
whamcloud

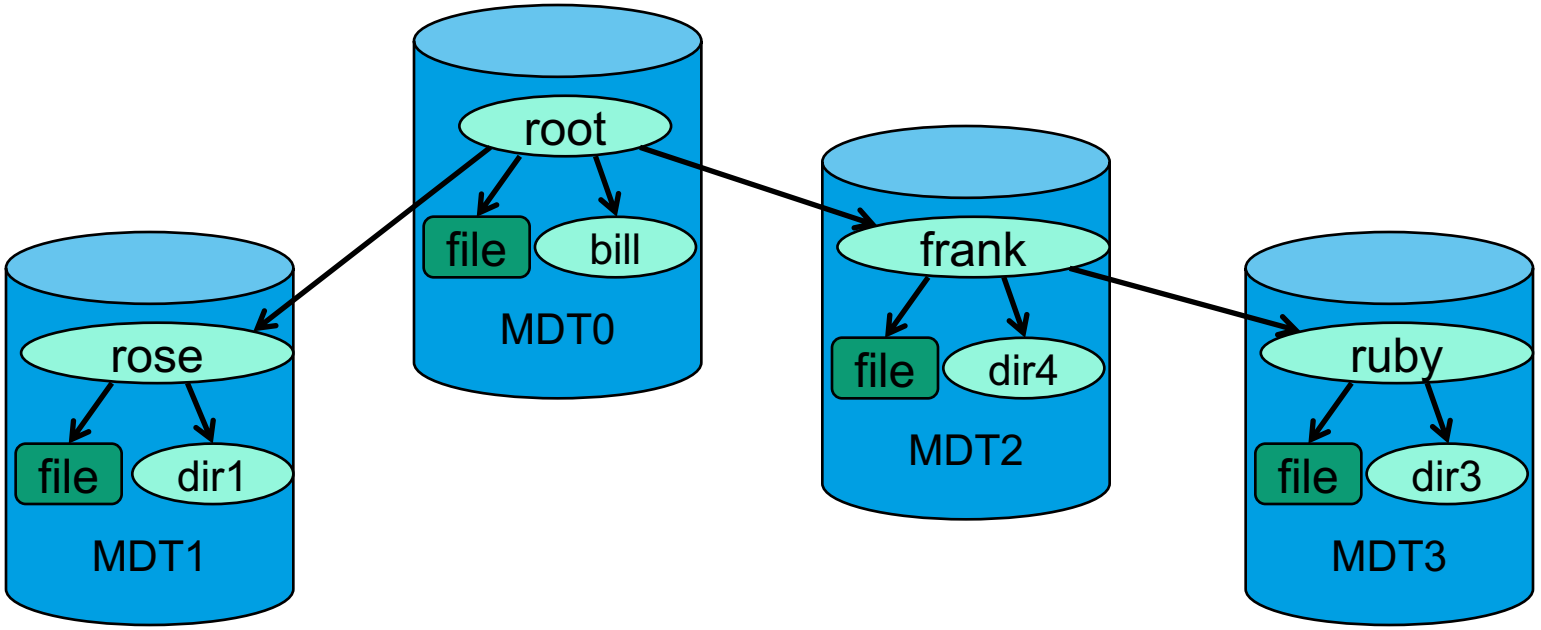
Distributed Namespace Status Phase I - Remote Directories

- Wang Di

  Whamcloud, Inc.

DNE Phase I - Remote Directory

- Subdirectories on a remote metadata target
- Scales MDT namespace, like OSTs can today
- Dedicated performance for users/jobs
- All MDTs can use any/all OSTs to create objects



Remote Directory Implementation

- Remote directory creation by administrator only
  - Remote directory creation is a synchronous disk operation

    lfs mkdir -i {mdtidx} /path/to/remote_dir
- Files/subdirs created in remote dir stay on MDT
  - Local operations (create, unlink, open, close) at maximum performance
  - Limit RPCs that need to communicate with multiple MDTs
  - Simplifies implementation for initial deployment

Remote Directory Limitations

- Failed/disabled MDT affects all of its subtrees
  - Accessing failed/disabled MDT will return EIO
  - Disabling MDT0 causes whole namespace to be inaccessible
- Remote directory can only be created on MDT0
  - Otherwise, failure of one MDT would isolate other MDTs
- Rename or link across MDTs returns –EXDEV
- Deliberate limitation of complexity
  - Limit testing, recovery, failure scenarios for initial deployment
  - Restrictions relaxed as experience is gained, or via override

