Lustre
Clustered Meta-Data (CMD)

Huang Hua
H.Huang@Sun.Com

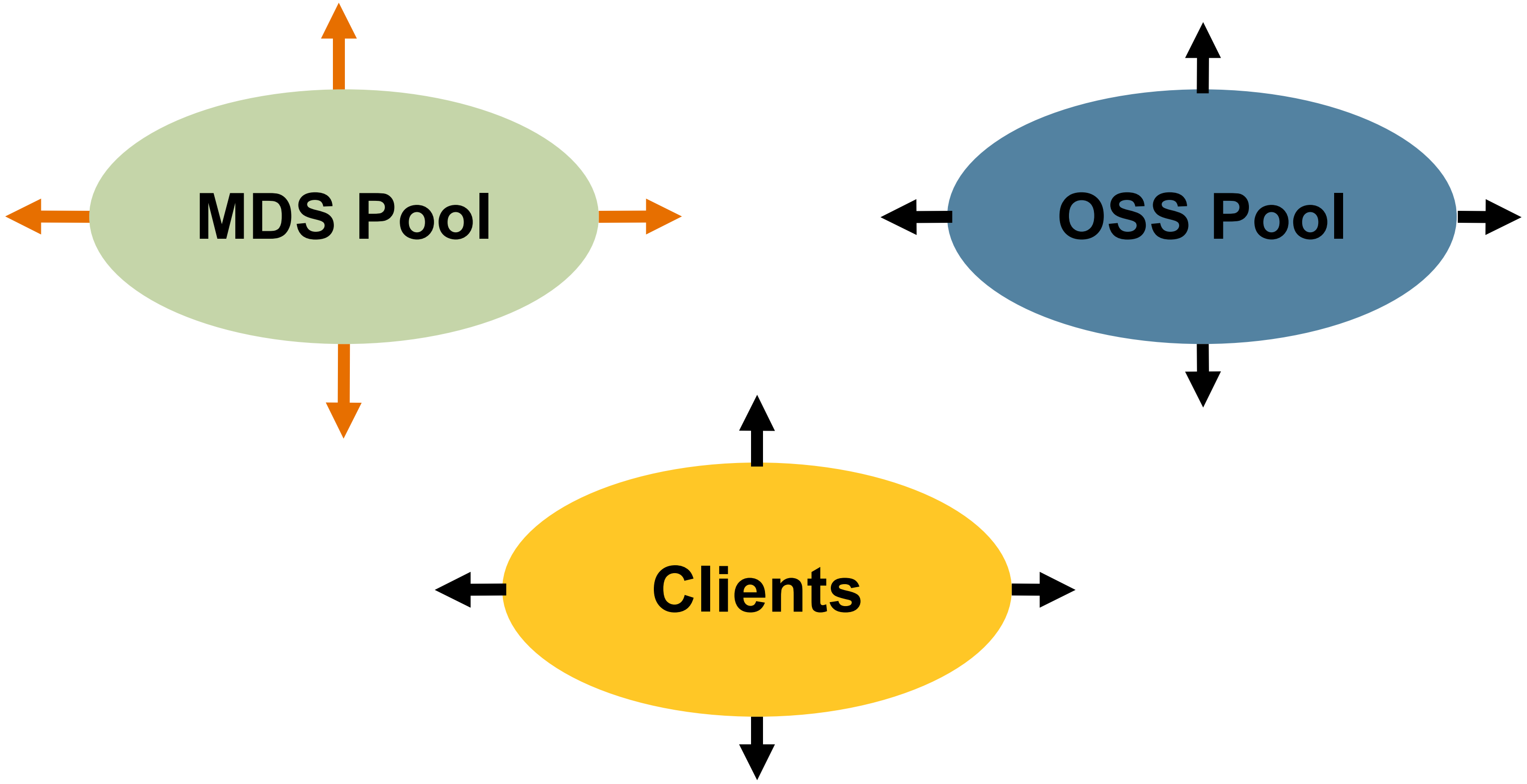
Andreas Dilger
adilger@sun.com

Lustre Group, Sun Microsystems

Agenda

- What is CMD?
- How does it work?
- What are FIDs?
- CMD features
- CMD tricks
- Upcoming development

Lustre Scalability with CMD



Capacity will be 100's of billion of files
Throughput will grow to a million operations per second

What is CMD?

- CMD - Clustered Meta-Data allows allows storing metadata spread over many MDS servers according to some policy
- First version was developed about 4 years ago as a part of Hendrix project
- Working on 3rd version of CMD and it is currently being tested internally

How does it work?

