Lustre* 2.9 and Beyond

Andreas Dilger

Principal Lustre Engineer

High Performance Data Division

Focus on Performance and Ease of Use

Beyond just looking at individual features...

- Incremental but continuous improvements
- Performance and scalability enhancements
- Usability and manageability
- Access control and data security
- Improved filesystem availability

Overview of Upcoming Lustre* Features

Features underway or landed for 2.9

- ZFS Enhancements (Intel®, LLNL)
- UID/GID mapping, Shared Secret Key (IU, OpenSFS*)
- Subdirectory mounts, Server IO advice (DDN*)
- File lockahead (Cray*)

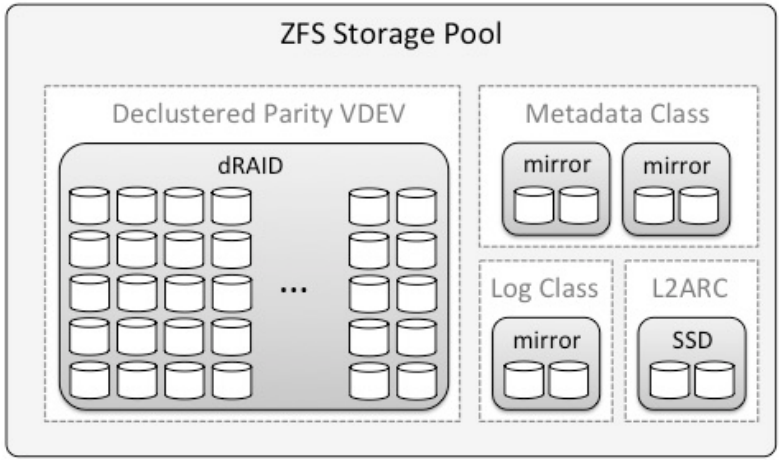
Features starting or underway for 2.10+

- Multi-Rail LNet (SGI*, Intel)
- Project quotas (DDN)
- Data on MDT (Intel)
- Composite File Layouts (Intel, ORNL)

ZFS Enhancements (Intel, LLNL 2.9+)

Changes for using ZFS better

- 1MB+ ZFS blocksize (IO performance, LLNL)
- Improved file create performance (Intel)
- Snapshots of whole filesystem (Intel)



Changes to core ZFS code

- Inode quota accounting (Intel)
- Large dnodes to improve xattr performance (LLNL)
- Declustered parity & distributed hot spaces to improve resilvering (Intel)
- Metadata allocation class to store all metadata on SSD/NVRAM (Intel)
- Reduce CPU with hardware-assisted checksums, compression (Intel)

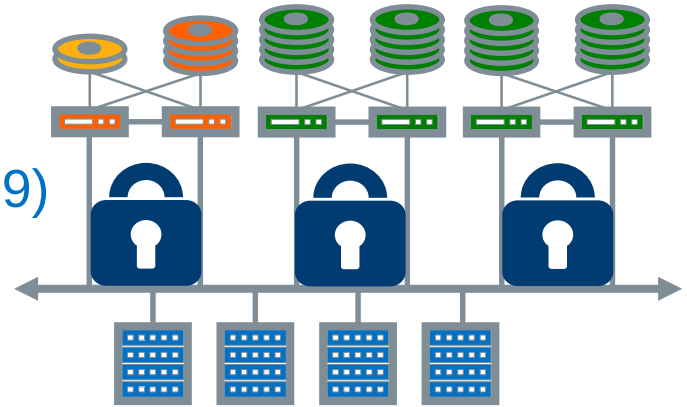
Data Security for All Environments

UID/GID Mapping and Shared Secret Key Crypto (IU, OpenSFS* 2.9)

- Data encryption for networks including RDMA (IB, OPA)
- Strong client node authentication into administrative node groups
- UID/GID mapping for WAN clients
- Block unauthorized clients by network

Data isolation via filesystem containers (DDN 2.9)

- Subdirectory mounts with client authentication
- Usable with hosted, isolated environments



Miscellaneous Features

Code cleanups (Cray, Intel, ORNL 2.9+)

- Update to match upstream kernel, port patches to/from kernel
- Patchless server kernel, ldiskfs patch cleanup
- Dead code removal for maintainability, security

Server-side IO advice - `ladvise` (DDN 2.9)

- Tunables per file/extent to manage internal cache on OSS
- Object readahead and random IO blocksize
- Client write lockahead (Cray 2.9)