Enable DNE on new/existing filesystem

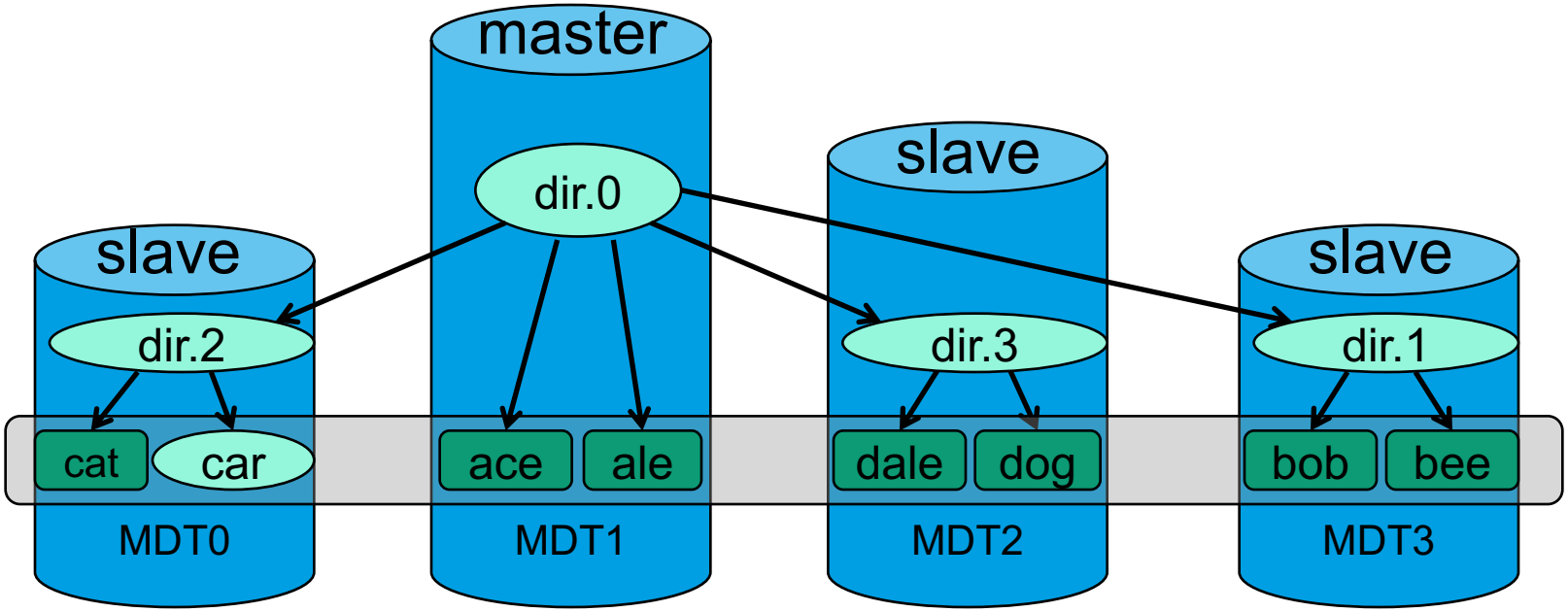
- MDT disk format must use ldiskfs dir_data feature
  - Default for any 2.x formatted filesystem
  - Allows storing remote directory entry pointers
  - Enable on 1.x filesystems: *tune2fs -O dir_data /dev/mdt0*
- Upgrade clients, MGS, MDS, OSS to Lustre 2.4+
  - Not *required* to enable DNE when upgrading to Lustre 2.4+
  - Once DNE is enabled, downgrade to older Lustre difficult
    - requires copying/deleting all files not on MDT0
- Add new MDTs to running filesystem
  - Clients without DNE support evicted at this point
  - New MDTs only used once a remote directory entry is created

