Lustre™ 2.0 Release Notes

Part No. 821-2077-10
July 2010, Revision 01

Please
Recycle

Lustre 2.0 Release Notes

The *Lustre 2.0 Release Notes* describe the testing environments, new features and enhancements, major fixes and known issues for Lustre 2.0.

Tested Environments

Lustre 2.0 has been tested against the following Linux platforms, architectures[1] and interconnects. Downloadable Lustre packages (RPMs) are available for the Linux platforms listed below.

	Linux Platform*	Architecture	Interconnect
Server	OEL 5.4 RHEL 5.4	x86_64	
Client	OEL 5.4 RHEL 5 SLES 10, 11 Scientific Linux 5 [New] Fedora 12 (2.6.31) [New]	x86_64 ia64 (RHEL) ppc64 (SLES) i686	
Server and Client			TCP/IP OFED

* Lustre does not support security-enhanced (SE) Linux (including clients and servers).

1. We encourage the use of 64-bit platforms.

New Features

Lustre 2.0 introduces the following new features and enhancements.

Lustre 2 Architecture

Lustre 2.0 introduces a new architecture for the metadata server (MDS) and clients. This architecture, known as Lustre 2, establishes a stable foundation for platform portability and major performance optimizations in future Lustre releases.

Changelogs

Changelogs record events that change the file system namespace or file metadata. Events such as file creation, deletion, renaming, attribute changes, etc. are recorded with the target and parent file identifiers (FIDs), the name of the target, and a timestamp. These records can be used for a variety of purposes:

- Record recent changes to feed into an archiving system.
- Use changelog entries to exactly replicate changes in a file system mirror.
- Set up "watch scripts" that take action on certain events or directories. Changelog record are persistent (on disk) until explicitly cleared by the user. The are guaranteed to accurately reflect on-disk changes in the event of a server failure.
- Maintain a rough audit trail (file/directory changes with timestamps, but no user information).

Commit on Share

The Commit on Share (COS) feature prevents missing clients from causing the evictions of other clients when Lustre is in recovery mode. If some clients miss the recovery window, the remaining clients are not evicted.

When an MDS starts up and enters recovery mode after a failover or service restart, clients begin to reconnect and replay their uncommitted transactions. If one or more clients miss the recovery window, this may cause other clients to abort their transactions or be evicted. The transactions of evicted clients cannot be applied and are aborted. This causes a cascade effect as transactions dependent on the aborted ones fail and so on. COS addresses this problem by eliminating dependent transactions. With no dependent, uncommitted transactions to apply, the clients replay their requests independently without the risk of being evicted.

COS is controlled with the MDT's `commit_on_sharing` parameter, which can be set to 1 (enabled) or 0 (disabled). In Lustre 2.0.0, COS is disabled, by default.

To enable COS, set `commit_on_sharing` to 1 on the MDS node:

```
# lctl set_param mdt.<mdt-name>.commit_on_sharing=1
```

To disable COS, set `commit_on_sharing` to 0 on the MDS node:

```
# lctl set_param mdt.<mdt-name>.commit_on_sharing=0
```

To store the `commit_on_sharing` parameter on disk, use `mkfs.lustre` or `tunefs.lustre` with `--param mdt.commit_on_sharing=0/1`.

Lustre_rsync

The lustre_rsync feature provides namespace and data replication to an external (remote) backup system without having to scan the file system for inode changes and modification times. Lustre metadata changelogs are used to record file system changes and determine which directory and file operations to execute on the replicated system. Lustre_rsync avoids full file system scans, which can be time-consuming on very large file systems. Lustre_rsync can be restarted from where it left off, so the replicated file system is fully synchronized when operation completes. Lustre_rsync may be bi-directional for distinct directories.

The replicated system may be another Lustre system or any other file system. The replica is an exact copy of the namespace of the original file system at a specific time. However, the replicated file system is not a snapshot of the source file system; its contents may differ from the original file system's contents. On the replicated file system, each file contains the data present when the file transfer occurred.

