

Platform LSF
Version 9 Release 1.3

Release Notes



Platform LSF
Version 9 Release 1.3

Release Notes



Note

Before using this information and the product it supports, read the information in "Notices" on page 35.

First edition

This edition applies to version 9, release 1 of IBM Platform LSF (product number 5725G82) and to all subsequent releases and modifications until otherwise indicated in new editions.

Significant changes or additions to the text and illustrations are indicated by a vertical line (|) to the left of the change.

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

In the IBM Knowledge Center, add your comments and feedback to any topic.

You can also send your suggestions, comments and questions to the following email address:

pccdoc@ca.ibm.com

Be sure include the publication title and order number, and, if applicable, the specific location of the information about which you have comments (for example, a page number or a browser URL). When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright IBM Corporation 1992, 2014.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Release Notes for IBM

Platform LSF Version 9.1.3. 1

Learn more about IBM Platform LSF	1
We'd like to hear from you	3
Requirements and compatibility	3
Installation and migration notes	7
LSF Express Edition (Linux only)	7
IBM Platform entitlement files	10
What's new in Platform LSF Version 9.1.3	11
Known issues.	29
Limitations	31

Bugs fixed.	31
---------------------	----

Chapter 2. Platform LSF product packages. 33

Downloading the Platform LSF product packages	34
---	----

Notices 35

Trademarks	37
Privacy policy considerations	37

Chapter 1. Release Notes for IBM Platform LSF Version 9.1.3

Release date: July 2014

Last modified: 18 July 2014

Learn more about IBM Platform LSF

Information about IBM Platform LSF (Platform LSF or LSF) is available from the following sources:

- IBM Platform Computing web site: www.ibm.com/systems/technicalcomputing/platformcomputing
- The LSF area of the IBM Support Portal: www.ibm.com/platformcomputing/support.html
- IBM Technical Computing community on IBM Service Management Connect: www.ibm.com/developerworks/servicemanagement/tc/plsf/index.html
- Platform LSF documentation

Platform LSF documentation

Platform LSF is available through a variety of channels and a variety of formats.

LSF documentation in the IBM Knowledge Center

The IBM Knowledge Center is the home for IBM product documentation. Find Platform LSF documentation in the IBM Knowledge Center on the IBM Web site: www.ibm.com/support/knowledgecenter/SSETD4/.

Search all the content in IBM Knowledge Center for subjects that interest you, or search within a product, or restrict your search to one version of a product. Sign in with your IBM ID to take full advantage of the personalization features available in IBM Knowledge Center. Create and print custom collections of documents you use regularly, and communicate with colleagues and IBM by adding comments to topics.

Documentation available through the IBM Knowledge Center may be updated and regenerated following the original release of Platform LSF 9.1.3.

LSF documentation packages

The Platform LSF documentation is contained in the LSF documentation packages:

- `lsf9.1.3_documentation.tar.Z`
- `lsf9.1.3_documentation.zip`

You can download, extract and install these packages to any server on your system to have a local version of the full LSF documentation set. Navigate to the location where you extracted the files and open `index.html` in any browser. Easy access to each document in PDF and HTML format is provided, as well as full search capabilities within the full documentation set or within a specific document type.

If you have installed IBM Platform Application Center (PAC), you can access and search the LSF documentation through the **Help** link in the user interface.

LSF documentation in PDF format

Platform LSF documentation is also available in PDF format On the IBM Publications Center: www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss.

Note: PDF format documentation available through www.ibm.com may be updated and regenerated following the original release of Platform LSF 9.1.3.

The documentation set for Platform LSF 9.1.3 includes the following PDF documents:

- *IBM Platform LSF Quick Start Guide* - GI13344000
- *Administering IBM Platform LSF* - SC27530203
- *IBM Platform LSF Foundations* - SC27530403
- *IBM Platform LSF Command Reference* - SC27530503
- *IBM Platform LSF Configuration Reference* - SC27530603
- *Running Jobs with IBM Platform LSF* - SC27530703
- *IBM Platform LSF Quick Reference* - GC27530903
- *Using IBM Platform LSF Advanced Edition* - SC27532103
- *Using IBM Platform LSF on Windows* - SC27531103
- *Using IBM Platform MultiCluster* - SC27531003
- *Installing IBM Platform LSF on UNIX and Linux* - SC27531403
- *Upgrading IBM Platform LSF on UNIX and Linux* - SC27531503
- *Migrating IBM Platform LSF Version 7 to IBM Platform LSF Version 9.1.3 on UNIX and Linux* -SC27531803
- *Installing IBM Platform LSF on Windows* - SC27531603
- *Migrating IBM Platform LSF Version 7 to IBM Platform LSF Version 9.1.3 on Windows* - SC27531703
- *IBM Platform LSF Security* - SC27530302
- *Using IBM Platform LSF with IBM Rational ClearCase* - SC27537700

Information about related Platform LSF Family products can be found in the following documents:

- *Using IBM Platform License Scheduler* - SC27530803
- *Release Notes for IBM Platform License Scheduler* - GI13341402
- *Using IBM Platform Dynamic Cluster* - SC27532002
- *Release Notes for IBM Platform Dynamic Cluster* - GI13341702
- *IBM Platform MPI User's Guide* - SC27475801
- *Release Notes for IBM Platform MPI: Linux* - GI13189602
- *Release Notes for IBM Platform MPI: Windows* - GI13189702
- *IBM Platform LSF Programmer's Guide* - SC27531202

LSF documentation in PDF format is also available for Version 9.1.2 and earlier releases on the IBM Support Portal: <http://www.ibm.com/support/customer/portal/sas/f/plcomp/platformlsf.html>.

IBM Technical Computing community

Connect. Learn. Share. Collaborate and network with the IBM Platform Computing experts at the IBM Technical Computing community. Access the Technical

Computing community on IBM Service Management Connect at <http://www.ibm.com/developerworks/servicemanagement/tc/>. Join today!

