Release Notes for Platform **LSF®**

Version 6.0
March 12 2004
Platform Computing

Comments to: doc@platform.com

Contents

Platform™

What's New in the Platform LSF Version 6.0

Platform LSF Version 6.0 introduces the following new features:

Policy management

Goal-oriented SLA-driven scheduling

Goal-oriented SLA-driven scheduling policies help you configure your workload so that your jobs are completed on time and reduce the risk of missed deadlines:

- They enable you to focus on the "what and when" of your projects, not the low-level details of "how" resources need to be allocated to satisfy various workloads.
- They define a "just-in-time" service-level agreement between LSF administrators and LSF users.

You implement your SLA scheduling policies in *service classes* associated with your projects and users. Each service class defines how many jobs should be run to meet different kinds of goals:

- *Deadline goals*—A specified number of jobs should be completed within a specified time window. For example, run all jobs submitted over a weekend.
- *Velocity goals*—Expressed as concurrently running jobs. For example: maintain 10 running jobs between 9:00 a.m. and 5:00 p.m. Velocity goals are well suited for short jobs (run time less than one hour). Such jobs leave the system quickly, and configuring a velocity goal ensures a steady flow of jobs through the system.

- *Throughput goals*—Expressed as number of finished jobs per hour. For example: finish 15 jobs per hour between the hours of 6:00 p.m. and 7:00 a.m. Throughput goals are suitable for medium to long running jobs. These jobs stay longer in the system, so you typically want to control their rate of completion rather than their flow.

You use the `bsla` command to track the progress of your projects and see whether they are meeting the goals of your policy.

Platform LSF License Scheduler

Platform LSF License Scheduler ensures that higher priority work never has to wait for a license. Prioritized sharing of application licenses allows you to make policies that control the way software licenses are shared among different users in your organization.

You configure your software license distribution policy and LSF intelligently allocates licenses to improve quality of service to your end users while increasing throughput of high-priority work and reducing license costs.

It has the following features:

- Applies license distribution policies fairly among multiple projects cluster-wide
- Easily configurable distribution policies; instead of assigning equal share of licenses to everyone, you can give more licenses to larger or more important projects
- Guaranteed access to a minimum portion of licenses, no matter how heavily loaded the system is
- Controls the distribution of licenses among jobs and tasks it manages and still allows users to check out licenses directly
- Preempts lower priority jobs and releases their licenses to allow higher priority jobs to get the license and run.
- Provides visibility of license usage with `blusers` command

See *Using Platform LSF License Scheduler* for installation and configuration instructions.

Platform LSF license-aware scheduling is available as separately installable add-on packages located in `/license_scheduler/` on the Platform FTP site (`ftp.platform.com/`).

Job-level exception management

Configure hosts and queues so that LSF takes appropriate action automatically when it detects exceptional conditions while jobs are running. Customize what exceptions are detected, and their corresponding actions.

LSF detects:

- Job exceptions:
  - Job underrun—job ends too soon (run time is less than expected). Underrun jobs are detected when a job exits abnormally
  - Job overrun—job runs too long (run time is longer than expected)
  - Idle job—running job consumes less CPU time than expected (in terms of cputime/runtime)
- Host exceptions:

- LSF detects "black hole" or "job-eating" hosts. LSF monitors the job exit rate for hosts, and closes the host if the rate exceeds a threshold you configure.
- A host can still be available to accept jobs, but some other problem prevents the jobs from running. Typically jobs dispatched to such problem hosts exit abnormally.

Queue-based fairshare

Prevents starvation of low-priority work and ensures high-priority jobs get the resources they require by sharing resources among queues. Queue-based fairshare extends your existing user- and project-based fairshare policies by enabling flexible slot allocation per queue based on slot share units you configure.

User fairshare by queue priority

Improves control of user-based fairshare by taking queue priority into account for dispatching jobs from different queues against the same user fairshare policy. Within the queue, dispatch order is based on share quota.

