Lustre 2.4 and Beyond

Andreas Dilger
Software Architect
High Performance Data Division

November 14, 2012

Features Planned for Lustre 2.4 and 2.5

Features must be ready before feature freeze (-3 months)

- Not all features listed here are guaranteed to be in the specified release
- Only a subset of potential 2.4/2.5 features are listed here

Features covered in this presentation

- ZFS OST/MDT backing storage
- Distributed NamespacE (DNE) - Remote Directory, Stripe/Shard Directory
- Lustre File System ChecK (LFSCK) - FID-in-dir/LinkEA, MDT/OST, DNE
- Network Request Scheduler (NRS)
- Hierarchical Storage Management (HSM)
- Linux kernel client updates

ZFS OST/MDT Backing Storage (2.4)

Can leverage many ZFS features in Lustre 2.4

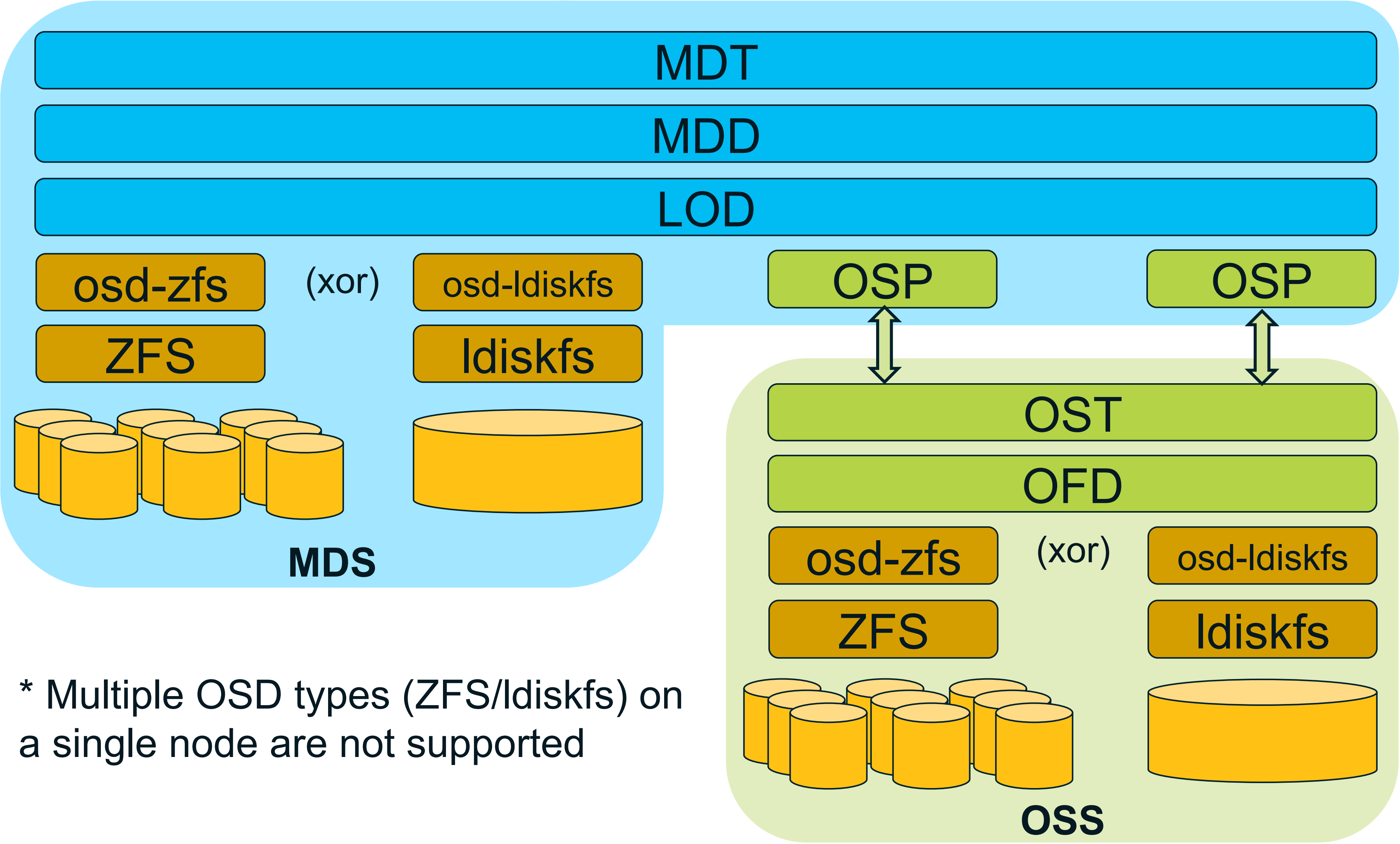
- Robust code with 10+ years maturity
- Data checksums on disk + Lustre checksums on network
- Online filesystem check/scrub/repair - *no more e2fsck!*
- Scales beyond current filesystem limits (object count/size, filesystem size)
- Easier management of many disks, commodity JBODs without RAID hardware
- Integrated with flash storage cache (L2ARC read cache)
- Data compression improves real-world IO performance and space utilization