Service Management Connect is a group of technical communities for Integrated Service Management (ISM) professionals. Use Service Management Connect the following ways:

- Connect to become involved with an ongoing, open engagement among other users, system professionals, and IBM developers of Platform Computing products.
- Learn about IBM Technical Computing products on blogs and wikis, and benefit from the expertise and experience of others.
- Share your experience in wikis and forums to collaborate with the broader Technical Computing user community.

We'd like to hear from you

Contact IBM or your LSF vendor for technical support.

Or go to the IBM Support Portal: www.ibm.com/support

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

In the IBM Knowledge Center, add your comments and feedback to any topic.

You can also send your suggestions, comments and questions to the following email address:

pccdoc@ca.ibm.com

Be sure include the publication title and order number, and, if applicable, the specific location of the information about which you have comments (for example, a page number or a browser URL). When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Requirements and compatibility

The following sections detail requirements and compatibility for version 9.1.3 of Platform LSF.

System requirements

- IBM AIX 6.x and 7.x on IBM Power 6/7/8
- Linux Kernel 2.6 and 3.x on IBM Power 6/7/8
- Linux x64 Kernel 2.6 and 3.x on x86_64
- HP UX B.11.31 (64-bit) on HP 9000 Servers (PA-RISC)
- HP UX B.11.31 (IA64) on HP Integrity Servers (Itanium2)
- Solaris 10 and 11 on Sparc
- Solaris 10 and 11 on x86-64
- Cray XE6, XT6, XC-30, Linux Kernel 2.6, glibc 2.3 on x86_64
- ARMv8 Kernel 3.12 glibc 2.17
- ARMv7 Kernel 3.6 glibc 2.15 (LSF slave host only)

- Apple Mac OS 10.x (LSF slave host only)
- Windows 2003 SP1/2, 2008 x86, 7 x86, 8 x86, and 8.1 x86 on x86/x86_64 (32-bit)
- Windows 2003 SP1/2, 2003 CCE SP1/SP2, 2008 x64, 7 x64, 8.1 x64, 2008 R2 x64, HPC server 2008, 2012 x64, and 2012 R2 x64 on x86_64 (64-bit)

For detailed LSF system support information, refer to the Compatibility Table at:

www.ibm.com/systems/technicalcomputing/platformcomputing/products/lsf/

Master host selection

To achieve the highest degree of performance and scalability, use a powerful master host.

There is no minimum CPU requirement. For the platforms on which LSF is supported, any host with sufficient physical memory can run LSF as master host. Swap space is normally configured as twice the physical memory. LSF daemons use about 40 MB of memory when no jobs are running. Active jobs consume most of the memory LSF requires.

Note: If a Windows host must be installed as the master host, only 2008 R2 Server and Windows 2012 R2 Server are recommended as LSF Master hosts.

Cluster size	Active jobs	Minimum required memory (typical)	Recommended server CPU
			(Intel, AMD, OpenPower or equivalent)
Small (<100 hosts)	1,000	1 GB (32 GB)	any server CPU
	10,000	2 GB (32 GB)	recent server CPU
Medium (100-1000 hosts)	10,000	4 GB (64 GB)	multi-core CPU (2 cores)
	50,000	8 GB (64 GB)	multi-core CPU (4 cores)
Large (>1000 hosts)	50,000	16 GB (128 GB)	multi-core CPU (4 cores)
	500,000	32 GB (256 GB)	multi-core CPU (8 cores)

Server host compatibility

Platform LSF 7.x, 8.0.x, 8.3, and 9.1.x, servers are compatible with Platform LSF 9.1.3 master hosts. All LSF 7.x, 8.0.x, 8.3, and 9.1.x features are supported by Platform LSF 9.1.3 master hosts.

Important: To take full advantage of all new features introduced in the latest release of Platform LSF, you *must* upgrade all hosts in your cluster.

LSF Family product compatibility

IBM Platform RTM

Customers can use IBM Platform RTM (Platform RTM) 8.3 or 9.1.x to collect data from Platform LSF 9.1.3 clusters. When adding the cluster, select 'Poller for LSF 8' or 'Poller for LSF 9.1'.

IBM Platform License Scheduler

IBM Platform License Scheduler (License Scheduler) 8.3 and 9.1.x are compatible with Platform LSF 9.1.3.

IBM Platform Analytics

IBM Platform Analytics (Analytics) 8.3 and 9.1.x are compatible with Platform LSF 9.1.3 after the following manual configuration:

To have Analytics 8.3 or 9.1.x collect data from Platform LSF 9.1.3 clusters:

1. Set the following parameters in `lsb.params`:

- **ENABLE_EVENT_STREAM=Y**
- **ALLOW_EVENT_TYPE="JOB_NEW JOB_FINISH2 JOB_STARTLIMIT JOB_STATUS2 JOB_PENDING_REASONS"**
- **RUNTIME_LOG_INTERVAL=10**

2. Copy `elim.coreutil` to LSF:

```
cp ANALYTICS_TOP/elim/os_type/elim.coreutil $LSF_SERVERDIR
```

3. In `lsf.shared`, create the following:

```
Begin Resource
RESOURCENAME  TYPE          INTERVAL INCREASING DESCRIPTION
CORE_UTIL     String        300      ()         (Core Utilization)
End Resource
```

4. In `lsf.cluster.cluster_name`, create the following:

```
Begin ResourceMap
RESOURCENAME  LOCATION
CORE_UTIL     [default]
End ResourceMap
```

5. Restart all LSF daemons.
6. Configure user group and host group.
7. Run `lsid` and check the output.
8. Install Platform Analytics with **COLLECTED_DATA_TYPE=LSF**.
9. Check `perf.conf` to see `LSF_VERSION`.
10. Restart the Platform loader controller (**plc**).
11. Check the log files and table data to make sure there are no errors.
12. Change all the LSF related data loader intervals to 120 seconds, and run for one day. Check the **plc** and data loader log files to make sure there are no errors.