Job group support

Use LSF job groups to organize and control a collection of individual jobs in higher level work units for easy management. A job group is a container for jobs in much the same way that a directory in a file system is a container for files. For example, you can organize jobs around groups that are meaningful to your business: a payroll application may have one group of jobs that calculates weekly payments, another job group for calculating monthly salaries, and a third job group that handles the salaries of part-time or contract employees.

Jobs groups increase end-user productivity by reducing complexity:

- Submit, view, and control jobs according to their groups rather than looking at individual jobs
- Create job group hierarchies
- Move jobs in and out of job groups as needed
- Kill, stop resume and send job control actions to entire job groups
- View job status by group

High Performance Computing

Dynamic ptile enforcement

Parallel jobs now have a flexible choice of the number of CPUs in the different kinds of hosts in a heterogeneous cluster.

Improves the performance and throughput of parallel jobs by setting multiple `ptile` values in a `span` string according to the CPU configuration of the host type or model.

You can specify various `ptile` values in the queue (RES_REQ in `lsb.queues`, or at job submission with `bsub -R`):

- Default `ptile` value, specified by *n* processors. For example:
  `span[ptile=4]`

  LSF allocates 4 processors on each available host, regardless of how many processors the host has.

- Predefined `ptile` value, specified by '!'. For example:

  ```
  span[ptile='!']
  ```

  LSF uses the predefined maximum job slot limit in `lsb.hosts` (MXJ per host type/model) as its value.
- Predefined `ptile` value with optional multiple `ptile` values, per host type or host model. For example:

  ```
  span[ptile='!',HP:8,SGI:8,LINUX:2] same[type]
  ```

  The job requests 8 processors on a host of type HP or SGI, and 2 processors on a host of type LINUX, and the predefined maximum job slot limit in `lsb.hosts` (MXJ) for other host types.

Resource requirement specification for advance reservation

You no longer need to specify a host list manually for your advance reservations. Specify a resource requirement string with the `-R` option of `brsvadd` instead of or in addition to a list of hosts. This makes advance reservation specification more flexible by reserving host slots based on your specific resource requirements. Only hosts that satisfy the resource requirement expression are reserved.

Administration and diagnosis

Scheduler dynamic debug

Enables dynamic debugging of the LSF scheduler daemon (`mbschd`) without reconfiguring the cluster. Administrators no longer need to run `badmin mbdrestart` to debug the LSF scheduler:

```
badmin schddebug [-c class_name] [-l debug_level] [-f logfile_name] [-o]
```

```
badmin schdtime [-l timing_level] [-f logfile_name] [-o]
```

Administrator action messages

Improves communication of LSF status to users. Users know the reason for the administrator actions, and administrators can easily communicate actions to users.

Administrators can attach a message to `mbatchd` restart, and host and queue operations:

- Use the `-C` option of `badmin mbdrestart` to log an administrator comment in `lsb.events`. For example,

  ```
  % badmin mbdrestart -C "Configuration change"
  ```

  The comment text `Configuration change` is recorded in `lsb.events`.
- Use the `-C` option of `badmin hclose` and `badmin hopen` to log an administrator comment in `lsb.events`. For example,

  ```
  % badmin hclose -C "Weekly backup" hostB
  ```

  The comment text `Weekly backup` is recorded in `lsb.events`. If you close or open a host group, each host group member displays the same comment string.
- Use the `-C` option of `badmin` queue commands `qclose`, `qopen`, `qact`, and `qinact` to log an administrator comment in `lsb.events`. For example,

  ```
  % badmin qclose -C "change configuration" normal
  ```

  The comment text `change configuration` is recorded in `lsb.events`.

To see administrator comments, users run `badmin hist`, `badmin mbdhist`, `badmin hhist`, or `badmin qhist`.

Platform LSF Reports

Understand cluster operations better, so that you can improve performance and troubleshoot configuration problems.

Platform LSF Reports provides a lightweight reporting package for single LSF clusters. It provides simple two-week reporting for smaller LSF clusters (about 100 hosts, 1,000 jobs/day) and shows trends for basic cluster metrics by user, project, host, resource and queue.