Other features will need extra effort to work with Lustre

http://en.wikipedia.org/wiki/Zfs

http://zfsonlinux.org/lustre.html

Updated MDS/OSS Module Layering



* Multiple OSD types (ZFS/ldiskfs) on a single node are not supported

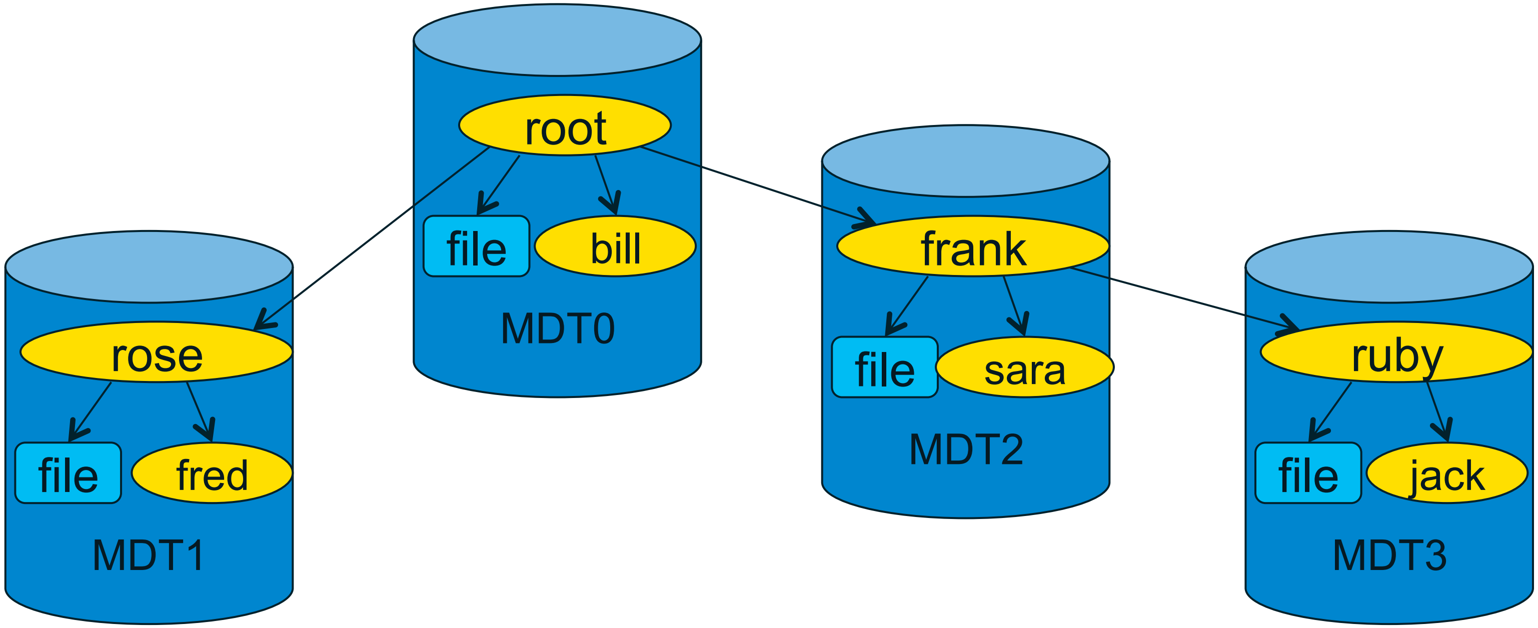
DNE Phase 1 - Remote Directory (2.4)