Size-on-MDS (Preview)

Note – In Lustre 2.0, size-on-MDS (SOM) is available as a preview feature. SOM is still under development, so it should not be enabled on a 2.0 production system.

The size-on-MDS feature caches file object attributes (file size, number of blocks, ctime and mtime) on the MDS; these attributes were originally stored on the OSTs. Storing the SOM cache on the MDS allows clients to issue just one RPC per file to the MDS and avoid sending multiple RPCs to the OSTs to find the file object attributes kept on those OSTs. The SOM enhancement significantly improves the performance of the directory listing command (`ls -l`).

Size-on-MDS can be enabled when the file system is created, using `mkfs.lustre` or set later when the file system is running, using `lctl`. The SOM parameter is `$FSNAME.mdt.som=$MODE`, where `$FSNAME` is the file system name ("`lustre`" by default) and `$MODE` is "`enabled`" or "`disabled`".

Note – Once SOM is enabled on the MDS, clients can only mount the file system if the `som_preview` mount option is specified. The `som_preview` option must be given on the client, not the MDS.

To create a file system with SOM enabled, run:

```
$ mkfs.lustre --mdt --param=mdt.som=enabled
```

To enable/disable SOM on an existing file system, run:

```
$ lctl conf_param $FSNAME.mdt.som=$MODE
```

SOM mode is applied after the file system is shut down, not on the fly.

For more information on SOM, see BZ 22864.

Landings to Lustre Master for 2.0

For a list of all bugs that were resolved and landed to Lustre Master (HEAD) and no earlier Lustre branches, see Bugs Landed on HEAD. For a list of all bugs fixed for this release, see the 2.0 changelog.

Known Issues and Workarounds

This section describes known issues with Lustre 2.0, which will be resolved in future releases.

- **BZ 16893 (Enabling ext4 by default)**

  Enabling ext 4 allows LUNs larger than 8 TB to be used in the Lustre file system. When ext4 is enabled, by default, in a system at scale, servers become overloaded (cause unknown). This results in clients timing out and attempting to reconnect, an action which the server does not accept. Eventually, the server evicts the client due to a lock timeout.

  Workaround: Do not enable ext4 in Lustre 2.0.0.

- **BZ 16919 (Asynchronous journal commit functionality)**

  Lustre asynchronous journal commit adds support for Lustre clients to submit write requests to an OST, but not require the server to do the I/O synchronously. Instead, the client keeps a copy of the data in cache until it receives a commit notification from the OST and rewrites the data if the OST crashes. This allows a single client to submit a large number of writes without having to commit the journal transaction. This enhancement was completed after the Lustre 2.0 code freeze.

  Workaround: If asynchronous journal commits are needed, use Lustre 1.8.0 or later (the latest 1.8 version is recommended).

- **BZ 16774 (Avoid replaying unused locks during recovery)**

  During recovery, Lustre clients can overwhelm the servers by replaying thousands of unused locks. This bug was fixed after the Lustre 2.0 code freeze. It will be available in a later 2.0.x release.

  Workaround: If this bug fix is needed, use Lustre 1.8.2 or later.

- **BZ 22040 (Parallel_scale_nfsv4 test times out after running 13000 seconds)**

  Using flock on an NFSv4 export of a Lustre file system can cause applications to hang. The root issue (tracked in **BZ 14080**) will be fixed after Lustre 2.0.0 is released.

  Workaround: We recommend not using flock on NFSv4 until BZ 14080 is fixed.

Additional Documentation

The *Lustre 2.0 Operations Manual* is a comprehensive resource that describes how to install, configure, and tune Lustre 2.0. The manual also contains troubleshooting and utilities information, and tips to improve Lustre operations and performance. For the latest version of the Lustre 2.0 manual, see:

http://wiki.lustre.org/manual/LustreManual20_HTML/index.html

The *Lustre 2.0 Changelog* describes networks and kernels that were tested with this Lustre version and provides a comprehensive list of fixed bugs. For the latest version of the Lustre 2.0 changelog, see:

http://wiki.lustre.org/index.php/Change_Log_2.0