mkfs.lustre --reformat --mgsnode={mgsnode} --mdt --index=N /dev/{mdtN}

mount –t lustre /dev/{mdtN} /mnt/{mdtN}

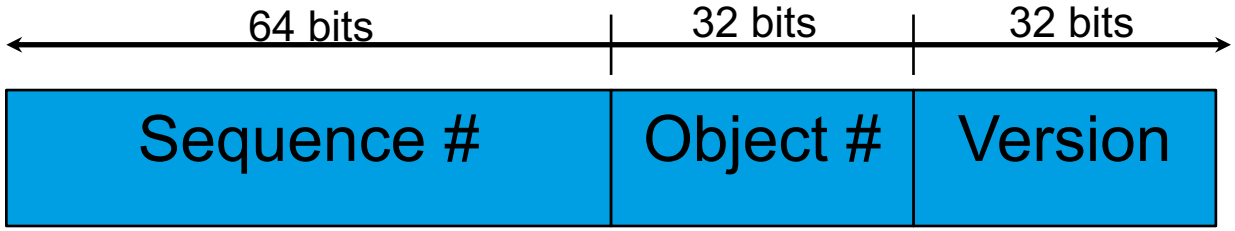
DNE Phase II - Shard/Stripe Directory

- Hash a single directory across multiple MDTs
- Multiple servers active for directory/inodes
- Improve performance for large directories



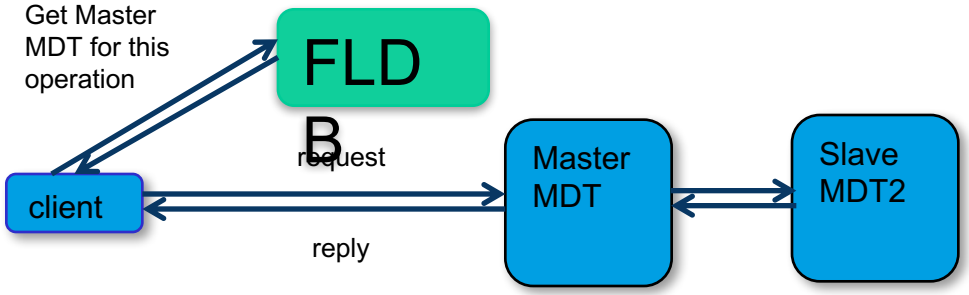
Lustre File IDentifier (FID)

- Unique cluster-wide identifier for file/directory
  - Introduced in Lustre 2.0
  - Three components form object address {f_seq, f_oid, f_ver}
  - Large sequence range is allocated to each server
  - Sequences are large, so FIDs are never re-used
- FID Location Database (FLDB) maps FID->server
  - FLDB is known to all clients and servers
  - Kept small due to few sequence ranges
  - Sequence is looked up in FLDB to find MDT/OST index
- Object Index (OI) maps FID->inode on server
  - OI maps FID to local inode number



