

# Cray Lustre File System Monitoring

esfsmon

Jeff Keopp  
OSIO/ES Systems  
Cray Inc.  
St. Paul, MN USA  
keopp@cray.com

Harold Longley  
OSIO  
Cray Inc.  
St. Paul, MN USA  
htg@cray.com

**Abstract—** The Cray Data Management Platform (formerly Cray External Services Systems) provides two external Lustre File System products – CLFS and Sonexion. The CLFS Lustre Monitor (esfsmon) keeps the CLFS file systems available by providing automated failover of Lustre assets in the event of MDS and/or OSS node failures. The Lustre Monitoring Tool (LMT) is now part of the standard ESF software release used by CLFS systems.

This paper provides an overview of the latest version of esfsmon which supports Lustre Distributed NamespacE (DNE). Topics include configuration, and Lustre failover and failback operations – both manual and automated. The Lustre Monitor Tool (LMT) will also be discussed.

*CLFS; esfsmon; Lustre Failover; Lustre DNE; LMT*

## I. INTRODUCTION

The Cray Data Management Platform offers two external Lustre Filesystem products – Lustre Filesystem by Cray (CLFS) and Cray Sonexion.

CLFS is managed and provisioned by the Cray Integrated Management Server (CIMS). Beginning with the ESM-XX-3.0.0 CIMS software release and ESF-XX-2.2.2 CLFS software release, Lustre Distributed NamespacE (DNE) configurations are supported.

Cray Sonexion is managed by the Cray Sonexion System Manager (CSSM) in conjunction with the Unified System Management firmware (USM).

Automated failover of Lustre targets on CLFS systems is provided by esfsmon. Esfsmon is a CLFS health monitor running on the CIMS. Failback of CLFS Lustre targets to their primary host is performed by the esfsmon\_failback command.

The Lustre Monitor Tool (LMT) is also available in the ESF software release. LMT provides real-time and historical data from /proc/fs/lustre as it pertains to the Lustre filesystem.

This paper discusses the operation and configuration of esfsmon, esfsmon\_failback and lustre\_control. The lustre\_control utility is used by esfsmon and esfsmon\_failback for managing the failover and failback of Lustre targets. It is also used for starting and stopping the

CLFS filesystem. Cray Sonexion monitoring is not discussed here.

## II. LUSTRE FILESYSTEM BY CRAY (CLFS)

The Lustre Filesystem by Cray (CLFS, formerly known as esFS) is a highly flexible external storage solution. Customers have flexibility in storage choices while utilizing Cray's best practices for Lustre. Being external to the Cray Supercomputer, CLFS is available regardless of the state of the Cray Supercomputer.

CLFS consists of Lustre MetaData Servers (MDS) and Lustre Object Storage Servers (OSS) in failover pairs along with their associated block storage devices, Ethernet and Infiniband networks, and management services. The CLFS system is managed and provisioned by the Cray Integrated Management Server (CIMS, formerly known as esMS).

The OSS servers are configured in active/active failover pairs while MDS servers are in an active/passive configuration. If configured for Lustre DNE, the MDS servers are in active/active failover pairs.

The CIMS employs Bright Cluster Manager software to monitor and provision the MDS and OSS nodes. Esfsmon is implemented as a health check in the Bright Cluster Manager monitoring framework. This allows esfsmon to leverage the Bright Cluster Manager infrastructure. The CIMS also monitors the block storage controllers and the Ethernet and Infiniband switches associated with the CLFS system. Multiple CLFS systems as well as Cray Development and Login Servers (CDL, formerly known as esLogin) may be managed and provisioned by the CIMS. CIMS may be a single server or a high-availability (HA) pair. HA CIMS are in an active/passive configuration.

## III. LUSTRE MANAGEMENT

The Lustre filesystems are controlled from the CIMS by the lustre\_control utility. This utility allows easy operation of the Lustre filesystems. It is also used by esfsmon and esfsmon\_failback to perform the failover and failback of Lustre targets. Components of lustre\_control reside on the CIMS and the CLFS nodes. The common lustre\_control commands are described in Table 1.

TABLE 1. LUSTRE CONTROL COMMANDS

Command	Description
install	Install filesystem definitions.
remove	Remove filesystem definitions.
start	Start services on Lustre servers.
stop	Stop services on Lustre servers.
reformat	Format Lustre devices.
write_conf	Regenerate Lustre configuration logs.
failover	Failover services from their specified primary server(s) to their respective backup server(s).
failback	Failback services onto their specified primary server(s) from their respective backup server(s).
status	Reports the status of Lustre services.
verify_config	Report differences between expected e2label and actual e2label for Lustre devices.
dump_csv_config	Print to stdout the file system configuration in a comma-separated value format.
help	Display detailed usage information for a particular command.
set_tune	Set Lustre tunable parameters.

