Lustre* 2.8 and Beyond

Andreas Dilger, High Performance Data Division

Development Process Changes

Features are not necessarily be tied to specific releases

- Versions and features listed here are only targets and not guaranteed

Feature/component landings will be more "all or nothing"

- Still possible to land independent functional parts of a feature
- Avoids landing only a few (otherwise useless) patches of a feature

More focus on testing patch series before landing

- Whole patch series to be landed via merge commit after testing

Improved focus on documentation, comments, man pages

- Make the code easier to understand and maintain in the future

Overview of Features

Features at or near completion

- LFSCK Phase 4 - Performance Improvements
- DNE Phase 2 Striped Directories - Asynchronous Commits
- Client IO Simplification and Speedup

Features starting early development

- Multiple metadata-modifying RPCs (multi-slot last_rcvd)
- First generation Intel Omni Path Fabric / Dynamic LNet Configuration 2
- ZFS* Enhancements
- Protocol Documentation
- Data on MDT Prototype (DoM)
- Progressive File Layout Prototype (PFL)

LFSCK Phase 4 (Intel/OpenSFS 2.8)

LFSCK performance improvements (Phase 4)

- Improve object iteration, don't load objects unnecessarily
- Avoid a full scrub if only a few objects are found inconsistent
  - Tunable, launch full scrub if more than 60 errors within 60s
- Limit DLM locking to only affected name instead of whole directory
- Predict locking based on recent history
  - LFSCK doesn't lock by default, only lock & reverify on inconsistency
  - If errors recently seen LFSCK locks objects before doing checks
- Improved logging of LFSCK-detected inconsistencies

LFSCK Phase 4 is the final phase of this project

http://wiki.opensfs.org/images/3/3c/LFSCK_Performance_SolutionArchitecture.pdf

http://cdn.opensfs.org/wp-content/uploads/2013/04/Zhuravlev_LFSCK.pdf

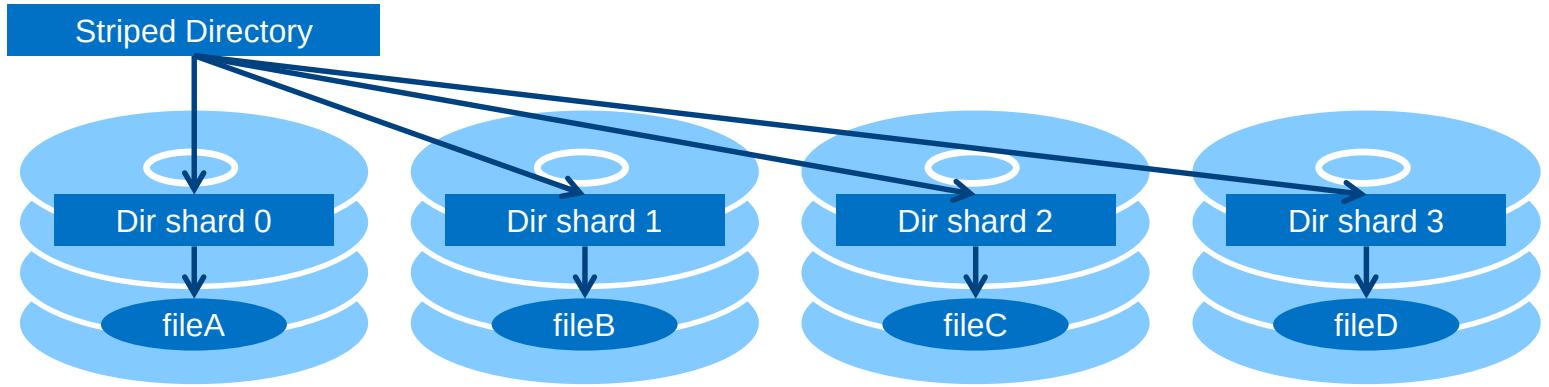
intel Software

DNE Phase 2 Striped Directories (Intel/OpenSFS 2.8)

Spread a single directory across multiple MDTs

- Reduce contention, improve performance for large directories
- Directory layout + name hash locates slave MDT directory entry
- Directory shard on each MDT independent (lock, lookup, modify)
- Inode created on the same MDT as name entry
- Tool to migrate directories from one MDT to another