- Cluster has a number of MDS nodes which communicate with each other
- Each MDS has an independent MDT file system for storage
- All clients connect to all MDSes and request root data and volume stats
- All clients do operations (getattr, setattr, unlink) directly with each MDS

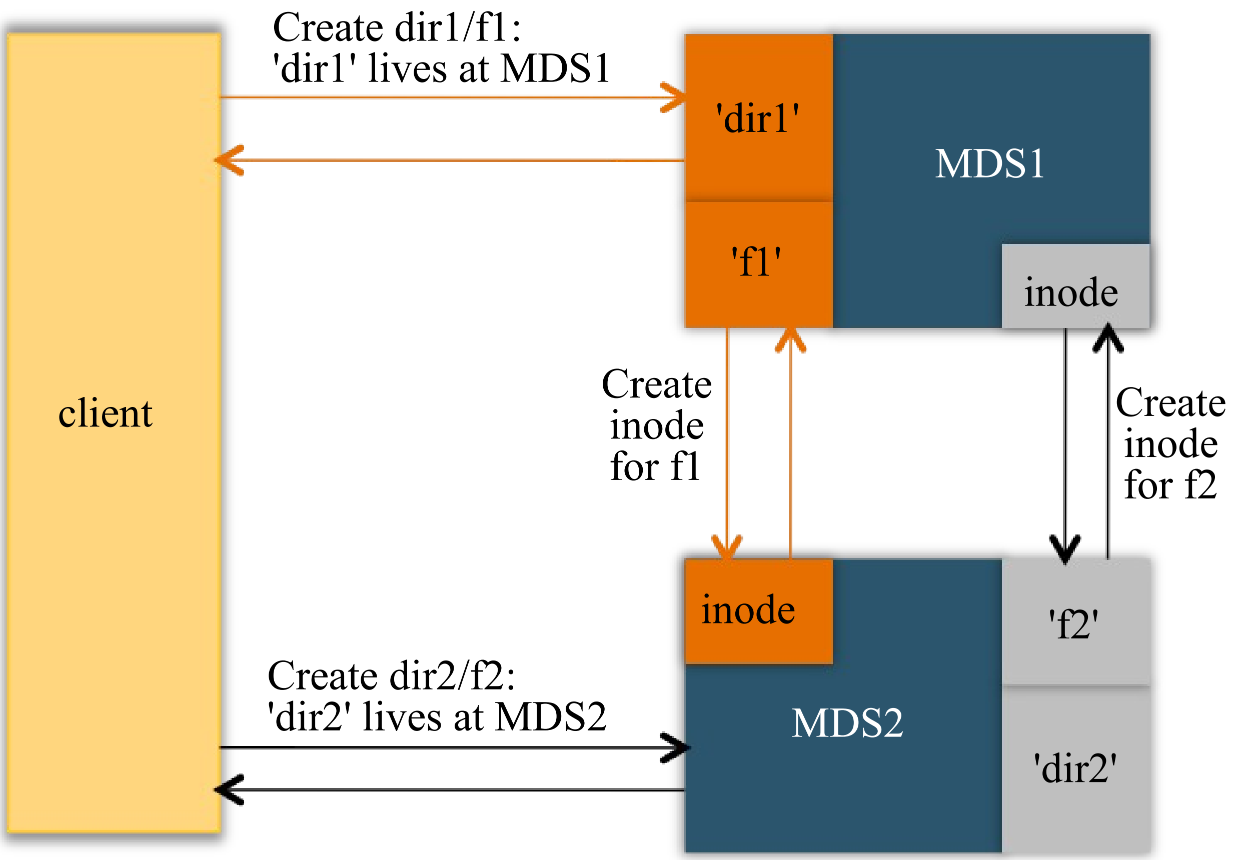
Create a File (local inode)

- Client generates FID for new file and sends create RPC to the MDS which holds the parent directory
- This MDS inserts {filename, FID} into the directory and allocates a local inode
- This happens for all non-directory inodes, so is the common case for file creates
- FID is chosen by client to put inode on the same MDS as the parent directory

Create a file (remote inode)

- Client generates FID for new object and sends create RPC to the MDS which holds the parent directory (call this MDS1)
- MDS1 inserts {filename, FID} into directory and finds out (through FID Location Database) which MDS should hold new file inode (call this MDS2)
- Create RPC sent from MDS1 to MDS2 to create the new inode with passed FID
- This happens for new subdirectories to balance load across MDSes, and in the case of hard-links across MDSes