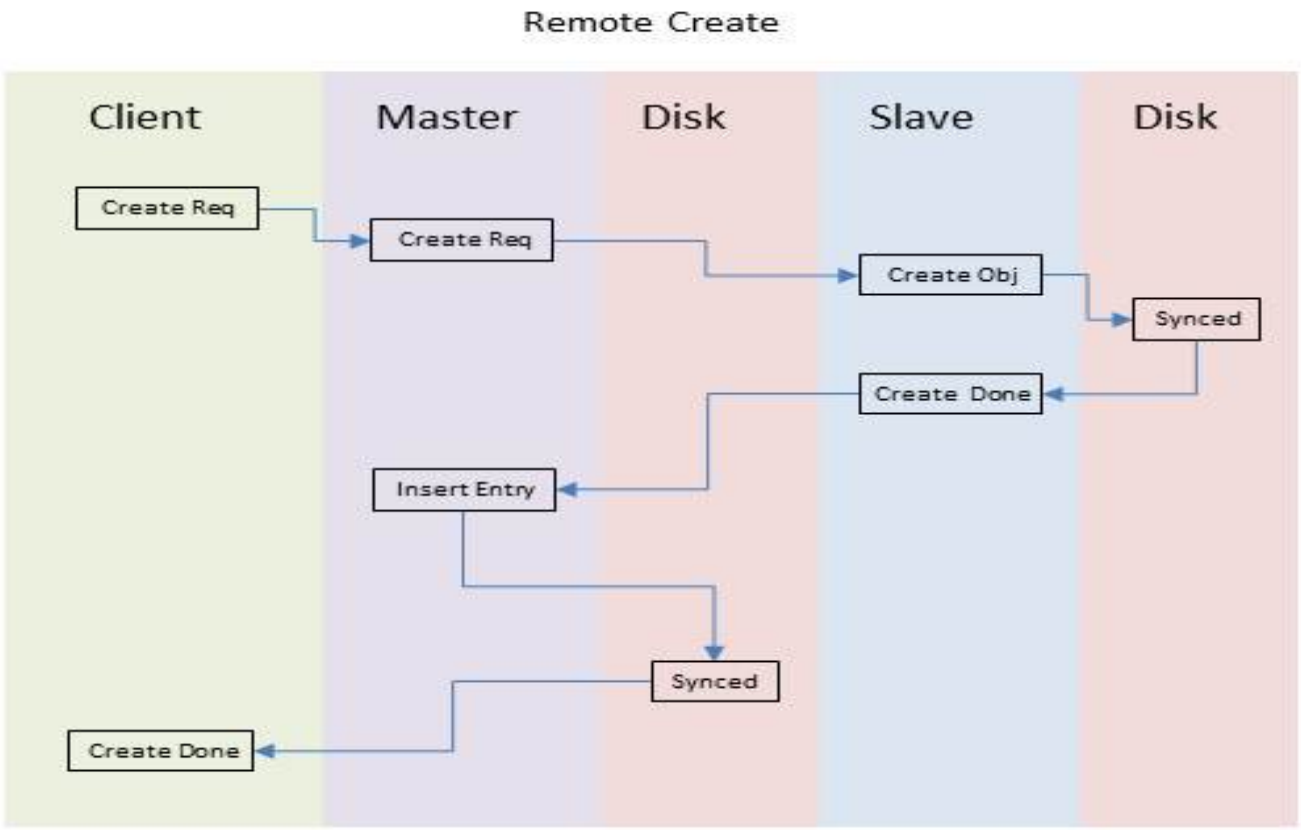
DNE Master and Slave MDTs

- Client does filename lookup in parent directory
  - Root directory lives on MDT0
- Client maps FID to *Master* MDT via FLDB
  - If request only involves one MDT, same as current single MDT
- Some operations need to access *Slave* MDTs
  - Called cross-MDT operations
  - Master MDT forwards update(s) other MDT(s) to finish the request
  - Create/unlink remote directory are only cross-MDT operations today



