# Lustre Software Notes

# Lustre software release version 2.4 includes support for multiple metadata servers.

**Performance-enhanced ext4 file system:**The Lustre file system uses an improved version of the ext4 journaling file system to store data and metadata. This version, called *ldiskfs* , has been enhanced to improve performance and provide additional functionality needed by the Lustre file system.

With the Lustre software release 2.4 and later, it is also possible to use ZFS as the backing filesystem

for Lustre for the MDT, OST, and MGS storage. This allows Lustre to leverage the scalability and data integrity features of ZFS for individual storage targets.

# Since Lustre software release 2.4, multiple MDTs are supported in the Distributed Namespace Environment (DNE). In addition to the primary MDT that holds the filesystem root, it is possible to add additional MDS nodes, each with their own MDTs, to hold sub-directory trees of the filesystem.

Since Lustre software release 2.8, DNE also allows the filesystem to distribute files of a single directory over multiple MDT nodes. A directory which is distributed across multiple MDTs is known as a *striped directory*.

Lustre software release 2.4 introduces metadata targets for individual sub-directories. Active-active failover configurations are available for MDSs that serve MDTs on shared storage.

Lustre software release 2.4 allows multiple MDTs. By placing two or more MDT partitions on storage shared by two MDSs, one MDS can fail and the remaining MDS can begin serving the unserved MDT. This is described as an active/active failover pair.

The Lustre software provides failover functionality only at the file system level. In a complete failover solution, failover functionality for system-level components, such as node failure detection or power control, must be provided by a third-party tool.

OST failover functionality does not protect against corruption caused by a disk failure. If the storage media (i.e., physical disk) used for an OST fails, it cannot be recovered by functionality provided in the Lustre software. We strongly recommend that some form of RAID be used for OSTs. Lustre functionality assumes that the storage is reliable, so it adds no extra reliability features.

Some sample scripts are included in the directory where the Lustre software is installed. If you have installed the Lustre source code, the scripts are located in the lustre/tests sub- directory. These scripts enable quick setup of some simple standard Lustre configurations.

If multiple MDTs are going to be present in the system, each MDT should be specified for the anticipated usage and load. For details on how to add additional MDTs to the filesystem, see [Section 14.6, “Adding](#_bookmark314) [a New MDT to a Lustre File System”](#_bookmark314).

MDT0 contains the root of the Lustre file system. If MDT0 is unavailable for any reason, the file system cannot be used.

Using the DNE feature it is possible to dedicate additional MDTs to sub-directories off the file system root directory stored on MDT0, or arbitrarily for lower-level subdirectories. using the lfs mkdir -i *mdt\_index* command. If an MDT serving a subdirectory becomes unavailable, any subdirectories on that MDT and all directories beneath it will also become inaccessible. Configuring multiple levels of MDTs is an experimental feature for the 2.4 release, and is fully functional in the 2.8 release. This is typically useful for top-level directories to assign different users or projects to separate MDTs, or to distribute other large working sets of files to multiple MDTs.

Starting in the 2.8 release it is possible to spread a single large directory across multiple MDTs using the DNE striped directory feature by specifying multiple stripes (or shards) at creation time using the lfs mkdir -c *stripe\_count* command, where *stripe\_count* is often the number of MDTs in the filesystem. Striped directories should typically not be used for all directories in the filesystem, since this incurs extra overhead compared to non-striped directories, but is useful for larger directories (over 50k entries) where many output files are being created at one time.

With a ZFS backing filesystem for the MDT or OST, the space allocation for inodes and file data is dynamic, and inodes are allocated as needed. A minimum of 4kB of usable space (before mirroring) is needed for each inode, exclusive of other overhead such as directories, internal log files, extended attributes, ACLs, etc. ZFS also reserves approximately 3% of the total storage space for internal and redundant metadata, which is not usable by Lustre. Since the size of extended attributes and ACLs is highly dependent on kernel versions and site-specific policies, it is best to over-estimate the amount of space needed for the desired number of inodes, and any excess space will be utilized to store more inodes.

Note that the number of total and free inodes reported by lfs df -i for ZFS MDTs and OSTs is estimated based on the current average space used per inode. When a ZFS filesystem is first formatted, this free inode estimate will be very conservative (low) due to the high ratio of directories to regular files created for internal Lustre metadata storage, but this estimate will improve as more files are created by regular users and the average file size will better reflect actual site usage.

Starting in release 2.4, using the DNE remote directory feature it is possible to increase the total number of inodes of a Lustre filesystem, as well as increasing the aggregate metadata performance, by configuring additional MDTs into the filesystem, see [Section 14.6, “Adding a](#_bookmark314) [New MDT to a Lustre File System”](#_bookmark314) for details.