LSF Reports provides the following historical information about a cluster:

- Cluster load

  Trends the LSF internal load indices: status, r15s, r1m, r15m, ut, pg, ls, it, swp, mem, tmp, and io.

- Cluster service level

  Shows the average cluster service level using the following metrics: CPU time, memory and swap consumption, job runtime, job pending time, and job turnaround time

- Cluster throughput

  Shows the amount of work pushed through the cluster, using both accounting information (total number of submitted, completed, and exited jobs) and sampled information (the minimum, maximum, and average number of running and pending jobs, by state and type).

- Shared resource usage

  Shows the total, free, and used shared resources for the cluster.

- Reserved resource usage

  Shows the actual usage of reserved resources.

- License usage

  Shows peak, average, minimum, and maximum license usage by feature.

- License consumption

  Shows license minutes consumed by user, feature, vendor, and server.

LSF Reports also provides two monitoring commands, `blmon` and `slamon`:

- Use `blmon` to track license data for licenses controlled by LSF License Scheduler.
- Use `slamon` to monitor the progress of jobs running in service classes configured for goal-oriented SLA scheduling.

See *Platform LSF Reports Reference* for installation and configuration instructions.

Platform LSF Reports is available as separately installable add-on packages located in `/lsf_reports/` on the Platform FTP site (`ftp.platform.com/`).

Run-time enhancements

Thread limit enforcement

Control job thread limit like other limits. Use `bsub -T` to set the limit of the number of concurrent threads for the whole job. The default is no limit. In the queue, set THREADLIMIT to limit the number of concurrent threads that can be part of a job. Exceeding the limit causes the job to terminate.

Non-normalized job run time limit

Presents consistent job run time limits no matter which host runs the job. With non-normalized job run limit configured, job run time is not normalized by CPU factor.

If ABS_RUNLIMIT=Y is defined in `lsb.params`, the run time limit is not normalized by the host CPU factor. Absolute wall-clock run time is used for all jobs submitted with a run limit.

Resource allocation limit display (blimits command)

Improves visibility to resource allocation limits. If your job is pending because some configured resource allocation limit has been reached, you can find out what limits may be blocking your job.

Use the `blimits` command to show the dynamic counters of each resource allocation limit configured in `lsb.resources`.

New platform support

Platform LSF Version 6.0 is now supported on the following systems:

- Windows 2000 Professional. Application-level checkpointing is supported. Platform Make and lstcsh, are not supported.
- Windows 2003 Server. Application-level checkpointing is supported. Platform Make and lstcsh, are not supported.
- Apple Macintosh systems running Mac OS X
- NEC Supercomputer SX-6 running SUPER-UX 12.1

  lstcsh, lsmake, and Platform LSF MultiCluster are not supported. Kernel-level and application-level checkpointing are supported. User-level checkpointing is not supported

New security settings in Windows 2003

New security defaults added to Windows 2003 affect Platform LSF operation:

- In previous versions of Platform LSF for Windows, all users have the **Log on as a batch job** privilege. In Platform LSF Version 6.0, all users must have the privilege **Impersonate after authentication** for the job to accept client connections.
- Platform LSF servers require access to the Windows command interpreter file (`cmd.exe`), which is disabled by default in Windows 2003.

Windows installation on Windows 2000 (SP4) and Windows 2003 now sets the following:

- Adds the privilege **Impersonate after authentication** to **Everyone**.
- Grants **read+execution** right of the Windows command interpreter file (`cmd.exe`) to the **Users** group of each LSF server.

Upgrade and Compatibility Notes

UPGRADE document

To upgrade to LSF Version 6.0, follow the steps in `upgrade.html`.

API Compatibility between LSF 5.x and Version 6.0

Full backward compatibility: your applications will run under LSF Version 6.0 without changing any code.

The Platform LSF Version 6.0 API is fully compatible with the LSF Version 5.*x* and Version 4.*x* API. An application linked with the LSF Version 5.*x* and Version 4.*x* library will run under LSF Version 6.0 without relinking.

