Lustre 2.4 and Beyond

Andreas Dilger
Software Architect
High Performance Data Division

September, 25 2012

Features Planned for Lustre 2.4 and 2.5

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

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

Features described in other presentations already

- HSM, Network Request Scheduler (NRS), ZFS, client kernel updates

Features covered in this presentation

- Distributed NamespacE (DNE) Phase 1 - Remote Directories
- Distributed NamespacE (DNE) Phase 2 - Stripe/Shard Directory
- Lustre File System CheckK (LFSCK) Phase 1.5 - FID-in-dir, LinkEA
- Lustre File System CheckK (LFSCK) Phase 2 - MDT/OST checks
- Lustre File System CheckK (LFSCK) Phase 3 - DNE consistency

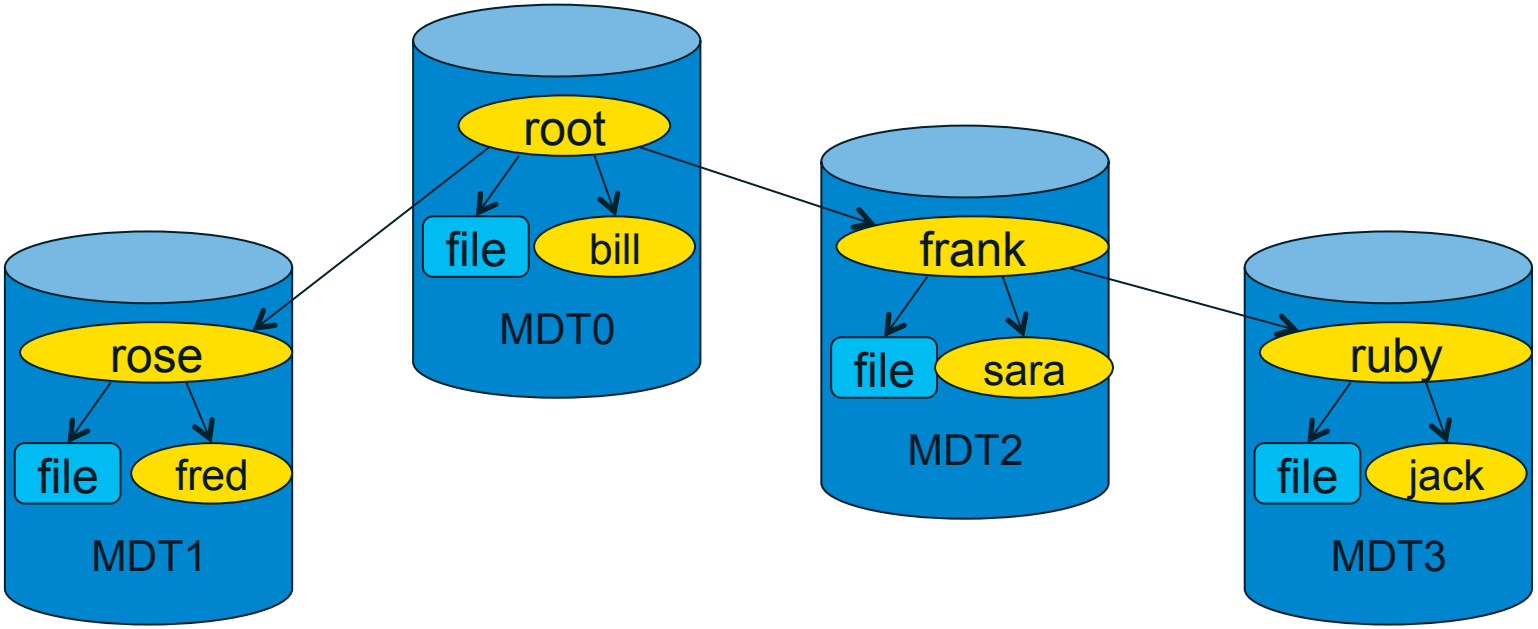
DNE Phase 1 - Remote Directory (2.4)