With Lustre software version 2.8, a new tunable is available to allow users with a specific group ID to create and delete remote and striped directories. This tunable is enable\_remote\_dir\_gid. For example, setting this parameter to the 'wheel' or 'admin' group ID allows users with that GID to create and delete remote and striped directories. Setting this parameter to -1 on MDT0 to permanently allow any non-root users create and delete remote and striped directories. On the MGS execute the following command:

mgs# lctl conf\_param *fsname*.mdt.enable\_remote\_dir\_gid=-1

For the Lustre filesystem 'scratch', the commands expands to:

mgs# lctl conf\_param scratch.mdt.enable\_remote\_dir\_gid=-1

. The change can be verified by executing the following command on every MDS:

mds# lctl get\_param mdt.*\**.enable\_remote\_dir\_gid

If you need to change the NID on the MDT or OST, a new replace\_nids command was added in Lustre software release 2.4 to simplify this process. The replace\_nids command differs from tunefs.lustre --writeconf in that it does not erase the entire configuration log, precluding the need the need to execute the writeconf command on all servers and re-specify all permanent parameter settings. However, the writeconf command can still be used if desired.

To generate a list of all files with more than 160 stripes use lfs find with the --stripe- count option:

lfs find ${mountpoint} --stripe-count=+160

In Lustre software release 2.4, a new feature allows using multiple MDTs, which can each serve one or more remote sub-directories in the file system. The root directory is always located on MDT0.

Note that clients running a release prior to the Lustre software release 2.4 can only see the namespace hosted by MDT0 and will return an IO error if an attempt is made to access a directory on another MDT.

This step only enables FID-in-dirent for newly created files. If you are upgrading to Lustre software release 2.4, you can use namespace LFSCK to enable FID-in-dirent for the existing files. For the case of upgrading from Lustre software release 1.8, it is important to note that if you do NOT enable dirdata via the tune2fs command above, the namespace LFSCK will NOT generate FID-in-dirent for the existing files. For more information about FID-in-dirent and related functionalities in LFSCK, see [Section 1.3, “ Lustre File System Storage and I/O”](#_bookmark30).

If the file system was used between the time the backup was made and when it was restored, then the online LFSCK tool (part of Lustre code after version 2.3) will automatically be run to ensure the file system is coherent. If all of the device file systems were backed up at the same time after the entire Lustre file system was stopped, this step is unnecessary. In either case, the file system will be immediately although there may be I/O errors reading from files that are present on the MDT but not the OSTs, and files that were created after the MDT backup will not be accessible or visible. See [Section 32.4, “ Checking the file](#_bookmark805) [system with LFSCK”](#_bookmark805)for details on using LFSCK.

With Lustre software version 2.6 and later, there is no longer a need to run ll\_recover\_lost\_found\_objs on the OSTs, since the LFSCK scanning will automatically move objects from lost+found back into its correct location on the OST after directory corruption.

In Lustre software release 2.0 through 2.2, the only successful way to backup and restore an MDT is to do a device-level backup as is described in this section. File-level restore of an MDT is not possible before Lustre software release 2.3, as the Object Index (OI) file cannot be rebuilt after restore without the OI Scrub functionality. **Since Lustre software release 2.3**, [Object Index files are automatically rebuilt at first mount after a restore is detected (see LU-957](http://jira.hpdd.intel.com/browse/LU-957))

Lustre software release 2.4 can be configured with multiple MDTs in the same file system. Each sub- directory can have a different MDT. To identify on which MDT a given subdirectory is located, pass the getstripe [--mdt-index|-M] parameters to lfs. An example of this command is provided in [the section Section 14.8.1, “Removing a MDT from the File System”](#_bookmark321).

The reserved space for each OST can be adjusted by the user. Use the lctl set\_param command, for example the next command reserve 1GB space for all OSTs.

lctl set\_param -P osp.\*.reserved\_mb\_low=1024

Lustre software version 2.8 includes a feature to migrate metadata (directories and inodes therein) between MDTs. This migration can only be performed on whole directories. For example, to migrate the contents of the /testfs/testremote directory from the MDT it currently resides on to MDT0000, the sequence of commands is as follows:

Lustre filesystems formatted with a Lustre release prior to 2.10 can be still safely upgraded to release 2.10, but will not have project quota usage reporting functional until tune2fs - O project is run against all ldiskfs backend targets. This command sets the PROJECT feature flag in the superblock and runs e2fsck (as a result, the target must be offline). See [Section 22.5, “ Quotas and Version Interoperability”](#_bookmark522) for further important considerations.