DNE Operation

- Create Remote Directory

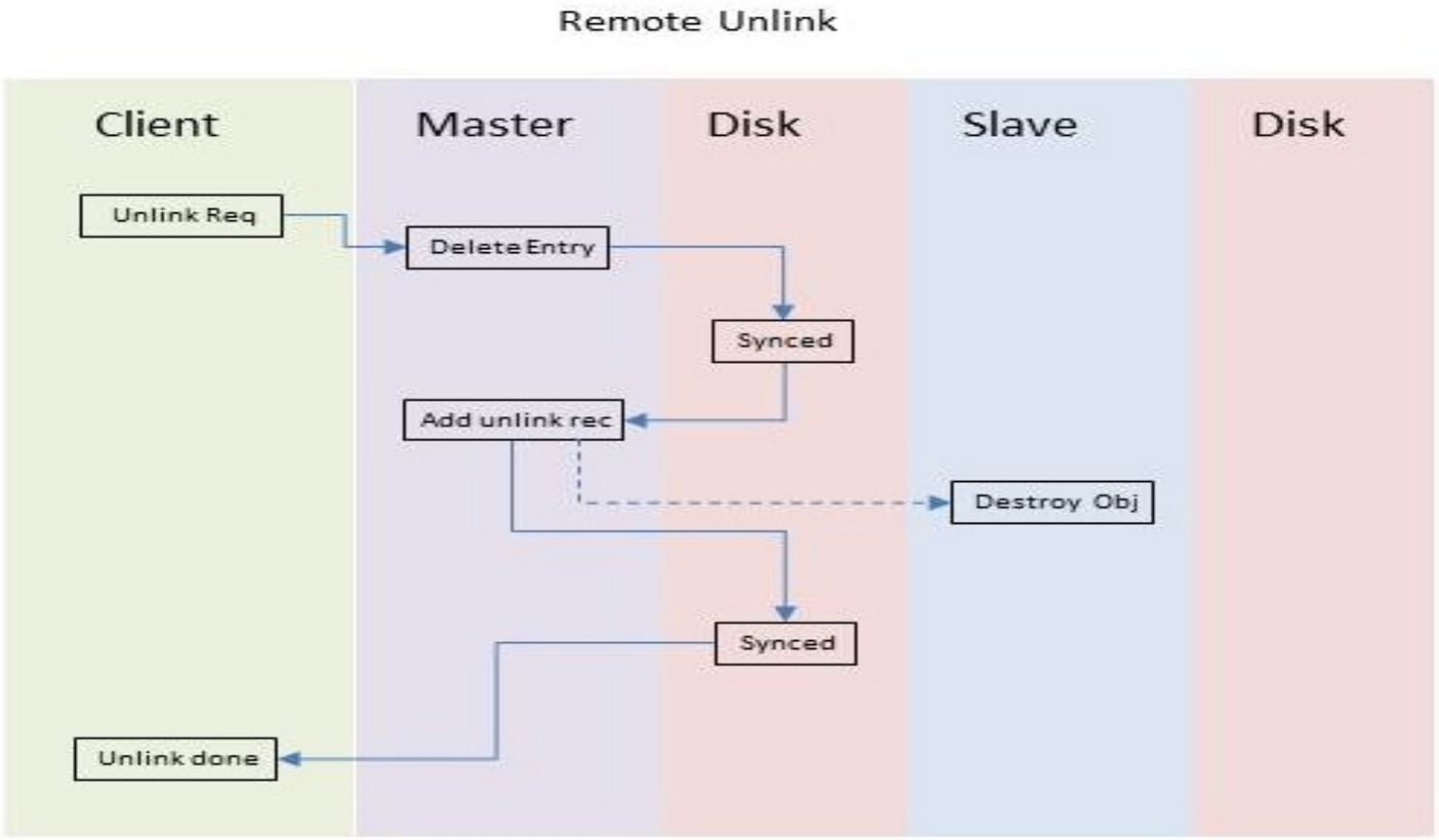


Create Resend between MDTs

- Master MDT checks RPC XID against last_rcvd file
  - Determines whether the operation was committed to disk or not
  - Committed: Master MDT reconstructs RPC reply from last_rcvd entry
  - Uncommitted: Master MDT redoes creation
    - Resend same directory creation RPC to Slave MDT using same FID
- Slave MDT checks if remote directory was created
  - Looks up FID requested by Master in local OI
  - Creates new subdirectory with FID if missing
  - Returns success to Master

DNE Operation

- Unlink Remote Directory

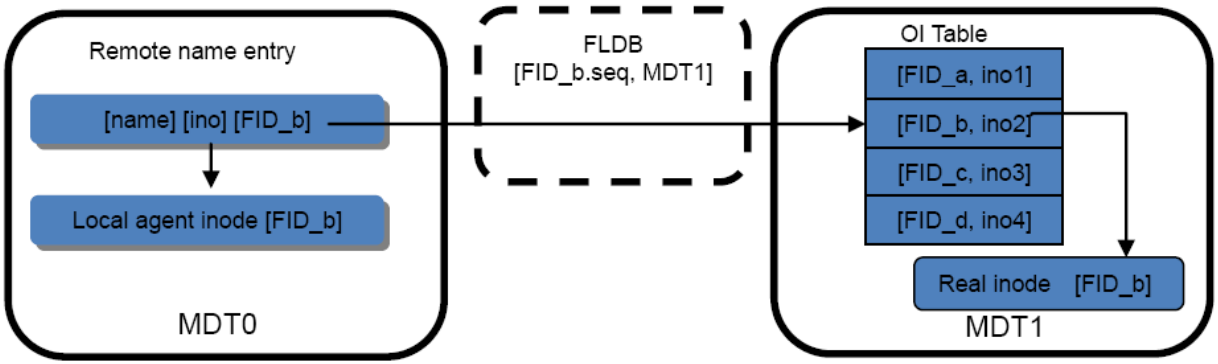


Unlink Resend between MDTs

- Master MDT checks RPC XID against last_rcvd file
  - Determines whether the operation was committed to disk or not
  - Committed: Master MDT reconstructs RPC reply from last_rcvd entry
  - Uncommitted: Master unlinks, deletes name, adds destroy log, etc.
- If Slave MDT fails during this process
  - llog sync thread on Master MDT will resend destroy to Slave MDT
  - Directory unlinks are idempotent, can be retried