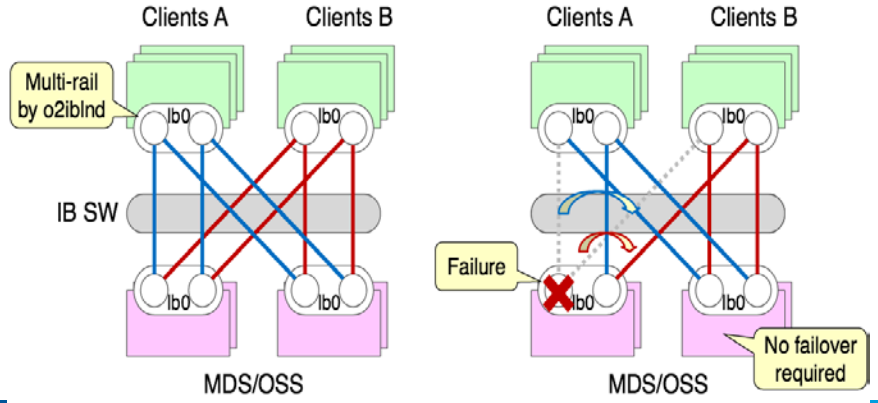
Project Quotas (DDN* 2.10)

- Allow quota tracking on directory subtrees independent of UID/GID
- Not strictly hierarchical, can be multiple trees with the same project

Networking Improvements (Intel, SGI 2.10)

Improved networking capabilities

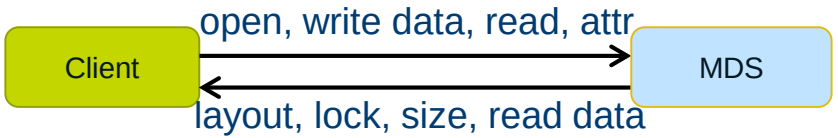
- Support for EDR and FDR InfiniBand, MLX5
- Intel OmniPath Architecture network support
- RPC crypto for RDMA networks like IB and OPA
- Multi-Rail support for all network types



Improved Small File Performance (Intel 2.10+)

Data-on-MDT optimizes small file IO

- Avoid OST overhead (data, lock RPCs)
- High-IOPS MDTs (mirrored SSD vs. RAID-6 HDD)
- Avoid contention with streaming IO to OSTs
- Prefetch file data with metadata
- Size on MDT for files
- Manage MDT usage by quota



Small file IO directly to MDS

Complementary with DNE 2 striped directories

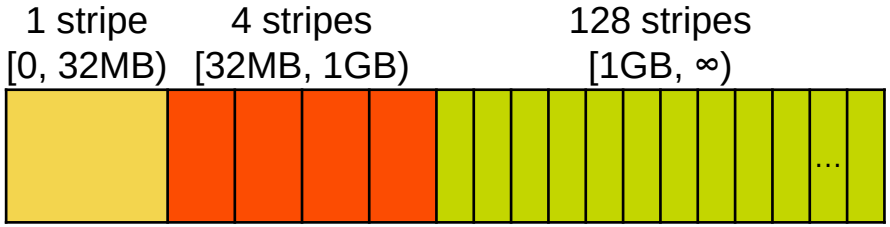
- Scale small file IOPS with multiple MDTs

Composite File Layouts (Intel, ORNL 2.10+)
Innovation in Storage Usage

Progressive File Layouts simplify usage and provide new options

- Optimize performance for diverse users/applications
- Low overhead for small files, high bandwidth for large files
- Lower new user usage barrier and administrative burden
- Multiple storage classes within a single file
  - HDD or SSD, mirror or RAID

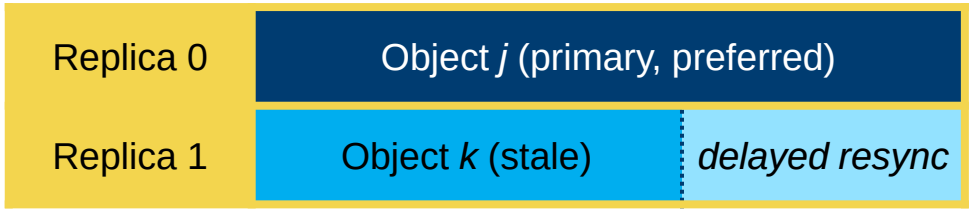
Example progressive file layout with 3 components



Improved Data Availability (Intel 2.11+)

File Level Redundancy provides significant value and functionality for HPC

- Configure on a per-file/dir basis (e.g. mirror input files and one daily checkpoint)
- Higher availability for server/network failure - finally better than HA failover
- Robustness against data loss/corruption - mirror or M+N erasure coding for stripes
- Increased read speed for widely shared files - mirror input data across many OSTs
- Replicate/migrate files between storage classes - NVRAM->SSD->HDD
- Local vs. remote replicas
- Partial HSM file restore
- File versioning, ...



Advanced Lustre* Research
Intel Parallel Computing Centers

Uni Hamburg + German Client Research Centre (DKRZ)

- Adaptive optimized ZFS data compression
- Client-side data compression

GSI Helmholtz Centre for Heavy Ion Research

- TSM HSM copytool for Lustre

University of California Santa Cruz

- Automated client-side load balancing

Johannes Gutenberg University Mainz

- Global adaptive Network Request Scheduler

Lawrence Berkeley National Laboratory

- Spark and Hadoop on Lustre

Legal Notices and Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure.

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 disclaims all express and implied warranties, including, without limitation, the implied warranties and 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.

Intel, the Intel logo and others 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®
experience
what's inside™