IBM Platform Application Center

IBM Platform Application Center (PAC) 8.3 and higher versions are compatible with Platform LSF 9.1.x after the following manual configuration.

If you are using PAC 8.3 with LSF 9.1.x, `$PAC_TOP/perf/lsf/8.3` must be renamed to `$PAC_TOP/perf/lsf/9.1`

For example:

```
mv /opt/pac/perf/lsf/8.3 /opt/pac/perf/lsf/9.1
```

API compatibility

To take full advantage of new Platform LSF 9.1.3 features, recompile your existing Platform LSF applications with Platform LSF 9.1.3.

Applications need to be rebuilt if they use APIs that have changed in Platform LSF 9.1.3.

New and changed Platform LSF APIs

The following APIs or data structures have changed or are new for LSF 9.1.3:

- `lsb_getallocFromHhostfile`
- `lsb_readrankfile`
- `struct addRsvRequest`
- `struct allocHostInfo`
- `struct appInfoEnt`
- `struct dependJobs`
- `struct eventRec`
- `struct jobFinishLog`
- `struct jobFinish2Log`
- `struct jobInfoEnt`
- `struct jobModLog`
- `struct jobNewLog`
- `struct jobResizeLog`
- `struct jobResizeNotifyStartLog`
- `struct jobResizeReleaseLog`
- `struct jobStartLog`
- `struct parameterInfo`
- `struct queriedJobs`
- `struct queueInfoEnt`
- `struct rsvInfoEnt`
- `struct submit`
- `struct userInfoEnt`

For detailed information about APIs changed or created for LSF 9.1.3, refer to the *IBM Platform LSF 9.1.3 API Reference*.

Third party APIs

The following third party APIs have been tested and supported for this release:

- DRMAA LSF API v 1.1.1
- PERL LSF API v1.0
- Python LSF API v1.0 with LSF 9

Packages are available at www.github.com.

For more information on using third party APIs with LSF 9.1.3 see the Technical Computing community on IBM Service Management Connect at www.ibm.com/developerworks/servicemanagement/tc/plsf/index.html.

Installation and migration notes

Consult the following note on installing and migrating from a previous version of LSF.

Upgrade Platform LSF on UNIX and Linux

Follow the steps in *Upgrading IBM Platform LSF on UNIX and Linux* (lsf_upgrade_unix.pdf) to run **lsfinstall** to upgrade LSF:

- Upgrade a pre-LSF Version 7 UNIX or Linux cluster to Platform LSF 9.1.x
- Upgrade an LSF Version 7 Update 2 or higher to Platform LSF 9.1.x

Important: *DO NOT* use the UNIX and Linux upgrade steps to migrate an existing LSF Version 7 or LSF 7 Update 1 cluster to LSF 9.1.3. Follow the manual steps in the document *Migrating IBM Platform LSF Version 7 to IBM Platform LSF Version 9.1.3 on UNIX and Linux* to migrate an existing LSF Version 7 or LSF 7 Update 1 cluster to LSF 9.1.3 on UNIX and Linux.

Migrate LSF Version 7 and Version 7 Update 1 cluster to LSF 9.1.3 on UNIX and Linux

Follow the steps in *Migrating IBM Platform LSF Version 7 to IBM Platform LSF Version 9.1.3 on UNIX and Linux* (lsf_migrate_unix.pdf) to migrate an *existing* LSF 7 or LSF 7 Update 1 cluster:

- Migrate an existing LSF Version 7 cluster to LSF 9.1.3 on UNIX and Linux
- Migrate an existing LSF Version 7 Update 1 cluster to LSF 9.1.3 on UNIX and Linux

Note: To migrate an LSF 7 Update 2 or higher cluster to LSF 9.1.3 follow the steps in *Upgrading IBM Platform LSF on UNIX and Linux*.

Migrate an LSF Version 7 or higher cluster to LSF 9.1.3 on Windows

To migrate an *existing* LSF 7 Windows cluster to Platform LSF 9.1.3 on Windows, follow the steps in *Migrating IBM Platform LSF Version 7 to IBM Platform LSF Version 9.1.3 on Windows*.

Note: If you want to migrate a pre-version 7 cluster to LSF 9.1.3, you must first migrate the cluster to LSF Version 7.

LSF Express Edition (Linux only)

LSF Express Edition is a solution for Linux customers with simple scheduling requirements and simple fairshare setup. Smaller clusters typically have a mix of sequential and parallel work as opposed to huge volumes of jobs. For this reason, several performance enhancements and complex scheduling policies designed for large-scale clusters are not applicable to LSF Express Edition clusters. Session Scheduler is available as an add-on component.

Platform product support with LSF Express Edition

The following IBM Platform products are supported in LSF Express Edition:

- IBM Platform RTM
- IBM Platform Application Center

- IBM Platform License Scheduler

The following IBM Platform products are *not* supported in LSF Express Edition:

- IBM Platform Analytics
- IBM Platform Process Manager

Default configuration for LSF Express Edition

The following table lists the configuration enforced in LSF Express Edition:

Parameter	Setting	Description
RESIZABLE_JOBS in <code>lsb.applications</code>	N	If enabled, all jobs belonging to the application will be auto resizable.
EXIT_RATE in <code>lsb.hosts</code>	Not defined	Specifies a threshold for exited jobs.
BJOBS_RES_REQ_DISPLAY in <code>lsb.params</code>	None	Controls how many levels of resource requirements <code>bjobs -l</code> will display.
CONDENSE_PENDING_REASONS in <code>lsb.params</code>	N	Condenses all host-based pending reasons into one generic pending reason.
DEFAULT_JOBGROUP in <code>lsb.params</code>	Disabled	The name of the default job group.
EADMIN_TRIGGER_DURATION in <code>lsb.params</code>	1 minute	Defines how often <code>LSF_SERVERDIR/eadmin</code> is invoked once a job exception is detected. Used in conjunction with job exception handling parameters <code>JOB_IDLE</code> , <code>JOB_OVERRUN</code> , and <code>JOB_UNDERRUN</code> in <code>lsb.queues</code> .
ENABLE_DEFAULT_EGO_SLA in <code>lsb.params</code>	Not defined	The name of the default service class or EGO consumer name for EGO-enabled SLA scheduling.
EVALUATE_JOB_DEPENDENCY in <code>lsb.params</code>	Unlimited	Sets the maximum number of job dependencies <code>mbatchd</code> evaluates in one scheduling cycle.
GLOBAL_EXIT_RATE in <code>lsb.params</code>	2147483647	Specifies a cluster-wide threshold for exited jobs
JOB_POSITION_CONTROL_BY_ADMIN in <code>lsb.params</code>	Disabled	Allows LSF administrators to control whether users can use <code>btop</code> and <code>bbot</code> to move jobs to the top and bottom of queues.
LSB_SYNC_HOST_STAT_FROM_LIM in <code>lsb.params</code>	N	Improves the speed with which <code>mbatchd</code> obtains host status, and therefore the speed with which LSF reschedules rerunnable jobs. This parameter is most useful for a large clusters, so it is disabled for LSF Express Edition.
MAX_CONCURRENT_QUERY in <code>lsb.params</code>	100	Controls the maximum number of concurrent query commands.
MAX_INFO_DIRS in <code>lsb.params</code>	Disabled	The number of subdirectories under the <code>LSB_SHAREDIR/cluster_name/logdir/info</code> directory.

Parameter	Setting	Description
MAX_JOBID in lsb.params	999999	The job ID limit. The job ID limit is the highest job ID that LSF will ever assign, and also the maximum number of jobs in the system.
MAX_JOB_NUM in lsb.params	1000	The maximum number of finished jobs whose events are to be stored in lsb.events.
MIN_SWITCH_PERIOD in lsb.params	Disabled	The minimum period in seconds between event log switches.
MBD_QUERY_CPUS in lsb.params	Disabled	Specifies the master host CPUs on which mbatchd child query processes can run (hard CPU affinity).
NO_PREEMPT_INTERVAL in lsb.params	0	Prevents preemption of jobs for the specified number of minutes of uninterrupted run time, where minutes is wall-clock time, not normalized time.
NO_PREEMPT_RUN_TIME in lsb.params	-1 (not defined)	Prevents preemption of jobs that have been running for the specified number of minutes or the specified percentage of the estimated run time or run limit.
PREEMPTABLE_RESOURCES in lsb.params	Not defined	Enables preemption for resources (in addition to slots) when preemptive scheduling is enabled (has no effect if queue preemption is not enabled) and specifies the resources that will be preemptable.
PREEMPT_FOR in lsb.params	0	If preemptive scheduling is enabled, this parameter is used to disregard suspended jobs when determining if a job slot limit is exceeded, to preempt jobs with the shortest running time, and to optimize preemption of parallel jobs.
SCHED_METRIC_ENABLE in lsb.params	N	Enables scheduler performance metric collection.
SCHED_METRIC_SAMPLE_PERIOD in lsb.params	Disabled	Performance metric sampling period.
SCHEDULER_THREADS in lsb.params	0	Sets the number of threads the scheduler uses to evaluate resource requirements.
DISPATCH_BY_QUEUE in lsb.queues	N	Increases queue responsiveness. The scheduling decision for the specified queue will be published without waiting for the whole scheduling session to finish. The scheduling decision for the jobs in the specified queue is final and these jobs cannot be preempted within the same scheduling cycle.

Parameter	Setting	Description
LSB_JOBID_DISP_LENGTH in <code>lsf.conf</code>	Not defined	By default, LSF commands <code>bjobs</code> and <code>bhist</code> display job IDs with a maximum length of 7 characters. Job IDs greater than 9999999 are truncated on the left. When <code>LSB_JOBID_DISP_LENGTH=10</code> , the width of the JOBID column in <code>bjobs</code> and <code>bhist</code> increases to 10 characters.
LSB_FORK_JOB_REQUEST in <code>lsf.conf</code>	N	Improves <code>mbatchd</code> response time after <code>mbatchd</code> is restarted (including parallel restart) and has finished replaying events.
LSB_MAX_JOB_DISPATCH_PER_SESSION in <code>lsf.conf</code>	300	Defines the maximum number of jobs that <code>mbatchd</code> can dispatch during one job scheduling session.
LSF_PROCESS_TRACKING in <code>lsf.conf</code>	N	Tracks processes based on job control functions such as termination, suspension, resume and other signaling, on Linux systems which support cgroups' freezer subsystem.
LSB_QUERY_ENH in <code>lsf.conf</code>	N	Extends multithreaded query support to batch query requests (in addition to <code>bjobs</code> query requests). In addition, the <code>mbatchd</code> system query monitoring mechanism starts automatically instead of being triggered by a query request. This ensures a consistent query response time within the system. Enables a new default setting for <code>min_refresh_time</code> in <code>MBD_REFRESH_TIME</code> (<code>lsb.params</code>).
LSB_QUERY_PORT in <code>lsf.conf</code>	Disabled	Increases <code>mbatchd</code> performance when using the <code>bjobs</code> command on busy clusters with many jobs and frequent query request.
LSF_LINUX_CGROUP_ACCT in <code>lsf.conf</code>	N	Tracks processes based on CPU and memory accounting for Linux systems that support cgroup's memory and <code>cpuacct</code> subsystems.

IBM Platform entitlement files

Entitlement files are used for determining which edition of the product is enabled. The following entitlement files are packaged for LSF:

- LSF Standard Edition: `platform_lsf_std_entitlement.dat`
- LSF Express Edition: `platform_lsf_exp_entitlement.dat`
- LSF Advanced Edition: `platform_lsf_adv_entitlement.dat`

The entitlement file for the edition you use must be installed as `LSF_TOP/conf/lsf.entitlement`.