Remote Directory Entry

- FID is packed into the name entry
- Each remote entry will have a local agent inode
- Real object (inode) on Remote MDT found via OI



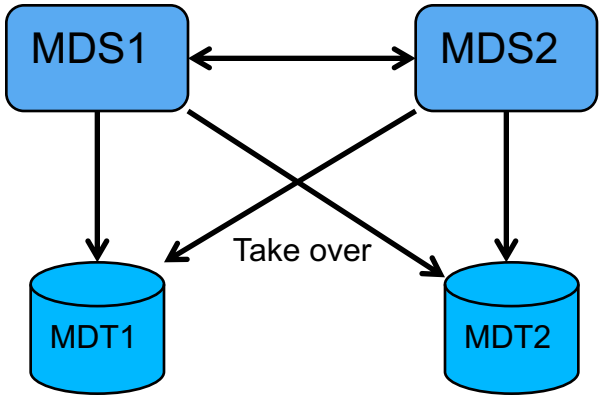
Remote Inode Entry

MDT Disk Layout

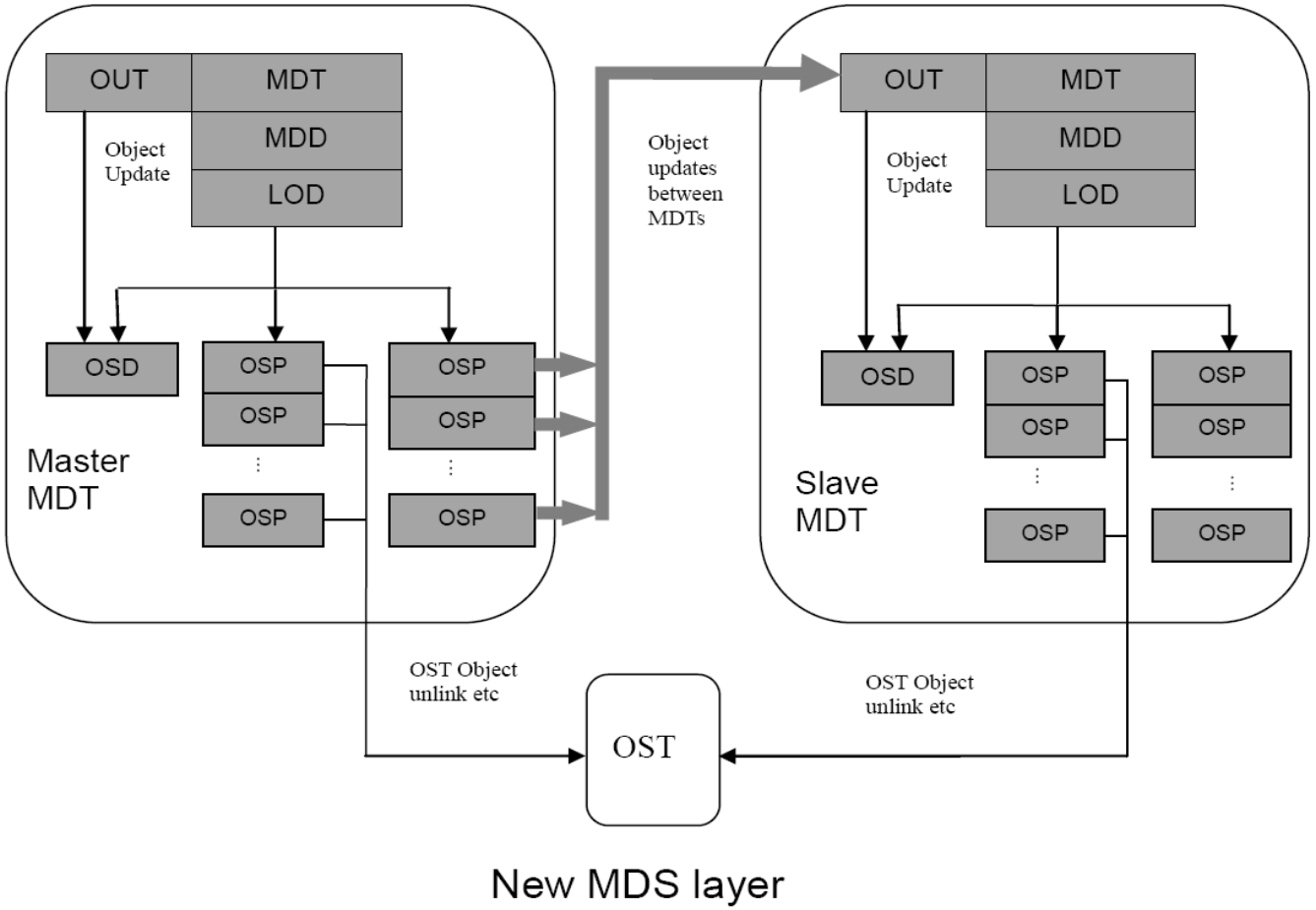
- Two directories (AGENT and REMOTE) added
- AGENT
  - Each remote entry has a local agent inode
  - Agent inodes located under /AGENT/MDTn, one for each remote MDT
- REMOTE
  - Remote directories on Slave MDT created under /REMOTE
- Keeps local disk filesystem consistent
- Allows efficient checking of cross links by LFSCK