### A. Defining Lustre Filesystems for `lustre_control`

Lustre filesystems are defined in `fs_defs` files. These files are installed to `lustre_control` using the `lustre_control install` command. For example, if the `scratch.fs_defs` file defines the Lustre filesystem named “scratch”, the following command would be used to install it for use by `lustre_control`:

```
# lustre_control install scratch.fs_defs
```

The `fs_defs` file is not used directly in normal operation. If changes are made to the `fs_defs` file, it must be installed again to `lustre_control` before the changes are effective.

Each Lustre filesystem must have its own `fs_defs` file. Table 2 describes the parameters used in `fs_defs` files for CLFS. An example `fs_defs` file is available on the CIMS at the following location.

```
# /opt/cray/esms/cray-lustre-control-XX/default/etc/example.fs_defs
```

TABLE 2. LUSTRE FILESYSTEM DEFINITIONS

Parameter	Description
fs_name	Filesystem name. Must be 8 characters or less.
nid_map	Lustre server hosts to LNET NIDs mapping.
<b>Lustre Targets</b>	
mgt	Management Target.
mdt	Meta Data Target
ost	Object Storage Target
<b>Lustre Targets have the following sub-parameters</b>	
node	Primary device host.
dev	Device path.
fo_node	Backup device host.
fo_dev	Backup device path. Only needed if different from the primary device path.
jdev	External journal device. (OST configuration only)
index	Force a particular OST or MDT index.
<b>Target Mount Paths</b>	
mgt_mount_path	MGT mount path
mdt_mount_path	MDT mount path
ost_mount_path	OST mount path
<b>Optional Definitions</b>	
routers	Service nodes that will be Lustre routers.
auto_fo	Automatic failover (enables monitoring via <code>esfsmon</code> )
imp_rec	Imperative Recovery. Not applicable to CLFS
stripe_size	Stripe size. Default: 1048576
stripe_count	Stripe count. Default: 1
journal_size	Journal size in megabytes. Default: 400
journal_block_size	Block size for the journal device in bytes. Default: 4096
timeout	Lustre timeout. Default: 300 seconds
back_fs_type	Lustre backing filesystem type. Default: <code>ldiskfs</code>
mgt_format_params	“Catch all” for <code>mkfs.lustre</code> options. Default: <code>-quiet</code>
mdt_format_params	
ost_format_params	
mgt_mkfs_options	Format options for the backing fs.
mdt_mkfs_options	
ost_mkfs_options	
mgt_mount_options	MGT mount options. Default: no options
mdt_mount_options	MDT mount options. Default: no options
ost_mount_options	OST mount options. Default: <code>errors=remount-ro,extents,malalloc</code>
recovery_time_hard	Rolling recovery window timeout. Default: 300
quota	Enable quota support. Default: no (deprecated for Lustre 2.4.0 or greater)
quota_type	If quota support is enabled, options for quota support. (Deprecated for Lustre 2.4.0 or greater)

### B. Tuning Lustre Filesystems with `lustre_control`

Tunable Lustre parameters may be defined in a `fs_tune` file. The `fs_tune` file is directly called by the `lustre_control set_tune` command. For example, if the Lustre filesystem name `scratch` has a `fs_tune` file named `scratch.fs_tune`, the following command would apply the tuning settings.

```
# lustre_control set_tune -f scratch \  
scratch.fs_tune
```

The following shows example tuning settings:

```
# OSS tunables  
# Enable async-journal commits  
ost: lctl set_param  
obdfilter.*.sync_journal=0  
  
# Enable writethrough cache  
ost: lctl set_param \  
obdfilter.*.writethrough_cache_enable=1  
  
# MGS tunables  
# Set llite max read ahead to 16 MB  
# Use '__fs_name__' keyword so we can apply  
this to any installed file system  
mgt: lctl conf_param \  
__fs_name__.llite.max_read_ahead_mb=16  
  
# MDS tunables  
# Set obd timeout to 300 seconds  
mdt: lctl set_param timeout=300
```

An example `fs_tune` file is available on the CIMS at the following location:

```
# /opt/cray/esms/cray-lustre-control-  
XX/default/etc/example.fs_tune
```

## IV. ESFSMON

Esfsmmon is implemented as a custom health check in Bright Cluster Manager. The main health check file is located on the CIMS at the following location:

```
/cm/local/apps/cmd/scripts/healthchecks/esfs  
mon_healthcheck
```

The health check is executed on the CIMS but also has components on each MDS/OSS. These are executed by the management daemon on each MDS/OSS eliminating ssh overhead.

Esfsmmon requires one argument when being configured into the CIMS monitoring setup for each filesystem to be monitored – the Lustre filesystem name. This allows esfsmmon to have separate instances monitoring multiple filesystems concurrently. The Lustre filesystems can be a mix of DNE and non-DNE configurations.