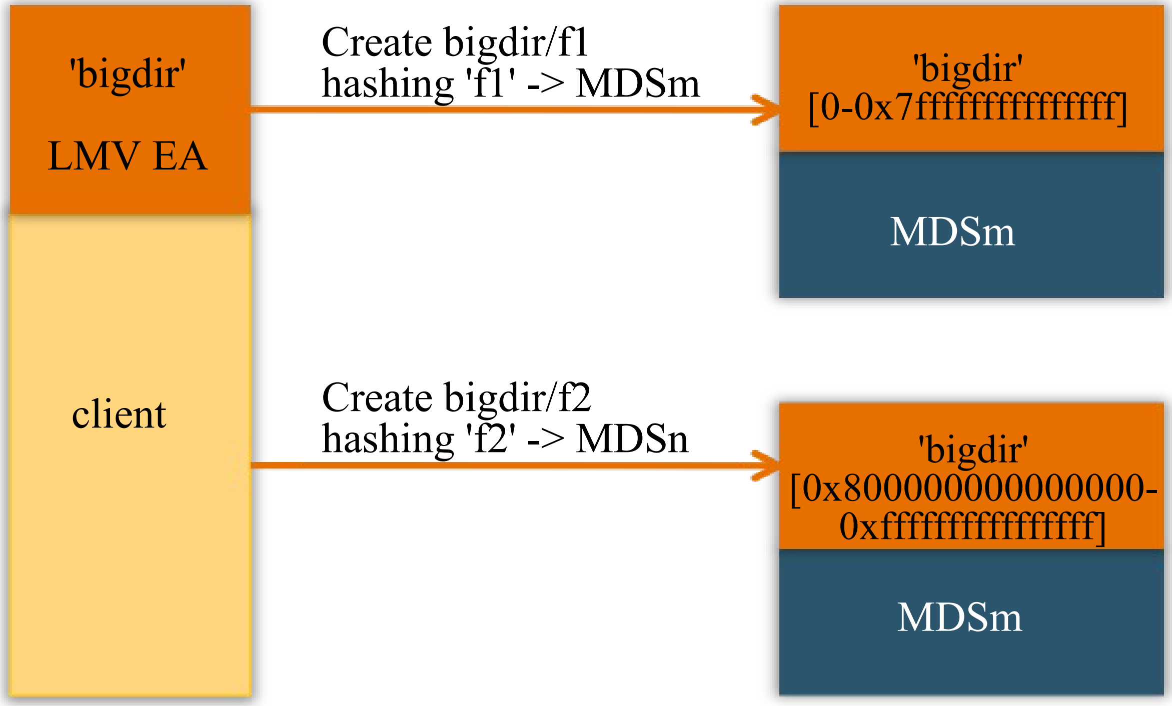
Create on Remote MDS



Directory Split

- When a directory grows too large it gets split over multiple MDSes
- Directory entries get divided into roughly equal chunks and spread over all MDSes in cluster based on hash of filename
- If we create file in split directory, directory 'striping' attribute allows client to decide which MDS holds a given filename for RPCs
- This allows parallel access and more scalability for single directories

2-MDS Split Directory



'bigdir'

LMV EA

client

Create bigdir/f1
hashing 'f1' -> MDSm

Create bigdir/f2
hashing 'f2' -> MDSn

'bigdir'
[0-0x7fffffffffffffff]

MDSm

'bigdir'
[0x8000000000000000-
0xffffffffffffffff]

MDSm

What are FIDs?

- FID (File identifier) is cluster wide 128-bit unique *object* identifier
- FID contains 64-bit sequence number, 32-bit object id, and 32-bit version number
- FID itself does not contain store related information like inode number/generation, or MDS number
- FID also stored in OI (object index) to do FID->inode mapping internally

FID Location Database

- FLD (FID Location Database) records which MDS holds each FID sequence
- All FIDs in one sequence live on the same MDS
- CMD uses FLD to find out which MDS should be contacted to perform an operation on an object
- FLD currently distributed over MDSes via round-robin, or may be replicated on all MDSes (not implemented yet)

CMD Benefits

- Better metadata performance due to parallel access from different clients
- Scale memory, network, and disk IO cost-effectively
- Increase total metadata capacity
- Parallel object creation on OSSes from different MDSes
- Parallelize big directories by storing them on multiple MDSes

Outstanding Issues

- Creating a file with remote inode depends on operations from two MDSes. Wait time is twice as long, and recovery crosses multiple nodes
- Split is complex process with thousands of RPCs and when something fails in the middle of split this can't currently be recovered
- Check for rename of directory into subdirectory is more complex as it needs to check more than one MDS

Current State of CMD

- Still under internal development, though it passed many strict tests
- Users can investigate this feature without any warranty
- The 2.0 release (late 2008) will have much of the CMD functionality
- Will be released as production in 2.2 release (mid 2009)

Upcoming Development

- Cluster wide rollback allows recovery in case of failure of one MDS to undo related transactions on other MDSes. Needed for recovery cases like failure during split and cross-MDS rename
- Replication for FLD, root inode

Huang Hua
H.Huang@Sun.Com

Andreas Dilger
adilger@sun.com