To take full advantage of new Platform LSF Version 6.0 features, you should recompile your existing LSF applications with LSF Version 6.0.

Server host compatibility Platform LSF

You must upgrade the LSF master hosts in your cluster to Version 6.0.

LSF 5.*x* servers are compatible with Version 6.0 master hosts. All LSF 5.x features are supported by 6.0 master hosts except:

To use new features introduced in Platform LSF Version 6.0, you must upgrade all hosts in your cluster to 6.0.

Platform LSF MultiCluster

You must upgrade the LSF master hosts in all clusters to Version 6.0.

New configuration parameters and environment variables

The following new parameters and environment variables have been added for LSF Version 6.0:

lsb.hosts

EXIT_RATE specifies a threshold in minutes for exited jobs

lsb.params

- EADMIN_TRIGGER_DURATION defines how often `LSF_SERVERDIR/eadmin` is invoked once a job exception is detected.
- JOB_EXIT_RATE_DURATION defines how long LSF waits before checking the job exit rate for a host.
- ABS_RUNLIMIT—if set, the run time limit specified by the `-W` option of `bsub`, or the RUNLIMIT queue parameter in `lsb.queues` is not normalized by the host CPU factor. Absolute wall-clock run time is used for all jobs submitted with a run limit.

lsb.queues

- DISPATCH_ORDER defines an ordered cross-queue fairshare set
- JOB_IDLE specifies a threshold for idle job exception handling
- JOB_OVERRUN specifies a threshold for job overrun exception handling
- JOB_UNDERRUN specifies a threshold for job underrun exception handling
- RES_REQ accepts multiple ptile specifications in the span section for dyamic ptile enforcement
- SLOT_POOL is the name of the pool of job slots the queue belongs to for queue-based fairshare

- SLOT_SHARE specifies the share of job slots for queue-based fairshare, representing the percentage of running jobs (job slots) in use from the queue
- THREADLIMIT limits the number of concurrent threads that can be part of a job. Exceeding the limit causes the job to terminate
- RUNLIMIT—if ABS_RUNLIMIT=Y is defined in `lsb.params`, the run time limit is not normalized by the host CPU factor. Absolute wall-clock run time is used for all jobs submitted to a queue with a run limit configured.

Environment variables

- LSB_SUB_EXTSCHED_PARAM

  Value of external scheduling options specified by `bsub -extsched`, or queue-level MANDATORY_EXTSCHED or DEFAULT_EXTSCHED
- LSB_SUB_JOB_WARNING_ACTION

  Value of job warning action specified by `bsub -wa`
- LSB_SUB_JOB_WARNING_TIME_PERIOD

  Value of job warning time period specified by `bsub -wt`

New command options and output

The following command options and output have changed for LSF Version 6.0:

bacct

- `-sla` *service_class_name* displays accounting statistics for jobs that ran under the specified service class
- `-x` displays jobs that have triggered a job exception (overrun, underrun, idle)

badmin

- `schddebug` sets message log level for `mbschd` to include additional information in log files
- `schdtime` sets timing level for `mbschd` to include additional timing information in log files
- `-C` *comment* logs the text of *comment* as an administrator comment record to `lsb.events` for the following subcommands:
  - `mbdrestart`
  - `qopen`
  - `qclose`
  - `qact`
  - `qinact`
  - `hopen`
  - `hclose`

bhist

`-l` displays:

- Job group modification
- Configured thread limit

bhosts

- `-x` displays hosts whose job exit rate has exceeded the threshold configured by EXIT_RATE in `lsb.hosts` for longer than JOB_EXIT_RATE_DURATION configured in `lsb.params`, and are still high
- `-l` displays the comment text if the LSF administrator specified an administrator comment with the `-C` option of the `badmin` host control commands `hclose` or hopen

bjobs

- `-g` *job_group_name* displays information about jobs attached to the specified job group
- `-l` displays the thread limit for the job
- `-sla` *service_class_name* displays jobs belonging to the specified service class
- `-x` displays unfinished jobs that have triggered a job exception (overrun, underrun, idle)

bkill

- `-g` *job_group_name* operates only on jobs in the specified job group
- `-sla` *service_class_name* operates on jobs belonging to the specified service class.