Subdirectories on remote MDT created by administrator

Scales namespace in similar manner to data servers

Allows isolated metadata performance for users/jobs

Shared OST IO bandwidth among all files on all MDTs



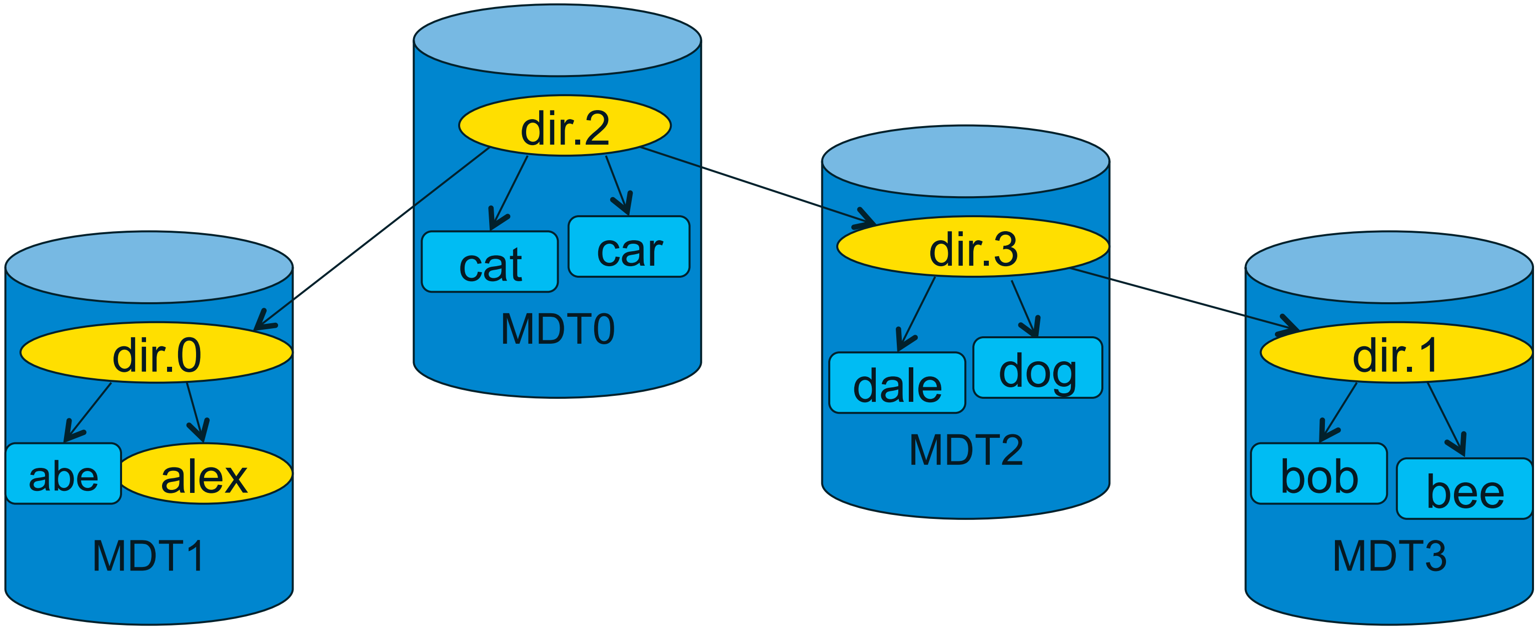
DNE Phase 2 - Shard/Stripe Directory (2.5)

Create new shard/stripe directory via `lfs mkdir`

Hash a single directory across multiple MDTs

Multiple servers active for directory/inodes

Improve performance for large directories



LFSCK Phase 1.5 - LinkEA, FID-in-dirent (2.4)

LFSCK is online Lustre File System ChecK tool

Verify Lustre File IDentifier (FID) stored in each directory entry

- Cannot preserve over file-level backups/transfer (`tar`, `rsync`, etc.)
- Not required for lookup, but important for `readdir()` performance
- Need to traverse each directory for name->{inode/FID} mappings
  - Piggy-backs on OI Scrub inode iteration (added in Lustre 2.3)
  - Do not need to traverse whole directory tree, piecewise for each directory
- If FID missing from dirent, get it from inode *`lma`* xattr (if available)

Verify inode->parent back-pointer in *`link`* xattr

- Stores `{parent directory FID, filename}` for each link to inode
  - Most inodes have only a single link
- Needed by `lfs fid2path` and `lustre_rsync` to generate path from FID
- Needs to be added to filesystems upgraded from Lustre 1.8

LFSCK Phase 2 - MDT/OST consistency (2.5)

Piggy-backs on OI Scrub inode iteration

- Does not depend on directory contents
- Sends RPCs to each OST for verification

Verifies MDT `lov` layout xattr matches OST objects

- Object must exist, cannot be referenced multiple times

Verifies OST `fid` xattr points back to matching MDT inode

- Allows detecting/creating missing objects

Verifies OST object is referenced by some MDT object

- Allows detecting/deleting orphan objects

Network Request Scheduler (NRS) (2.4)

Reorder RPC requests on the server

- Sort RPCs before data is sent across network
- Can sort many more requests than block layer elevator
- Optimize disk IO and load balance to improve efficiency

Currently implements multiple optimization policies

- FIFO (default, matches current behavior)
- Object-based round robin (optimize disk ordering)
- Client-based round-robin (improved fairness, avoid stragglers)

Allows other policies to be developed in the future (2.5)

- Guaranteed bandwidth to client(s) (min/max QOS, avoid starvation)
- JobID-based scheduling (finish one job’s IO before next job’s IO starts)

Hierarchical Storage Management (HSM) (2.4)

Originally developed by CEA France

Simple archive back-end interface

Initially supports HPSS and POSIX API

- HPSS copytool only available to HPSS users due to license
- POSIX copytool can interface to any archive with a “filesystem” interface

Uses CEA Robin Hood for policy engine

- Leverages Lustre ChangeLog to avoid scanning

Infrastructure usable for other projects

- Data migration within Lustre between storage pools/tiers
- Asynchronous file mirroring

Linux kernel Client Updates (2.4/2.5)

Desire to include Lustre client in upstream Linux kernel

- Ease of use for customer installations
- Reduce/eliminate lag for new kernel updates

Need to clean up ten years of legacy code to Linux style

- Code formatting is only a small part of this
- Remove/rename Solaris/WinNT/MacOS API wrapper functions
- Update code to use new VFS/VM interfaces

EMC is doing most of this work

- Changes landing incrementally in 2.3/2.4/2.5

Thank You

intel®