Ceph: the long (long) road from research to production

Sage Weil
new dream network / DreamHost
Outline

- Background
- Research beginnings
- Evolution
- Adoption
- Business
- Future
Background
Ceph architecture
What makes it different

- Robust distributed object store
  - Functional object placement
  - Direct client access to data
- Scalable and adaptive metadata cluster
  - POSIX
  - Directory snapshots
  - Recursive accounting
- Open source
Multiple interfaces

- RADOS distributed object store
  - Librados
  - RESTful radosgw proxy (S3 compatible)
- RBD distributed block device
  - Scalable, reliable alternative to iSCSI/AoE/FCoE
- Ceph scalable distributed file system
  - Exabytes; POSIX; HPC
Research beginnings
UCSC

- Petascale object-based storage systems
  - LLNL/LANL/Sandia
  - Thousands of nodes, billions of files
  - Concurrent access to same file, directory
- Dynamic subtree partitioning
  - Balance load across cluster based on popularity of subtrees in hierarchy
Prototype

- First MDS cluster implementation (2004)
  - Simple message passing
  - Trivial object storage targets
- Initial distributed object store (2005)
- Flexible object placement (2005)
- Paxos monitor cluster (2006)
- Ebofs object file system (2004/2006)
Release!

- Paper published (2006)
- Moved project to SourceForge
  - Name (after the class cephalopod)
  - Logo
  - LGPL license

- Wait for the users and contributors to come running...
The next step

• No open source enterprise storage options
  – Most Linux file systems don't scale
  – Lack enterprise features (e.g., snapshots)
  – No community

• Enterprise players want you, not your project

• Continued development
  – Supported by new dream network / DreamHost
Successful open source

• Most projects are islands
  – Individual developers, few users
  – Few high profile projects

• Key factors
  – Merit
  – Users
  – License
  – Community
Users live in the real world

- Users are conservative, especially with storage
  - Nobody wants to lose data
  - Performance matters
  - Robust error handling
  - fsck
- Fuse isn't good enough
  - NFS/pNFS or native Ceph client
- No users!
Licensing

- GPL vs BSD
  - LGPL as compromise
- Dual licensing
- Copyright assignment
- Few flavors
  - Eucalyptus
  - MySQL
  - Lustre
Community

- Users
  - Testing, early adoption
  - Customers
- Developers
  - Contribute
  - Limit dependency on individuals, organizations
- Open development and transparency
  - E-mail list, IRC, public bug tracker
Ambitious feature set

- Native kernel client (2007-)
- Architecture continuing to evolve
  - Per-directory based snapshots (2008)
  - Recursive accounting (2008)
  - Object classes (2009)
  - REST gateway (2009)
  - Strong authentication (2009)
  - RBD (2010)
Local object storage

- **OBFS**
  - Designed specifically for object workloads

- **EBOFS**
  - Extents, Btrees
  - Custom interface (transactions)

- **Btrfs**
  - Robust, well supported and optimized
  - Kernel-level cache management
Core code

- Message passing reimplemented 4 times
  - MPI, TCP (*3)
  - Fixed cluster → dynamic
  - Node naming → address based
  - Stateful → stateless

- Data serialization
- Configuration
- Security
Linux kernel client

- Initial version in 2008
  - Substantial revision of client/MDS protocol
- Upstream review
  - Difficult to solicit reviews
  - Linux community can be demanding, fickle
- Linus initially declined
  - Perceived user demand
- Finally merged in 2.6.34
Adoption
Adoption is key

- Users → Testing
- Testing → Stability
- Stability → Users
- Stability → Support
- Support → Paying users
Upstream

- Kernel, Fedora 12, Ubuntu 10.10
  - Easier to test
  - Exposure
- More contributors
  - Many fixes on kernel side
- Backward compatibility
  - Most users run old kernels
Stabilization

• Phases
  – Distributed object storage
  – Single MDS
  – Snapshots
  – Clustered MDS

• Simpler use cases
  – Object storage
librados, radosgw

- Most large users don't need POSIX or files
  - S3 object storage immensely popular
- radosgw is a simple S3-compatible proxy
  - fastcgi
  - Talks directly to distributed object store
  - Coexist with librados
- Cloud infrastructure
Rados block device (RBD)

- Network block device striped over objects
  - Scalable, reliable, thinly provisioned
  - iSCSI, DRDB, nbd
  - Snapshots
- Native Linux kernel block device
- Qemu/KVM storage driver
  - librados
  - Huge level of community interest
Leverage existing communities

- Hadoop
  - Diverse, active community
  - Lots of temporary, intermediate data
  - Two glue layers (uclient, kclient)
  - Some talk, little action

- Hypertable
  - Distributed database
  - Small community, little adoption
Step 3: Profit
Infrastructure

- DFS is key infrastructure for hosted services
  - Web hosting
  - Backup
  - Cloud storage
- Enterprise storage is expensive
  - Prudent investment for organizations with large data center infrastructure
Scaling development team

- System is large and complex
  - Easy problems have been solved
  - Hard to delegate
- Recruiting
  - Location
  - Experience
- I'd rather be coding
Support

- Traditional revenue source
  - Enterprise needs someone to lean on when there are problems
  - Value is in developer expertise, not code
- Simple deployments are very stable
  - Object store
  - RBD
  - Single MDS
How is it going?

- **Successes**
  - Feature set
  - Community
  - RBD

- **Failures**
  - QA
  - Performance tuning
  - Hadoop
Future

• Community
  – Cloud computing
  – HPC, exascale
  – Hadoop

• Team
  – QA, developers
  – Support infrastructure
  – Business infrastructure
Thanks