bmod

- `-g` *job_group_name* | `-gn`
- `-sla` *service_class_name* | `-slan`

bqueues

`-l` displays:

- Configured job exception thresholds and number of jobs in each exception state for the queue
- The job slot share (SLOT_SHARE) and the name of the share pool (SLOT_POOL) that the queue belongs to for queue-based fairshare
- DISPATCH_ORDER in a master queue for cross-queue fairshare
- The comment text if the LSF administrator specified an administrator comment with the `-C` option of the queue control commands `qclose`, `qopen`, `qact`, and `qinact`, `qhist`

bresume

`-g` *job_group_name* resumes only jobs in the specified job group

brsvadd

`-R` selects hosts for the reservation according to the specified resource requirements

bstop

- `-g` *job_group_name* suspends only jobs in the specified job group
- `-sla` *service_class_name* suspends jobs belonging to the specified service class

bsub

- `-g` *job_group_name* submits jobs in the specified job group
- `-R` accepts multiple ptile specifications in the span section for dyamic ptile enforcement
- `-sla` *service_class_name* specifies the service class where the job is to run
- `-T` *thread_limit* sets the limit of the number of concurrent threads to thread_limit for the whole job.
- `-W`—if ABS_RUNLIMIT=Y is defined in `lsb.params`, the run time limit is not normalized by the host CPU factor. Absolute wall-clock run time is used for all jobs submitted with a run limit.

New files added to installation

The following new files have been added to the Platform LSF Version 6.0 installation:

- `LSB_CONFDIR/`*cluster_name*`/configdir/lsb.serviceclasses`

- LSF_BINDIR/bgadd
- LSF_BINDIR/bgdel
- LSF_BINDIR/bjgroup
- LSF_BINDIR/blimits
- LSF_BINDIR/bsla
- LSF_SERVERDIR/eadmin
- LSF_LIBDIR/schmod_jobweight.so

Symbolic links to LSF files

If your installation uses symbolic links to other files in these directories, you must manually create links to these new files.

New accounting and job event fields

The following fields have been added to lsb.acct and lsb.events:

lsb.acct

- JOB_FINISH:

  sla (%s) is the SLA service class name under which the job runs.

lsb.events

- JOB_NEW:
  - sla (%s) is the SLA service class name under which the job runs
  - SLArunLimit (%d) is the absolute run time limit of the job for SLA service classes
  - jobGroup (%s) is the job group under which the job runs
- JOB_MODIFY2:
  - sla (%s) is the SLA service class name that the job is to be attached to
  - jobGroup (%s) is the job group under which the job runs
- JOB_EXECUTE:

  SLAscaledRunLimit (%d) is the run time limit for the job scaled by the execution host
- QUEUE_CTRL:

  ctrlComments (%s) is the administrator comment text from the -C option of badmin queue control commands qclose, qopen, qact, and qinact
- HOST_CTRL:

  ctrlComments (%s) is the administrator comment text from the -C option of badmin host control commands hclose and hopen
- MBD_DIE:

  ctrlComments (%s) is the administrator comment text from the -C option of badmin mbdrestart

Known Issues

Platform LSF Version 6.0 has no current issues.

Learning About Platform LSF

Finding Platform LSF information

Information about Platform LSF is available online from the following sources:

- "World Wide Web and FTP"
- "README, release notes, and UPGRADE"
- "Platform LSF documentation"

World Wide Web and FTP

The latest information about all supported releases of Platform LSF is available on the Platform Web site at `www.platform.com`.

If you have problems accessing the Platform Web site or the Platform FTP site, send email to `support@platform.com`.

Visit the Platform User Forum at `www.platformusers.net` to discuss workload management and strategies pertaining to distributed and Grid Computing.

