Lustre* Future Features

Andreas Dilger

April 8, 2014

Agenda

Overview of feature submission process

Features proposed for 2.7 and later releases

More details on features not already covered at LUG'14

Questions

There may be other features under development.

Please share any development plans with the community.

Feature Submission Process

T minus 3 months: Features must be **LANDED**

- Feature description and design in Jira ticket
- Patch submission must be started **well before** feature cutoff date
- Need to test, inspect, update, retest, integrate with other new features

T minus 2 months: Documentation and test plan completed

- Plan for new functionality/performance/load testing
- Manual updates for user-facing features/tunables
- Unix man pages for tools and APIs better if with patch itself

T minus 1 month: only bug fixes landed after this point

No features are guaranteed to be in any release

- Train model only includes features that are ready by feature cutoff
- Conversely, smaller features not on roadmap can be landed if no major conflicts

http://wiki.opensfs.org/Lustre_Community_Development_in_Progress

Features planned for 2.7 and beyond

- DNE Phase 2 Asynchronous Updates (Intel, 2.7)
- LFSCK Phase 3 DNE Consistency Checking (Intel, 2.7)
- UID/GID Mapping (IU, 2.7)
- Dynamic LNET Config (Intel, 2.7)
- OST-specific setstripe (Intel/Fujitsu, 2.7)
- Quota for Projects (DDN, 2.8)
- Kerberos revival (Xyratex)
- T10 DIF/PI end-to-end checksum (Xyratex)
- 16MB Bulk RPCs (Intel)
- Shared Secret Key Encryption (IU)
- Layout Enhancement (Intel)
- Data on MDT (Intel)
- File Level Replication (Intel)

Features Discussed In Other Presentations

Dynamic LNET Config (Intel, 2.7)

- Allow runtime configuration of networks and routers

Quota for Projects (DDN, 2.8)

- Track quota with "project" identifier from parent directory

Layout Enhancement (Intel, design complete)

- Infrastructure for Data on MDT, File Level Replication, others

Data on MDT (Intel, design complete)

- Store small files directly on the MDT

File Level Replication (Intel, design complete)

- RAID-0+1 layout for files

Distributed Namespace (DNE) – Intel (2.7)

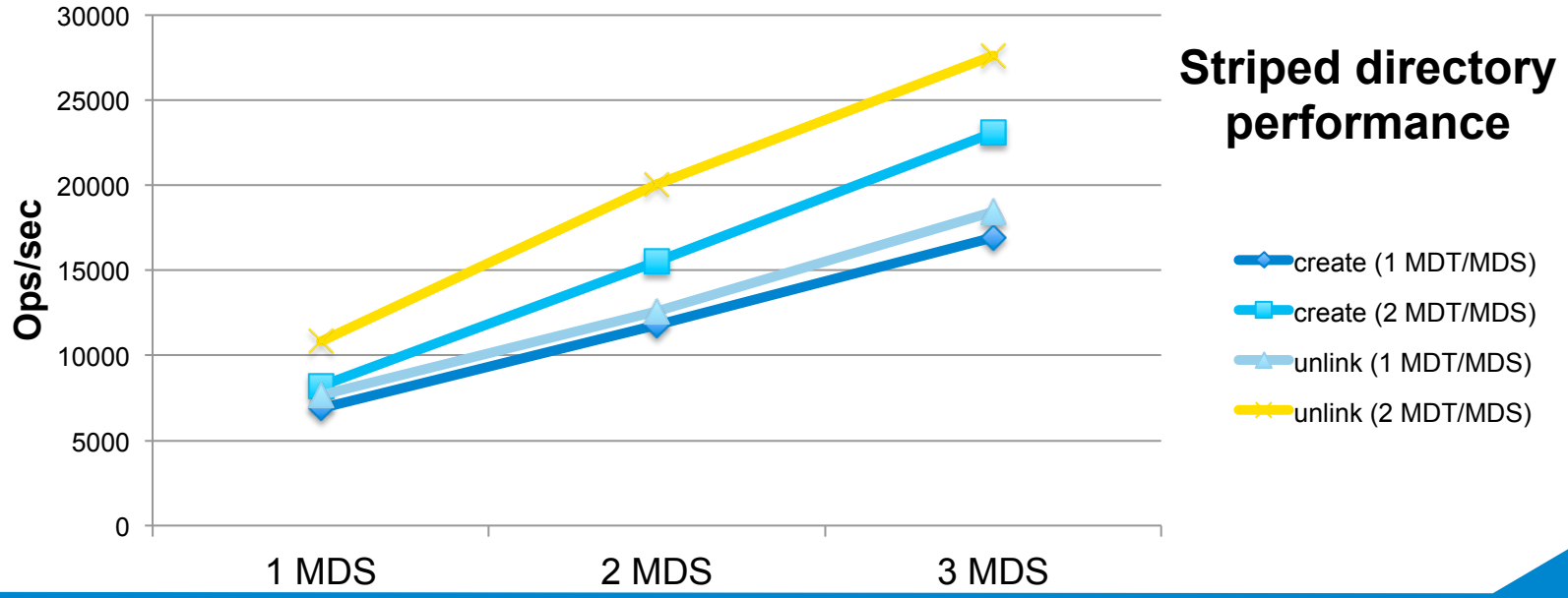
Phase 2

Striped directories & Inode migration tool (2.6, LUG 2013)