Subdirectories on remote metadata target by administrator

Scales namespace in similar manner to data servers

Isolated metadata performance for users/jobs

Shared OST IO bandwidth among all files on all MDTs

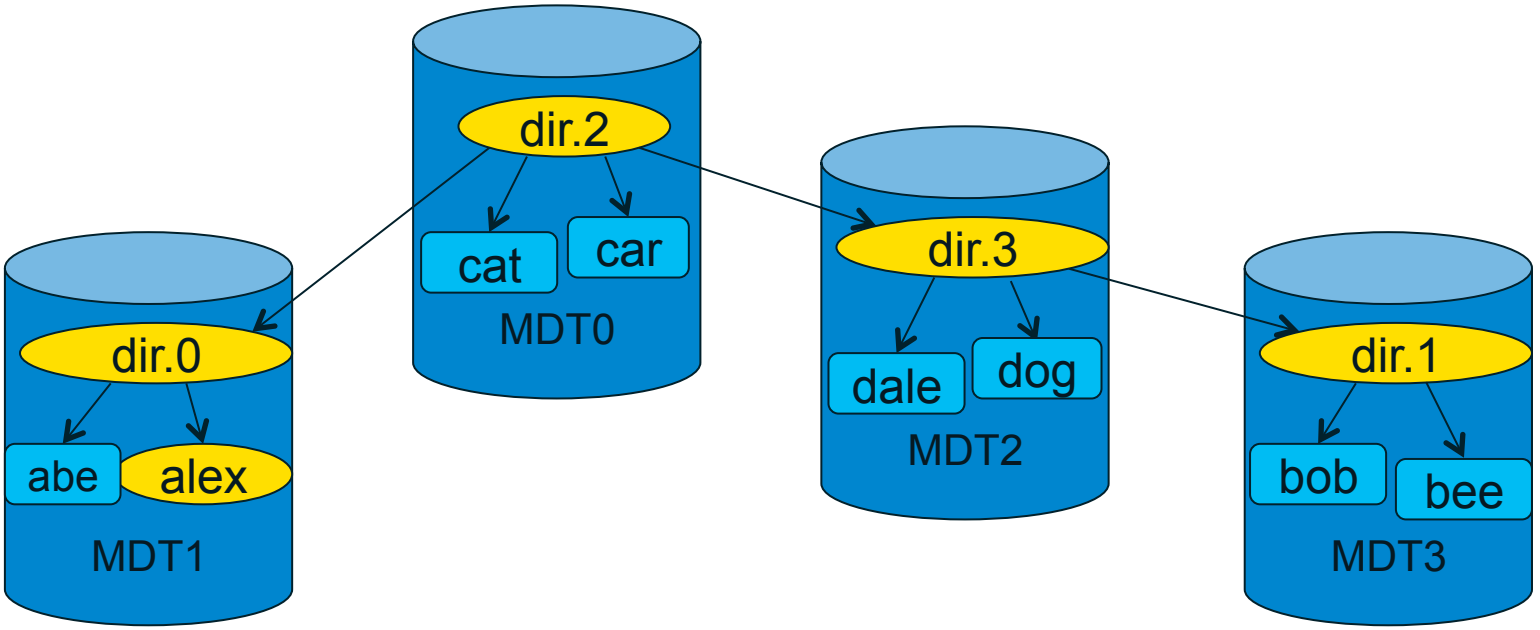


DNE Phase 2 - Shard/Stripe Directory (2.5)

Hash a single directory across multiple MDTs

Multiple servers active for directory/inodes

Improve performance for large directories



LFSCK Phase 1 - OI Scrub (2.3)

Verify and/or rebuild Object Index (OI) file

- OI file maps Lustre object FIDs to local MDT inode numbers
- FID->inode mapping invalided by file-level MDT backup/restore
- Automatically starts if backup/restore detected at MDT startup
- Can verify/repair OI file while filesystem is in use

Iterates over **all** in-use inode objects in MDT filesystem

- Efficient linear reads, readahead from disk
- Verifies FID in inode LMA xattr matches FID->inode OI mapping
- Rate limited to avoid overloading running metadata operations
- Object iterator is building block for later LFSCK features

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

Verify Lustre FID stored in each directory entry

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

Verify inode->parent back-pointer in *link* extended attribute

- 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
- Missing entirely for 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

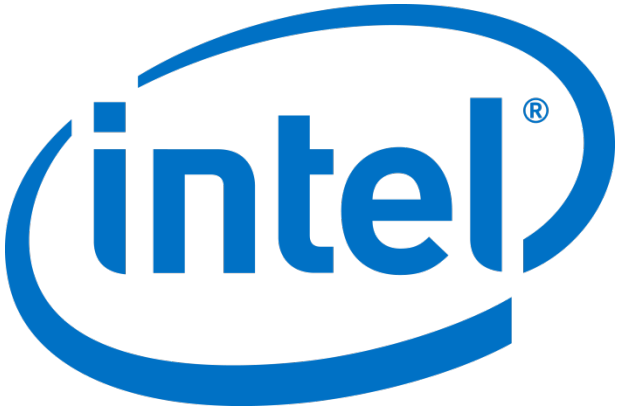
Verifies OST `fid` xattr points back to matching MDT inode

- Allows detecting/creating missing objects

Verifies OST object is referenced by some MDT object

- Allows detecting/deleting orphan objects

Thank You


intel