### A. ESFSMON: Modes of Operation

There are 3 modes of operation for esfsmmon. They are controlled by the existence or non-existence of files in `/var/esfsmmon` on the CIMS.

- 1) **NORMAL** – monitors the filesystem and will execute automated failover on errors.
  - a. Set by the non-existence of `/var/esfsmmon/esfsmmon_runsafe_fsname` or `/var/esfsmmon/esfsmmon_suspend_fsname`
  - b. Failures are logged to `/var/log/messages` and `/tmp/esfsmmon/fsname.fo_log`
- 2) **RUNSAFE** – monitors the filesystem but will not execute automated failover on errors.
  - a. Set by existence of `/var/esfsmmon/esfsmmon_runsafe_fsname`
  - b. Failures are logged to `/var/log/messages`
- 3) **SUSPENDED** – filesystem not monitored.
  - a. Set by existence of `/var/esfsmmon/esfsmmon_suspend_fsname`

To prevent unwanted automated failovers, esfsmmon sets itself to `runsafe` mode during failover and failback operations. The `lustre_control` utility also sets `runsafe` mode when stopping the Lustre filesystem.

### B. ESFSMON: Operations

By default, all non-failed nodes in the monitored Lustre filesystem are checked every 2 minutes. This check interval is configurable.

CLFS servers are grouped into node categories within Bright Cluster Manager. This allows simultaneous testing on all nodes within a category. The following 3 categories are used for each Lustre filesystem. The “`fsname`” would be replaced with the actual filesystem name.

- 1) **esfs-even-fsname** – All even-numbered nodes in the Lustre filesystem named “`fsname`”.
- 2) **esfs-odd-fsname** – All odd-numbered nodes in the Lustre filesystem named “`fsname`”.
- 3) **esfs-failed-fsname** – All nodes in the Lustre filesystem named “`fsname`” that have failed esfsmmon testing. These nodes will be excluded from further esfsmmon tests.

All “`esfs-even-fsname`” category nodes are tested then all “`esfs-odd-fsname`” category nodes are tested. Nodes in the “`esfs-failed-fsname`” category are not tested.

### C. ESFSMON: Health Checks

The following health checks are performed by esfsmon.

- 1) Power Status
  - a. Check of BMC power status. Failure of initial check triggers a retry to avoid declaring power failure on transient IPMI failures.
- 2) Node Status
  - a. Check for node DOWN status reported by the management daemon.
  - b. Must also fail a TCP ping before being declared “DOWN”.
- 3) TCP Ping
  - a. Failure of the initial ping triggers a retry to avoid transient ping failures on highly loaded servers.
- 4) LNet Ping
  - a. “ibstat” is checked. No active IB interfaces triggers a failure.
  - b. “lctl ping” to two nodes is attempted. At least one “lctl ping” must succeed. This test will retry over a period of 90 seconds to avoid transient failures.
- 5) Lustre Mounts
  - a. A missing Lustre target mount will trigger a failure.

Failure of any of the above tests will trigger a failover to the backup server.

When a failure is detected, esfsmon\_healthcheck calls the esfsmon\_action script which moves the failed node to the esfs-failed-fsname category, power off the failed node and then calls lustre\_control to failover the Lustre targets and perform any tuning.

The esfsmon\_action script is located on the CIMS at:

```
/cm/local/apps/cmd/scripts/actions/esfsmon_
action
```

### D. ESFSMON: Restoring a Failed Node to Service

Restoring a failed node back to service is performed with the esfsmon\_failback command on the CIMS. This command takes the failed node hostname as an argument. It is important to use esfsmon\_failback as it moves the node back to its original operational node category in Bright Cluster Manager so it will be monitored and calls lustre\_control to failback the Lustre targets from its failover partner. It will also run any Lustre tuning specified for the Lustre filesystem.

Prior to running esfsmon\_failback, the node should be powered on and manually tested. Tests to perform would include running “lsscsi” to ensure the node can see the Lustre targets and “lctl ping” of one other node to ensure the LNet is operational on this node.

Output from esfsmon\_failback is sent to stdout and logged to /var/log/messages on the CIMS.

### E. ESFSMON: esfsmon.conf

Environmental and other shared information that esfsmon components need are located in the esfsmon.conf file. This file is located on the CIMS at:

```
/cm/local/apps/cmd/etc/esfsmon.conf
```

This file is sourced by esfsmon\_healthcheck, esfsmon\_action and esfsmon\_failback. Changes to this file are read on the next execution of these scripts.

The following parameters are defined in the esfsmon.conf file.