DNE Phase 2 Async Commits is the final phase of this project



DNE 2 Asynchronous Commit (Intel/OpenSFS 2.8)

Change ***within*** MDT (mkdir, rmdir, rename) **never** synchronous

DNE remote/striped directory create synchronous in 2.4-2.7

- Cross-MDT rename() or link() weren't working (returned -EXDEV)

Async commit implements distributed DNE recovery

- Each target (master/slave) writes a full redo log of all updates
- If *any* target commits a change it can be replayed on *all* involved targets
- Ensures all-or-nothing semantic for namespace-visible changes
- Reduced latency for remote/striped directory creates
- Allow rename() and link() to work correctly across MDTs
- Foundation for future features (e.g. cross-MDT mirrored objects)

http://wiki.opensfs.org/images/f/ff/DNE_StripedDirectories_HighLevelDesign.pdf

intel Software

Client IO Cleanup/Speedup (Intel/OpenSFS 2.8+)

Clean up CLIO code and interfaces

- Simplify complex internal locking code
- Replace old ioctl interfaces with proper methods
- Remove non-functional interop code for WinNT and MacOS
  - Remove extra abstraction layer complexity and overhead
- Remove access to LOV internals throughout code
  - Preparation for handling of more complex file layouts (e.g. PFL)

Client Performance Improvements

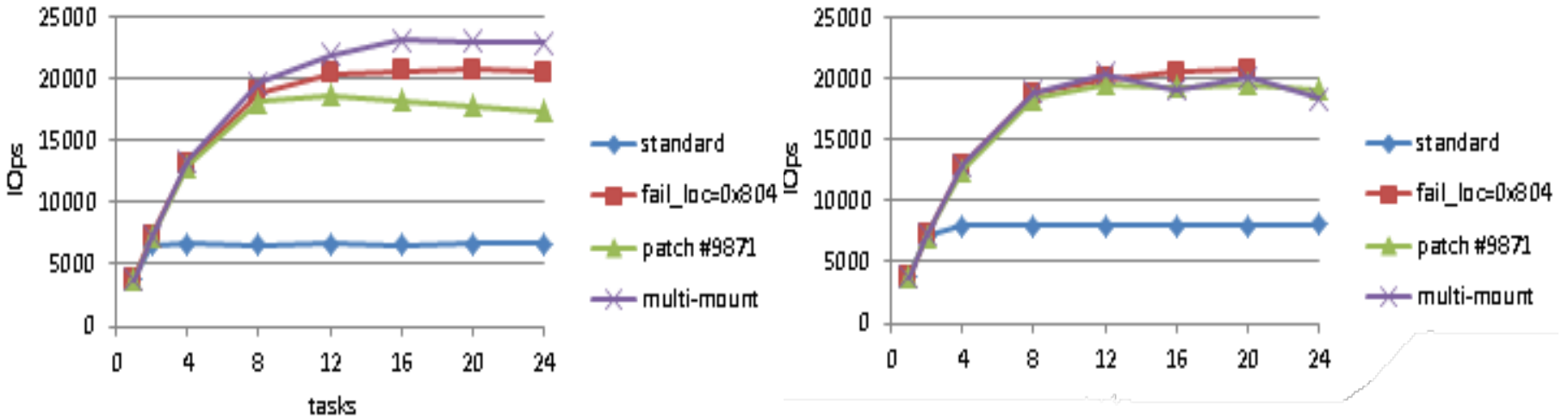
- Larger RPC sizes for improved allocation and disk IO
- Single-threaded IO performance improvements

http://wiki.opensfs.org/images/b/b7/CLIOSimplificationDesign_HighLevelDesign.pdf

Client Metadata RPC Scaling (Bull/Intel 2.8)
(aka multi-slot last_rcvd)

Currently limited to one modifying RPC (+close) per client
- last_rcvd slot on MDT for each client to reconstruct reply
- Many concurrent clients limited by MDS performance

Dynamic log on MDT for multiple saved RPC replies per client
- Each metadata-modifying RPC has a separate tag/index
- Single client multi-threaded create/unlink performance improved



https://jira.hpdd.intel.com/browse/LU-5319

Intel® Omni-Path Architecture Gen 1 (Intel 2.8)
Dynamic LNet Config Phase 2 (Intel 2.9)

