



# Technical Bulletin

No. 440 ♦ November 21, 2008

## CHAOS 4.1/Toss 1.1 Release Notes

### Summary

CHAOS 4.1/TOSS 1.1 represents a refresh of the CHAOS software stack against Red Hat Enterprise Linux 5.2, along with other minor bug fixes and enhancements. The following table represents some of the major software updates contained in CHAOS 4.1/TOSS 1.1:

Software	CHAOS 4.0	CHAOS 4.1
SLURM	1.2	1.3
Lustre	1.6.2	1.6.6

### Red Hat Release Notes

Red Hat's release notes for RHEL5.2 are available from their Web site at the following URL:

[http://www.redhat.com/docs/en-US/Red\\_Hat\\_Enterprise\\_Linux/5.2/html/Release\\_Notes/x86\\_64/index.html](http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.2/html/Release_Notes/x86_64/index.html)

### GCC 4.3 Preview

An install of GNU Compiler Collection (GCC) 4.3 is included with CHAOS 4.1 as a technology preview release. This preview includes a version of the 4.3 gfortran compiler. All tools in this release are accessible via a "43" suffix, e.g., gcc43, gfortran43, and g++43.

### Lustre 1.6.6

Beyond the numerous bug fixes and small performance improvements included in the latest Lustre release, there are also some major design enhancements. Adaptive Timeouts is a new approach of managing timeouts internal to Lustre that will result in greater resilience of Lustre operations to variations of load. The scheduling of LNET (the networking layer internal to Lustre) has been improved. "Lazystatfs" is a modification to `statfs`, which allows the statusing of a storage server (OSS) to become non-blocking, thus eliminating some false impressions that the file system is hung. Finally, a new and noteworthy feature to the Lustre utilities is "lfs check servers." It is the recommended probe of file system health, and it is lightweight and avoids calls that may block under some common conditions. Additional details are provided below.

#### Adaptive Timeouts

Adaptive timeouts are a mechanism to set RPC timeouts. Servers track actual RPC completion times and report estimated completion times for future RPCs back to clients. The clients use these estimates to set their future RPC timeout values. If server request processing slows for any reason, the RPC completion estimates increase, and the clients allow more time for RPC completion.

If RPCs queued on the server approach their timeouts, then the server sends an early reply to the client telling the client to allow more time. In this manner, clients avoid RPC timeouts and disconnect/reconnect cycles. Conversely, as a server speeds up, RPC timeout values decrease, allowing faster detection of non-responsive servers and faster attempts to reconnect to a server's failover partner.

## Better Scheduling of LNET Router Resources

In previous releases, Lustre network traffic pending for down servers can tie up resources on LNET routers, thus affecting access to all Lustre file systems routed through those routers. This release includes improvements to LNET's ability to track the health of network peers and manage router resources.

### Lazystatfs

Clients may use the `lazystatfs` mount option to modify `statfs(2)` to skip down OSTs. The storage capacity on the skipped servers is then not included in statistics reported by `df(1)`. Without this option, `statfs(2)` would hang if any servers are down.

Please use `lfs check servers` (a Lustre-specific command) instead of `df(1)` to check Lustre file system health.

## SLURM 1.3

The version of SLURM in CHAOS 4.1/TOSS 1.1 has been updated to SLURM 1.3, the latest release series. There are several user-visible changes in this release, but it is not expected that many users will need to change how they use SLURM relative to the SLURM 1.2 series. The changes in this release are described below.

The `srun` options to allocate nodes interactively, attach to a running process, and submit a batch script have been replaced by dedicated commands that provide the same functionality:

<b>srun option</b>	<b>replaced by</b>
<code>--allocate</code>	<code>salloc</code>
<code>--attach</code>	<code>sattach</code>
<code>--batch</code>	<code>sbatch</code>

There are many other options to `srun`, and the new commands support these same options where appropriate. New `srun` options are:

<code>--exclusive</code>	allows job steps to be allocated processors not already assigned to other job steps. This can be used to execute multiple job steps simultaneously within a job allocation and have SLURM perform resource management for the job steps much like it does for jobs. If dedicated resources are not immediately available, the job step will be executed later unless the <code>--immediate</code> option is also set.
<code>--pty</code>	starts the job with a pseudo terminal attached to task zero (all other tasks have I/O discarded).
<code>--mem-per-cpu</code>	memory limits are newly applied on a per-CPU basis. (Replaces <code>--job-mem</code> option from <code>srun</code> , <code>salloc</code> , and <code>sbatch</code> commands.)

The `slaunch` command has been removed. All of its functionality is provided by the `srun` command.

While `srun` used to offer the ability to specify a feature that must be present on all the nodes allocated to the job, the SLURM 1.3 `srun` provides the flexibility to independently specify the number of nodes in the allocation that must have that feature. For example:

```
srun --nodes=16 --constraint=graphics*4 ...
```

More flexible time limit specification options:

```
<minutes>
<minutes>:<seconds>
<hours>:<minutes>:<seconds>
<days>-<hours>:<minutes>:<seconds>
```

Much richer job dependency support:

- Each job can be dependent upon many other jobs.
- Several dependency types added.
  - Wait for other jobs to begin.
  - Wait for other jobs to complete successfully (exit code of zero).
  - Wait for other jobs to complete (any exit status).
  - Wait for other jobs to fail.

New variables in execution environment (see “ENVIRONMENT VARIABLES” section of the `srun(1)` man page for complete list).

```
SLURM_JOB_DEPENDENCY
SLURM_JOB_NAME
```

## New Behavior

SLURM 1.3 now offers the ability to configure a default behavior when jobs fail. Formerly, the behavior was to requeue the job. SLURM 1.3 installations on LC machines will now have `no-requeue` configured as the default behavior. Users can override this default and request the requeue behavior via the `msub -l resfailpolicy=requeue` or `sbatch --requeue` options.

The `--requeue` option is independent from the `srun --no-kill` option. The `--no-kill` option instructs SLURM to continue running the job when one or more nodes fail. This feature has not changed between SLURM 1.2 and 1.3.

The equivalent Moab command for the `--no-kill` behavior is

```
msub -l resfailpolicy=ignore
```

## New SLURM Database

SLURM 1.3 includes the new SLURM database. This database contains all the fair-share account (bank) hierarchy and user memberships. The database also contains a record of all running and completed jobs. The SLURM database will soon replace LCRM’s (and Moab’s) job accounting database, `lrmusage`. SLURM tools available to generate accounting reports include `sacct` and `sreport`. There is also a new SLURM tool to view or modify various scheduling limits and parameters called `sacctmgr`. Man pages are available for all of these tools.

**If you have any questions, please contact the LC Hotline—  
send e-mail to [lc-hotline@llnl.gov](mailto:lc-hotline@llnl.gov) or [lc-hotline@pop.llnl.gov](mailto:lc-hotline@pop.llnl.gov) (SCF)  
or phone (925) 422-4531**

Web Pages	
<a href="http://computing.llnl.gov/">http://computing.llnl.gov/</a>	User Information
<a href="http://computation.llnl.gov/icc/">http://computation.llnl.gov/icc/</a>	Department home page
<a href="https://lc.llnl.gov/computing/techbulletins/">https://lc.llnl.gov/computing/techbulletins/</a>	Technical Bulletins
<a href="http://www.llnl.gov/computing/">http://www.llnl.gov/computing/</a>	SCF only