If you have installed LSF Express Edition, you can upgrade later to LSF Standard Edition or LSF Advanced Edition to take advantage of the additional functionality. Simply reinstall the cluster with the LSF Standard entitlement file (`platform_lsf_std_entitlement.dat`) or the LSF Advanced entitlement file (`platform_lsf_adv_entitlement.dat`).

You can also manually upgrade from LSF Express Edition to Standard Edition or Advanced Edition. Get the LSF Standard or Advanced Edition entitlement file, copy it to `LSF_TOP/conf/lsf.entitlement` and restart you cluster. The new entitlement enables the additional functionality of LSF Standard Edition, but you may need to manually change some of the default LSF Express configuration parameters to use the LSF Standard or Advanced features.

To take advantage of LSF SLA features in LSF Standard Edition, copy `LSF_TOP/LSF_VERSION/install/conf_tmpl/lsf_standard/lsb.serviceclasses` into `LSF_TOP/conf/lsbatch/LSF_CLUSTERNAME/configdir/`.

Once LSF is installed and running, run the `lsid` command to see which edition of LSF is enabled.

What's new in Platform LSF Version 9.1.3

The following topics detail new and changed behavior, new and changed commands, options, output, configuration parameters, environment variables, accounting and job event fields.

Changes to default LSF behavior

The following details changes to default LSF behavior.

Changes to task and slot concept

To keep up with the increasing density of hosts (cores/threads per node) and the growth in threaded applications (for example, a job may request 4 slots, then run 4 threads per slot, so in reality it is using more than 4 cores) there is greater disparity between what a user requests and what needs to be allocated to satisfy the request. This is particularly true in HPC environments where exclusive allocation of nodes is more prevalent.

In this release, an “HPC Allocation” feature is introduced, where the resources allocated (rather than the resources requested) can be used for accounting and fairshare purposes. For example, in a cluster of 16 nodes with 24 cores each, submitting a 16 way parallel job that places 1 task per node, which runs 4 threads per task, will:

- By default, show 16 cores (slots) in use. If the same job was submitted exclusively, by default LSF will still only show 16 cores (slots) in use.
- With the HPC allocation policy enabled, the same job would show 64 slots (cores) in use ($16 \times 1 \times 4$). And if the job had been submitted exclusively, it would show 384 slots (cores) in use.

For consistency, the “slot” concept in LSF has been superseded by “task”. In the first example above, a job running 4 processes each with 4 threads is 16 tasks, and with one task per core, it requires 16 cores to run.

A new parameter, **LSB_ENABLE_HPC_ALLOCATION** in `lsf.conf` is introduced. For new installations, this parameter will be enabled automatically (set to Y). For upgrades, it will be set to N and must be enabled manually.

When set to Y|y, this parameter changes the concept of the required number of slots for a job to the required number of tasks for a job. The specified numbers of tasks (using **bsub**), will be the number of tasks to launch on execution hosts. The allocated slots will change to all slots on the allocated execution hosts for an exclusive job in order to reflect the actual slot allocation.

When **LSB_ENABLE_HPC_ALLOCATION** is not set or is set to N|n, the following behavior change will still take effect:

- Pending reasons in **bjobs** output keep task concept
- **TASKLIMIT** replaces **PROCLIMIT**
- **PER_TASK** replaces **PER_SLOT**
- **IMPT_TASKBKLK** replaces **IMPT_SLOTBKLK**
- **FWD_TASKS** replaces **FWD_SLOTS**
- **RESOURCE_RESERVE_PER_TASK** replaces **RESOURCE_RESERVE_PER_SLOT**
- Event and API changes for task concept
- Field "alloc_slot nalloc_slot" for **bjobs -o** is available
- -alloc option is available for **bqueues**, **bhosts**, **busers**, and **bapp**
- Command help messages change to task concept
- Error messages in logs change to task concept

The following behavior changes take effect if only if **LSB_ENABLE_HPC_ALLOCATION** is set to Y|y:

- Command output for **bjobs**, **bhist**, and **bacct**
- Exclusive job slot allocation change

New and changed behavior

The following details new and changed behavior for LSF 9.1.3.

Restrict job size requested by parallel jobs

Specifying a list of allowed job sizes (number of tasks) in queues or application profiles enables LSF to check the requested job sizes when submitting, modifying, or switching jobs.

Certain applications may yield better performance with specific job sizes (for example, the power of two, so that the job sizes are x^2). The **JOB_SIZE_LIST** parameter in `lsb.queues` or `lsb.applications` defines a discrete list of allowed job sizes for the specified queues or application profiles. LSF will reject jobs requesting job sizes that are not in this list, or jobs requesting multiple job sizes.

The first job size in the **JOB_SIZE_LIST** is the default job size, which is assigned to jobs that do not explicitly request a job size. The rest of the list can be defined in any order:

```
JOB_SIZE_LIST=default_size [size ...]
```

For example, the following defines a job size list for the queue1 queue:

```

Begin Queue
QUEUE_NAME = queue1
...
JOB_SIZE_LIST=4 2 8 16
...
End Queue

```

This job size list allows 2, 4, 8, and 16 tasks. If you submit a parallel job requesting 10 tasks in this queue (`bsub -q queue1 -n 10 ...`), that job is rejected because the job size of 10 is not explicitly allowed in the list. The default job size is 4 tasks, and job submissions that do not request a job size are automatically assigned a job size of 4.

When using resource requirements to specify job size, the request must specify a single fixed job size and not multiple values or a range of values:

- When using compound resource requirements with `-n` (that is, `-n` with the `-R` option), ensure that the compound resource requirement matches the `-n` value, which must match a value in the job size list.
- When using compound resource requirements without `-n`, the compound resource requirement must imply a fixed job size number, and the implied total job size must match a value in the job size list.
- When using alternative resource requirements, each of the alternatives must request a fixed job size number, and all alternative values must match the values in the job size list.