LNet support for Intel Omni-Path host fabric interface (HFI)

- Next generation interconnect from Intel
- Compatible with OFED verbs interface
- May need LNet o2iblnd tuning for best performance

Improved Dynamic LNET Config

- Per-NI tunables instead of per-LND
- Auto-tune parameters based on network interface type
  - Optimize for Mellanox*, Intel® True Scale HCA, and Intel® Omni-Path HFI

ZFS Enhancements (Intel, 2.8+)

Changes for ZFS OSD (2.8)

- 1MB+ ZFS blocksize (IO performance)
- Read IO optimization (IO performance)
- ZIL support for fast sync (IO & metadata performance)

Changes to core ZFS code (2.10?)

- Parity declustering (availability)
- Distributed hot spares (availability)

Protocol Documentation (Intel/OpenSFS)

Document the PTLRPC wire structures, message flow, states

- POSIX operations (mount, open, close, setattr, create, unlink, etc)
- MDS state handling (connect, disconnect, FLD, SEQ, PING, etc)
- IO operations (read, write, truncate, setattr, grant)
- OSS state handling (precreate, orphan cleanup, destroy, etc)
- Quota management
- OUT distributed updates (DNE, LFSCK, Async Commit)

http://wiki.opensfs.org/Contract_SFS-DEV-005

Data on MDT Prototype (Intel/OpenSFS)

Efficiently store small files on the MDT(s)

- Avoid OST BRW RPC + disk seek + OST lock for each file access
- Use small-file optimized MDT storage (RAID-10/SSD/NVRAM)
- Avoid RAID-5/6 read-modify-write for small writes

Space usage on MDT(s) managed by quota

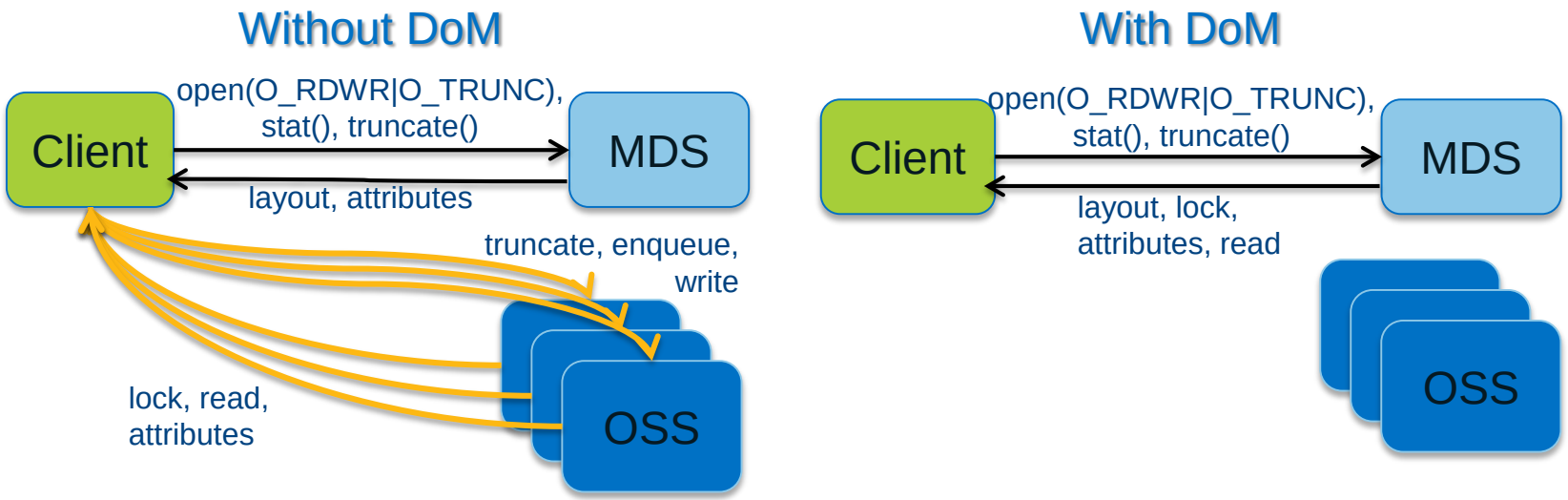
Small files are determined by the file layout