DNE High Availability

- Active-Active MDT failover available with DNE
  - Allows multiple MDTs to be exported from one MDS
  - Ensures file system remains available in face of MDS node failure
  - Prevents isolation of large parts of the filesystem

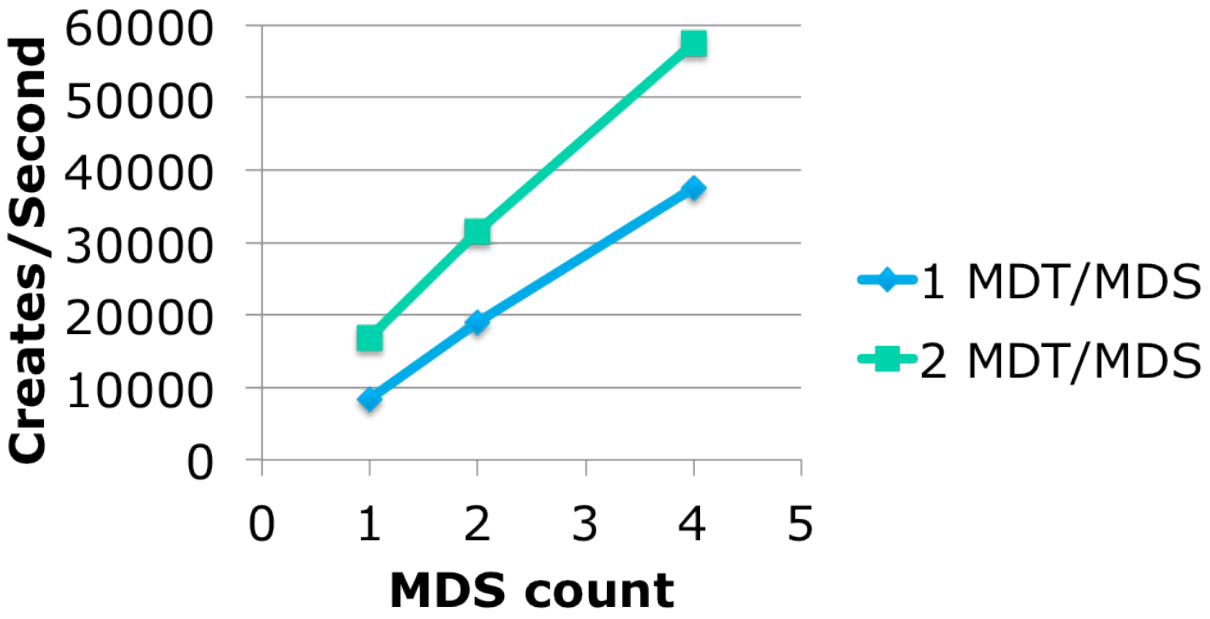


Internal Architecture



Early Test Results

- Testing done on LLNL Hyperion
  - 100 clients, 8 mount points
  - Separate directory per mount point
  - One stripe per file



Thank You

- Wang Di

  Whamcloud, Inc.