When defined in both a queue (`lsb.queues`) and an application profile (`lsb.applications`), the job size request must satisfy both requirements. In addition, **JOB_SIZE_LIST** overrides any **TASKLIMIT** (**TASKLIMIT** replaces **PROCLIMIT** in LSF 9.1.3) parameters defined at the same level.

Terminate Orphan Jobs

Often, complex workflows are required with job dependencies for proper job sequencing as well as job failure handling. For a given job, called the parent job, there can be child jobs which depend on its state before they can start. If one or more conditions are not satisfied, a child job remains pending. However, if the parent job is in a state such that the event on which the child depends will never occur, the child becomes an orphan job. For example, if a child job has a **DONE** dependency on the parent job but the parent ends abnormally, the child will never run as a result of the parent's completion and it becomes an orphan job.

Keeping orphan jobs in the system can cause performance degradation. The pending orphan jobs consume unnecessary system resources and add unnecessary loads to the daemons which can impact their ability to do useful work.

Orphan job termination may be enabled in two ways:

- An LSF administrator enables the feature at the cluster level by defining a cluster-wide termination grace period with the parameter **ORPHAN_JOB_TERM_GRACE_PERIOD** in `lsb.params`. The cluster-wide termination grace period applies to all dependent jobs in the cluster.
- Users can use the **-ti** suboption of jobs with job dependencies specified by **bsub -w** to enforce immediate automatic orphan termination on a per-job basis even if the feature is disabled at the cluster level. Dependent jobs submitted with this option that later become orphans are subject to immediate termination without the grace period, even if it is defined.

Submitting a job with a user-specified host file

When submitting a job, you can point the job to a file that specifies hosts and number of slots for job processing.

For example, some applications (typically when benchmarking) run best with a very specific geometry. For repeatability (again, typically when benchmarking) you may want it to always run it on the same hosts, using the same number of slots.

The user-specified host file specifies a host and number of slots to use per task, resulting in a rank file.

The `-hostfile` option allows a user to submit a job, specifying the path of the user-specified host file:

```
bsub -hostfile "spec_host_file"
```

Any user can create a user-specified host file. It must be accessible by the user from the submission host. It lists one host per line. The format is as follows:

```
# This is a user-specified host file
<host_name1>  [<# slots>]
<host_name2>  [<# slots>]
<host_name1>  [<# slots>]
<host_name2>  [<# slots>]
<host_name3>  [<# slots>]
<host_name4>  [<# slots>]
```

The following rules apply to the user-specified host file:

- Insert comments starting with the `#` character.
- Specifying the number of slots for a host is optional. If no slot number is indicated, the default is 1.
- A host name can be either a host in a local cluster or a host leased-in from a remote cluster (*host_name@cluster_name*).
- A user-specified host file should contain hosts from the same cluster only.
- A host name can be entered with or without the domain name.
- Host names may be used multiple times and the order entered represents the placement of tasks. For example:

```
#first three tasks
host01                3
#fourth tasks
host02
#next three tasks
host03                3
```

The resulting rank file is made available to other applications (such as MPI).

The `LSB_DJOB_RANKFILE` environment variable is generated from the user-specified host file. If a job is not submitted with a user-specified host file then `LSB_DJOB_RANKFILE` points to the same file as `LSB_DJOB_HOSTFILE`.

The `esub` parameter `LSB_SUB4_HOST_FILE` reads and modifies the value of the `-hostfile` option.

Use `bsub -hostfile` (or `bmod -hostfile` for a pending job) to enter the location of a user-specified host file containing a list of hosts and slots on those hosts. The job will dispatch on the specified allocation once those resources become available.

Use **bmod -hostfile n** to remove the hostfile option from a job.

bjobs -l and **bhist -l** show the host allocation for a given job.

Use **-hostfile** together with **-l** or **-UF**, to view the user-specified host file content as well.

The following are restrictions on the usage of the **-hostfile** option:

- **bsub -hostfile** cannot be used with the **-ext** option.
- With **bsub** and **bmod**, the **-hostfile** option cannot be used with either the **-n** or **-m** option.
- With **bsub** and **bmod**, the **-hostfile** option cannot be combined with **-R** compound *res_req*.
- With **bjobs** and **bhist**, the **-hostfile** option must be used with either the **-l** or **-UF** option.

Smart memory limit enforcement

The new parameter **LSB_MEMLIMIT_ENF_CONTROL** in *lsf.conf* further refines the behavior of enforcing a job memory limit for a host. In the case that one or more jobs reach a specified memory limit for the host (both the host memory and swap utilization has reached a configurable threshold) at execution time, the worst offending job on the host will be killed. A job is selected as the worst offending job on that host if it has the most overuse of memory (actual memory usage minus memory limit of the job).

You also have the choice of killing all jobs exceeding the thresholds (not just the worst).

For a description of usage and restrictions on this parameter, see **LSB_MEMLIMIT_ENF_CONTROL**.

Note: This is an alternative to using cgroups memory enforcement.

Host-based memory and swap limit enforcement by Linux cgroup

LSF can now impose strict job-level host-based memory and swap limits on systems that support Linux cgroups. When **LSB_RESOURCE_ENFORCE="memory"** is set, memory and swap limits are calculated and enforced as a multiple of the number of tasks running on the execution host when memory and swap limits are specified for the job (at the job-level with **-M** and **-v**, or in *lsb.queues* or *lsb.applications* with **MEMLIMIT** and **SWAPLIMIT**).

The new **bsub -hl** option enables job-level (irrespective of the number of tasks) host-based memory and swap limit enforcement regardless of the number of tasks running on the execution host. **LSB_RESOURCE_ENFORCE="memory"** must be specified in *lsf.conf* for host-level memory and swap limit enforcement with the **-hl** option to take effect. If no memory or swap limit is specified for the job (the merged limit for the job, queue, and application profile, if specified), or **LSB_RESOURCE_ENFORCE="memory"** is not specified, a host-based memory limit is not set for the job. The **-hl** option only applies to memory and swap limits; it does not apply to any other resource usage limits.