- Allow a single directory to be distributed over multiple MDTs
- Move files & directories between MDTs without copying data

Asynchronous remote updates (2.7)

- Improved performance for cross-MDT ops (mkdir, rmdir, striped dir, …)
- Allow rename and hard link to remote MDTs



Lustre* Filesystem Check (LFSCK) – Intel (2.7) Phase 3

Builds on previous LFSCK Phases (2.3-2.6, LUG 2013)

- Phase 1: OI Scrub for local inode iteration and OI reconstruction
- Phase 1.5: Local namespace check (name->FID, linkEA->parent)
- Phase 2: MDT-OST consistency check (LOV EA check, orphans)

MDT-MDT consistency checking (2.7)

- Verify remote directory and file links
- Reconnect remote orphan directories to `lost+found`
- Fix directory entries referencing missing inodes

UID Mapping – Indiana University (2.7)

Groups of nodes with different UID/GID maps (2.6, LAD 2013)

- WAN or other separate administrative domains
- UID/GID maps are maintained only on a nodemap granularity

Remote cluster nodes defined by client NID range (2.6)

- Optionally authenticated by Shared-Key Crypto authentication
- *Nodemap* can be one node or a whole campus

Map remote UID/GID to local values on MDS/OSS (2.7)

- Does not need any changes to remote clients
- Store local UID/GID on MDS for permissions, ACLs, quota
- Map remote UID/GID on OSS for quota
- Allow squashing UID/GID outside of nodemap to block remote file access

OST-Specific lfs setstripe – Intel/Fujitsu (2.7)

Allow specifying individual OSTs at file creation

Fine grained control of object placement

Useful for picking replicas with File Level Replication

```
lfs setstripe --ost-list 2,4,6,8 /mnt/lustre/new_file
lfs setstripe --ost-list [0-8] /mnt/lustre/next_file
```

Kerberos Revival – Xyratex

Fix problems in current Kerberos code (LAD 2013)

- Was never an officially supported feature
- Has been untested for several years

Interest in Lustre* Kerberos usage increasing

Co-exist with Shared Key Crypto

- Share same GSSAPI infrastructure
- Improved testing and code coverage for both projects

T10 DIF/PI Checksum – Xyratex

Bulk RPC checksum improvements (2.3)

- Use Kernel CryptoAPI for hardware acceleration

Allow end-to-end checksums to disk (LUG 2012)

- 16-bit checksum + 32-bit block address + 16-bit "app tag"
- Leverage kernel DIF infrastructure
- Send per-sector checksums with each bulk RPC
- Depends on disks/RAID controllers with T10 PI support

16MB Bulk RPC – Intel/DDN

4MB OST RPC support added (2.4)

- Improve streaming RAID read/write performance
- Send multiple LNET RDMAs over network
- Didn't change IO request engine significantly

16MB+ Bulk RPCs for further improvement

- Improve client IO request engine to avoid regressions
- Improve client/server memory handling for large RPCs
- Allow larger RPC request sizes on OST for random IO

Shared Key Crypto – Indiana University

Simplified node authentication and RPC encryption (LAD 2013)

- WAN or other separate administrator domains
- Use existing Lustre GSSAPI/sptlrpc infrastructure from Kerberos

Shared secret key is known by clients and servers

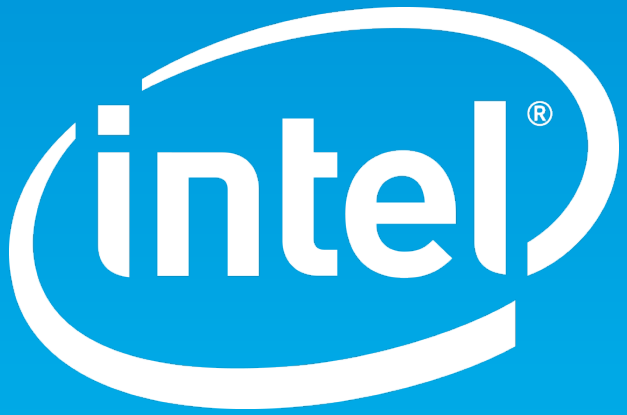
- Key distribution external to Lustre* (USB key, phone, (e)mail, pigeon)
- Different keys for different client clusters
- Servers can understand multiple keys per cluster
- Rotate keys as needed, lifetimes can overlap

Authenticate remote *nodes* instead of *users* like Kerberos

Uses AES-128 encryption

- Flexible to allow other algorithms in the future

Questions?


intel