ZFS"

What is Open ZFS?

- Open ZFS is our ZFS
- Possibilities:
  - Website
    - links to corporate users?
  - Community
    - Mailing list?
    - Virtual office hours?
  - Common test methodology & test suite?
    - Public automated test results?
  - Common repo?
  - Everything but ZPL should have 100% shared codebase?
    - also for userland?

Open ZFS @ BSDCan

May 2013
Matt Ahrens
mahrens@delphix.com
@mahrens1

Features unique to Open ZFS

- Space accounting
- Debugging and testing improvements
- Performance improvements
- CLI usability
- Programmatic usability

Space accounting - only in Open ZFS

- refratio property
- written and written@... properties
- zfs **send stream size estimation**
- zfs send **progress reporting** (Bill Pijewski)
- zfs destroy -nv <snapspec> tells how much space is shared by list of snaps

Debugging & testing - only in Open ZFS

- zfs ioctl args in truss (Gordon Ross)
- ztest backwards compatibility option (Chris Siden)
- **ZFS test suite** returned to working order
    - many tests added
    - new, simpler test framework developed
    - (John Kennedy)
- more complete “zpool history” logging

Performance - only in Open ZFS

- **single-copy ARC** (George Wilson)
- imbalanced LUNs performance improvements (George Wilson)
- zfs destroy <snapspec> destroys many snaps at once
- empty_bpobj feature
- **background destroy** of filesystems, >100x faster destroy of clones (Chris Siden)
- improved performance for **partial-block writes** (in Spectra Logic)
- per-zone i/o throttling (in SmartOS)

CLI Usability - only in Open ZFS

- “zfs get -t <type>” (Andrew Stormont)
- “zpool iostat” separates out log devices (Mark Harsch)
- “zfs get” takes mountpoint (Sham Pavman)
- zpool “comment” property (Dan McDonald)
- per-vdev space usage & expandsz (George Wilson)
- zdb manpages written (Richard Lowe)

Programmatic Usability - only in Open ZFS

- clones property
- **libzfs_core**
- zfs snapshot <arbitrary snaps>
- zfs destroy <arbitrary snaps>
- background destroy of filesystems

Features - only in Open ZFS

- feature flags (Chris Siden & Basil Crow)
- zpool reguid (Garrett D’Amore & George Wilson)
- dump to RAID-Z (in SmartOS)
- restore “aclmod” property (Albert Lee)
- SCSI UNMAP support (Dan McDonald)
- TRIM support (in Linux and FreeBSD)
- support for **4k sector size devices** (George Wilson)
- SPA i/o deadman (George Wilson)

ZFS Send and Receive

- NFS + low-bandwidth, high-latency link + rsync
  - result: thumb twiddling
- ZFS send + receive
  - latency-insensitive (unidirectional)
  - quickly finds & sends only the changed blocks / files
- Remote replication

ZFS Send progress monitoring

- 2005: incremental zfs send
  - just wait a while
- 2011: zfs send size estimation
- 2012: zfs send progress reporting
  - from Bill Pijewski of Joyent

The future of Open ZFS

- lz4 compression (Saso Kiselkov)
- compressed l2arc (Saso Kiselkov)
- performance on fragmented pools (George Wilson)
- observability -- zfs dtrace provider
- sub-block write performance
  - (Justin Gibbs & Will Andrews)
- wild, application-specific solutions
  - easily extensible architecture
  - modern, object oriented implementation
  - enables things like cross-platform provisioning

ZFS @ BSDCan

May 2013
Matt Ahrens
matt@delphix.com
@mahrens1