See *Administering IBM Platform LSF* for more information about memory and swap resource usage limits, and memory enforcement based on Linux cgroup memory subsystem.

Change the default behavior of a job when it reaches the pre-execution retry threshold

When a job's pre-execution fails, the job will be requeued and tried again. When the pre-exec has failed a defined number of times (**LOCAL_MAX_PREEEXEC_RETRY** in `lsb.params`, `lsb.queues`, or `lsb.applications`) LSF suspends the job and places it in the PSUSP state. If this is a common occurrence, a large number of PSUSP jobs can quickly fill the system, leading to both usability issues and system degradation.

In this release, a pre-execution retry threshold is introduced so that a job exits once the pre-execution has failed a specified number of times. You can set **LOCAL_MAX_PREEEXEC_RETRY_ACTION** cluster-wide in `lsb.params`, at the queue level in `lsb.queues`, or at the application level in `lsb.applications`. The default behavior specified in `lsb.applications` overrides `lsb.queues`, and `lsb.queues` overrides the `lsb.params` configuration.

Set **LOCAL_MAX_PREEEXEC_RETRY_ACTION=EXIT** to have the job exit and to have LSF sets its status to EXIT. The job exits with the same exit code as the last pre-execution fail exit code.

MultiCluster considers TASKLIMIT on remote clusters before forwarding jobs

In the MultiCluster job forwarding model, the local cluster now considers the application profile or receive queue's **TASKLIMIT** setting on remote clusters before forwarding jobs. This reduces the number of forwarded jobs that stay pending before returning to the submission cluster due to the remote cluster's **TASKLIMIT** settings being unable to satisfy the job's task requirements. By considering the **TASKLIMIT** settings in the remote clusters, jobs are no longer forwarded to remote clusters that cannot run these jobs due to task requirements.

If the receive queue's **TASKLIMIT** definition in the remote cluster cannot satisfy the job's task requirements, the job is not forwarded to that remote queue. Likewise, if the application profile's **TASKLIMIT** definition in the remote cluster cannot satisfy the job's task requirements, the job is not forwarded to that cluster.

Enhancements to advance reservation

Two enhancements have been made to the advance reservation features:

- Advance reservation requests can be made on a unit of hosts by specifying the host requirements such as the number of hosts, the candidate host list, and/or the resource requirement for the candidate hosts. LSF creates the host-based advance reservation based on these requirements. Each reserved host is reserved in its entirety and cannot be reserved again nor can it be used by other jobs outside the advance reservation during the time it is dedicated to the advance reservation. If **MXJ** (in `lsb.hosts`) is undefined for a host, a host-based reservation reserves all CPUs on that host.

The command option `-unit` is introduced to **brsvadd** to indicate either slot or host for the advance reservation:

```
brsvadd -unit [slot | host]
```

If `-unit` is not specified for **brsvadd**, the advance reservation request will use the slot unit by default.

With either slot-based or host-based advance reservation, the request must specify the following:

- The number of slots or hosts to reserve, using the `-n` option.
- The list of candidate hosts, using `-m`, `-R`, or both.
- Users or user groups that have permission to use the advance reservation, using `-u`.
- A time period for the reservation, using either `-t` or `-b` and `-e` together.

The commands `brsvmod addhost` and `brsvmod rmhost` expand to include both slots or hosts, depending on the unit originally specified for the advance reservation through the command `brsvadd -unit`.

- An advance reservation request may specify a list of user and user group names. Each user or user group specified may run jobs for that advance reservation. Multiple users or user groups can be specified for an advance reservation using the **brsvmod** command:

- `brsvmod -u "user_name | user_group"` replaces an advance reservation's list of users and user groups.

If the advance reservation was created with the `-g` option, **brsvmod** cannot switch the advance reservation type from group to user. In this case, use `brsvmod -u` can be used to replace the entire list of users and user groups.

Note: The `-g` option is obsolete after the 9.1.3 release.

- `brsvmod adduser -u "user_name | user_group"` adds users and user groups to the advance reservation.
- `brsvmod rmuser -u "user_name | user_group"` removes users and user groups from the advance reservation.
- The `"job_slots"` variable has been changed to `"number_unit"` for the **addhost** and **rmhost** subcommands using the `-n` option. This is to agree with the expansion of **brsvadd** to cover both slot and host-based reservations.

The output of **brsvs** is expanded to show:

- Whether the advance reservation was created with the user (`-u`) or group (`-g`) option, as shown under the TYPE heading. Possible values are "user", "sys", and "group" (deprecated with `-g` after LSF 9.1.3).
- The list of users or user groups specified for the advance reservation, under the USER heading.
- The Resource Unit (Slot or Host) specified for an advance reservation (with the `-l` option).

If you downgrade your installation from 9.1.3 to 9.1.2 or lower:

- A host-based advance reservation will become a slot-based reservation in which each host is entirely reserved.
- Reservations that specify multiple users and/or groups will have only 1 user, being the first name in the original user name list ordered alphabetically. A list of user group names specified by `-u` is not allowed in previous versions of LSF. After downgrading, reservations created with LSF 9.1.3 will not be valid and cannot be used to run jobs. You can use the **brsvmod** command to change such reservations.

Control the propagation of job submission environment variables

When using **bsub** and **tssub** to submit jobs, you can use the `-env` option to control the propagation of job submission environment variables to the execution hosts:

```
-env "none" | "all [ , ~var_name[, ~var_name] ...] [ , var_name=var_value[,  
var_name=var_value] ...]" | "var_name[=var_value][, var_name[=var_value] ...]"
```

Specify a comma-separated list of environment variables. Controls the propagation of the specified job submission environment variables to the execution hosts.