- State and Data directories
  - State directory: /var/esfsmon
  - Data directory: /tmp/esfsmon
- Node category names for each Lustre filesystem
- LNet network names used by each Lustre filesystem
  - For example, the scratch filesystem may have nids on x.x.x.x@o2ib while scratch2 has nids on x.x.x.x@o2ib1.
- Base hostname for each Lustre filesystem
  - For example, the scratch filesystem may use hostnames like lustre01-mds001 and lustre01-oss001 while the scratch2 filesystem has lustre02-mds001 and lustre02-oss001. The base hostname for scratch is “lustre01-” and scratch2 is “lustre02-”.
- Absolute path on the CIMS to the lustre\_control fs\_tune files for each filesystem.
- For filesystems not using Lustre DNE, the hostname of the passive MDS.

### F. ESFSMON: Status and Logging

The failover activity of esfsmon is logged to /var/log/messages and /tmp/esfsmon/fsname.fo\_log.

The failback activity of esfsmon\_failback is logged to /var/log/messages and displayed to *stdout*.

The current esfsmon status is available from the Bright Cluster Manager “latesthealthdata” command for the CIMS. This information can be accessed in the interactive cluster management shell (cmsh) or directly from the normal shell as shown in the following examples. The examples use a CIMS hostname of “cims1”.

Interactive shell (cmsh):

```
# cmsh
[cims1]% device
[cims1->device]% latesthealthdata cims1 -v
```

Normal shell:

```
# cmsh -c "device;latesthealthdata cims1 -v"
```

## V. LUSTRE MONITOR TOOL

The Lustre Monitor Tool (LMT) is included with the ESF software release. LMT collects statistics published in `/proc/fs/lustre` every 5 seconds using the Cerebro monitoring system.

LMT components are installed on both the CIMS and the CLFS nodes. The `lmt-server` package is installed on the CIMS. This provides the Cerebro monitoring system. Cerebro uses the MySQL database server on the CIMS to store data. The data aggregation script, `/usr/share/lmt/cron/lmt_agg.cron`, should be prepared to run via cron.

The `lmt-server-agent` package is installed on each CLFS node. This provides the Cerebro monitor plug-in, `ltop` client and other utilities for administering LMT.

Table 3 describes the statistics collected by LMT.

TABLE 3. LMT DATA COLLECTION

OSS/OST	
Statistic	Description
OSC Status	The MDS's view of the OST.
OSS	The OSS node hostname.
Export Count	There is one for each client that has this filesystem mounted plus one for the MDS and one for the OST itself.
Reconnects per second	Number of connect and reconnects per second.
Read/Write Rates	In megabytes per second.
Bulk RPCs per Second	IOPS – read and write combined.
OST Locks	The number of OST resource locks currently granted.
OST Lock Grant Rate	The lock grant rate.
OST Lock Cancellation Rate	The lock cancellation rate.
% CPU	The percentage of CPU in use on the OSS.
% MEM	The percentage of Memory in use on the OSS.
% SPC	The percentage of space in use.
MDS/MDT	
% CPU	The percentage of CPU in use on the MDS.
KB Free	The number of KiloBytes free on the MDS.
KB used	The number of KiloBytes used on the MDS.
inodes free	The number of free inodes.
inodes used	The number of inodes used.
open/close Rates	The rates of open and close operations.
mknod	The rate of mknod operations.
link/unlink	The rate of link and unlink operations.
mkdir/rmdir	The rate of mkdir and rmdir operations.
rename	The rate of rename operations.

To setup a Cerebro database for a Lustre filesystem named “scratch”, the following command is used.

```
# lmtinit -a scratch
```

Monitoring is enabled or disabled by starting or stopping the Cerebro daemon (`cerebrod`) on both the CIMS and all CLFS nodes in the filesystem.

LMT provides a live streaming “top”-like data monitor called “`ltop`” that displays all statistics being monitored. There is also an interactive shell “`lmtsh`” that can be used to display historical data from the Cerebro database for each filesystem being monitored.

## VI. TROUBLESHOOTING

Troubleshooting `esfsmon` is best approached by checking “`latesthealthdata`” in Bright Cluster Manager (`cmsh` or `cmgui`) for the filesystem and `/var/log/messages` for indications of what is happening or not happening correctly. The `/tmp/esfsmon/fsname.fo_log` is another source of information to be used.

If a filesystem is not failing over automatically, be sure to check the `/var/esfsmon/` directory for “`esfsmon_runsafe_fsname`” and “`esfsmon_suspend_fsname`” files that would prevent failover actions.

## VII. DOCUMENTATION

The following documents provide installation, configuration and operational information for `esfsmon` and LMT on CLFS systems.

- Data Management Platform (DMP) Administrator’s Guide – S-2327-C
  - `esfsmon` installation and configuration
  - LMT installation and configuration
- Installing Lustre® Filesystem by Cray® (CLFS) Software – S-2521-C
- Installing Cray® Integrated Management Services (CIMS) Software – S-2522-E
- LMT – <https://github.com/chaos/lmt/wiki>