README, release notes, and UPGRADE

Before downloading and installing LSF, be sure to read the files named `readme.html` and `release_notes.html`.

To upgrade to LSF Version 6.0, follow the steps in `upgrade.html`.

Platform LSF documentation

The LSF documentation directory `/distrib/6.0/docs/` contains the LSF documentation set in HTML and PDF format.

Download or view LSF documentation online

View or download the LSF documentation in HTML or PDF format:

- Platform Computing Web site:
  `www.platform.com/lsf_docs`
- Platform Computing FTP site (`ftp.platform.com`):
  `/distrib/6.0/docs/`

Online manuals

TItle	PDF	HTML .zip	HTML .tar.Z
Installing and upgrading			
README for Platform LSF Version 6.0	330 KB		
Release Notes for Platform LSF Version 6.0	205 KB		
Installing Platform LSF on UNIX and Linux (`lsfinstall`)	195 KB	40 KB	60 KB
Upgrading Platform LSF on UNIX and Linux	290 KB		
Using			
Running Jobs with Platform LSF	795 KB	105 KB	145 KB
Administering			
Platform LSF Administrator's Primer	755 KB	170 KB	170 KB
Administering Platform LSF	5,055 KB	760 KB	1,040 KB
Platform LSF Reference	4,790 KB	515 KB	640 KB
Platform LSF Quick Reference	130 KB	—	—
Platform LSF on Windows	1295 KB	200 KB	255 KB
Licensing Platform LSF	640 KB	90 KB	135 KB
Platform LSF Cluster Management Tools	435 KB	50 KB	70 KB
Extending			
Using Platform LSF MultiCluster	920 KB	80 KB	110 KB
Using Platform LSF License Scheduler	130 KB	80 KB	115 KB
Using Platform Globus Toolkit	1,040 KB	115 KB	115 KB
Using the Platform LSF SDK	1,290 KB	210 KB	280 KB
Using Platform LSF Make	165 KB	—	—
Platform LSF Reports Reference	1,935 KB	—	—
Customizing the Platform LSF Web User Interface	315 KB	—	—
Platform LSF Web User Interface API Reference	560 KB	—	—
Complete Platform LSF Version 6.0 HTML Doc Set		2,360 KB	3,190 KB
Complete Platform LSF Version 6.0 PDF Doc Set		19,535 KB	14,007 KB

Integrations

Integrating Platform LSF and Avaki Data Grid	320 KB
Using Platform LSF with Rational ClearCase	335 KB
Using Platform LSF Frame Arrays	325 KB
Using Platform LSF with SNMP	170 KB

Platform training

Platform's Professional Services training courses can help you gain the skills necessary to effectively install, configure and manage your Platform products. Courses are available for both new and experienced users and administrators at our corporate headquarters and Platform locations worldwide.

Customized on-site course delivery is also available.

Find out more about Platform Training at `www.platform.com/training`, or contact `Training@platform.com` for details.

Getting Technical Support

Contacting Platform

Contact Platform Computing or your LSF vendor for technical support. Use one of the following to contact Platform technical support:

Email support@platform.com

World Wide Web www.platform.com

Phone

- North America: +1 905 948 4297
- Europe: +44 1256 370 530
- Asia: +86 10 6238 1125

Toll-free Phone 1-877-444-4LSF (+1 877 444 4573)

Mail

Platform Support
Platform Computing Corporation
3760 14th Avenue
Markham, Ontario
Canada L3R 3T7

When contacting Platform, please include the full name of your company.

We'd like to hear from you

If you find an error in any Platform documentation, or you have a suggestion for improving it, please let us know:

Email doc@platform.com

Mail

Information Development
Platform Computing Corporation
3760 14th Avenue
Markham, Ontario
Canada L3R 3T7

Be sure to tell us:

- The title of the manual you are commenting on
- The version of the product you are using
- The format of the manual (HTML or PDF)

Copyright

Document redistribution policy

Internal redistribution

Trademarks