- Maximum MDT file size can be specified by min(user, admin)
- Typically expected to be <= 1MB, dependent on MDT space

Complimentary with DNE 2 striped directories

- Scale small file IOPS horizontally with multiple MDTs

Data on MDT Implementation



DoM layout chosen at file creation time like files on OSTs

- Can't do it after write because objects are allocated at open()
- Default DoM striping on subdirectories inherited by newly created files

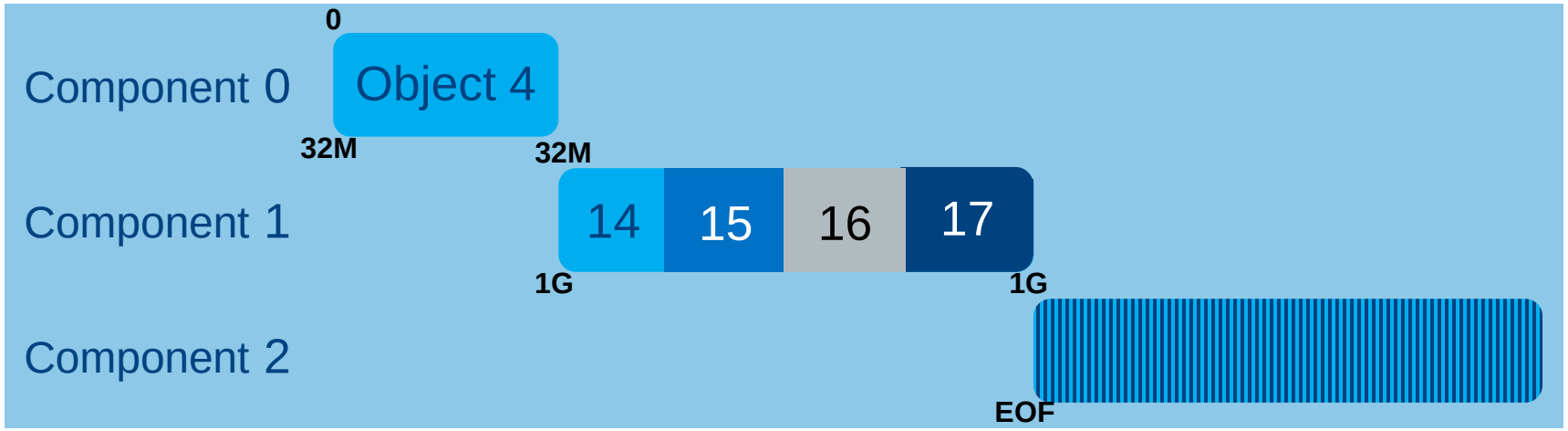
http://cdn.opensfs.org/wp-content/uploads/2014/04/D1_S10_LustreFeatureDetails_Pershin.pdf

http://wiki.opensfs.org/images/b/be/DataonMDSDesign_HighLevelDesign.pdf

Progressive File Layout Prototype (Intel/ORNL)

Allow compound layouts for regular files

- Component layouts describe one or more extents of a file
- Layout extents do not overlap for PFL files
- Start with few stripes, increase stripe count as file size increases
- Balance lower overhead vs. performance and space balance



Miscellaneous features

Code cleanups (Cray*/Intel®/ORNL)

- Remove dead code and useless wrappers
- Update to match upstream kernel coding style
- Port patches to/from upstream kernel
- Clean up or eliminate server kernel/ldiskfs patches

Project Quotas (DDN*)

- Allow quota tracking on subtrees independent of UID/GID

Network Authentication and Encryption (Bull*/IU*/Seagate*)

- Kerberos user/node authentication, RPC encryption
- Shared Secret Key node authentication, RPC encryption

Legal Information

- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.
- This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.
- The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.
- Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.
- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at http://www.intel.com/content/www/us/en/software/intel-solutions-for-lustre-software.html.
- Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase.
- For more complete information about performance and benchmark results, visit http://www.intel.com/performance.
- Intel, the Intel logo and Intel® Omni-Path are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others

intel Software


intel
Software