- Specify `none` to submit jobs that do not propagate any environment variables.
- Specify the variable name without a value to propagate the environment variable using its existing specified value.
- Specify the variable name with a value to propagate the environment variable with the specified value to overwrite the existing specified value. The specified value may either be a new value or quote the value of an existing environment variable. Job packs do not allow you to quote the value of an existing environment variable.

For example,

- In UNIX, `fullpath=/tmp/:$filename` appends `/tmp/` to the beginning of the `filename` environment variable and assigns this new value to the `fullpath` environment variable. Use a colon (`:`) to separate multiple environment variables.
- In Windows, `fullpath=\Temp\;%filename%` appends `\Temp\` to the beginning of the `filename` environment variable and assigns this new value to the `fullpath` environment variable. Use a semicolon (`;`) to separate multiple environment variables.
- Specify `all` at the beginning of the list to propagate all existing submission environment variables to the execution hosts. You may also assign values to specific environment variables.
For example, `-env "all, var1=value1, var2=value2"` submits jobs with all the environment variables, but with the specified values for the `var1` and `var2` environment variables.
- When using the `all` keyword, add `~` to the beginning of the variable name to prevent the environment variable from being propagated to the execution hosts.

The environment variable names cannot be "none" or "all".

The environment variable names cannot contain the following symbols: comma (`,`), `"~"`, `"="`, double quotation mark (`"`) and single quotation mark (`'`).

The variable value can contain a comma (`,`) and `"~"`, but if it contains a comma, you must enclose the variable value in single quotation marks.

An **esub** can change the `-env` environment variables by writing them to the file specified by the **LSB_SUB_MODIFY_FILE** environment variable. If the **LSB_SUB_MODIFY_ENVFILE** environment variable is also specified and the file specified by this environment variable contains the same environment variables, the environment variables in **LSF_SUB_MODIFY_FILE** take effect.

When `-env` is not specified with **bsub**, the default value is `-env "all"` (that is, all environment variables are submitted with the default values).

The entire argument for the `-env` option may contain a maximum of 4094 characters for UNIX and Linux, or up to 255 characters for Windows.

If `-env` conflicts with `-L`, the value of `-L` takes effect.

The following environment variables are not propagated to execution hosts because they are only used in the submission host and are not used in the execution hosts:

- HOME, LS_JOBPID, LSB_ACCT_MAP, LSB_EXIT_PRE_ABORT, LSB_EXIT_QUEUE, LSB_EVENT_ATTRIB, LSB_HOSTS, LSB_INTERACTIVE, LSB_INTERACTIVE_SSH, LSB_INTERACTIVE_TTY, LSB_JOBFILENAME, LSB_JOBGROUP, LSB_JOBID, LSB_JOBNAME, LSB_JOB_STARTER, LSB_QUEUE, LSB_RESTART, LSB_TRAPSIGS, LSB_XJOB_SSH, LSF_VERSION, PWD, USER, VIRTUAL_HOSTNAME, and all variables with starting with LSB_SUB_
- Environment variables about non-interactive jobs. For example: TERM, TERMCAP
- Windows-specific environment variables. For example: COMPUTERNAME, COMSPEC, NTRESKIT, OS2LIBPATH, PROCESSOR_ARCHITECTURE, PROCESSOR_IDENTIFIER, PROCESSOR_LEVEL, PROCESSOR_REVISION, SYSTEMDRIVE, SYSTEMROOT, TEMP, TMP

The following environment variables do not take effect on the execution hosts: LSB_DEFAULTPROJECT, LSB_DEFAULT_JOBGROUP, LSB_TSJOB_ENVNAME, LSB_TSJOB_PASSWD, LSF_DISPLAY_ALL_TSC, LSF_JOB_SECURITY_LABEL, LSB_DEFAULT_USERGROUP, LSB_DEFAULT_RESREQ, LSB_DEFAULTQUEUE, BSUB_CHK_RESREQ, LSB_UNIXGROUP, LSB_JOB_CWD

View job file names with job ID and array index values

When submitting jobs with specified input, output, and error file names (using **bsub** `-i`, `-is`, `-o`, `-oo`, `-e`, and `-eo` options), you can use the special characters `%J` and `%I` in the name of the files. `%J` is replaced by the job ID. `%I` is replaced by the index of the job in the array, if the job is a member of an array, or by 0 (zero) if the job is not a member of an array. When viewing job information, **bjobs** `-o`, `-l`, or `-UF` now replaces `%J` with the job ID and `%I` with the array index when displaying job file names. Previously, **bjobs** `-o`, `-l`, or `-UF` displayed these file names with `%J` and `%I` without resolving the job ID and array index values.

Documentation and online help enhancements to **bjobs** and **bsub**

The documentation and online help for **bjobs** and **bsub** are now reorganized and expanded. The **bjobs** and **bsub** command options are grouped into categories, which describes the general goal or function of the command option.

In the *IBM Platform LSF Command Reference* documentation, the **bjobs** and **bsub** sections now list the categories, followed by the options, listed in alphabetical order. Each option lists the categories to which it belongs and includes a detailed synopsis of the command. Any conflicts that the option has with other options are also listed (that is, options that cannot be used together).

The online help in the command line for **bjobs** and **bsub** is organized by categories and allows you to view help topics for specific options in addition to viewing the entire man page for the command. To view the online help, run **bjobs** or **bsub** with the `-h` (or `-help`) option. This provides a brief description of the command and lists the categories and options that belong to the command. To view a brief description

of all options, run `-h all` (or `-help all`). To view more details on the command, run `-h description` (or `-help description`). To view more information on the categories and options (in increasing detail), run `-h` (or `-help`) with the name of the category or the option:

```
bjobs -h[elp] [all] [description] [category_name ...] [-option_name ...]
```

```
bsub -h[elp] [all] [description] [category_name ...] [-option_name ...]
```

If you list multiple categories and options, the online help displays each entry in the order in which you specified the categories and options.

For example,

- To view a brief description of the **bjobs** command, run **bjobs -h**. The description includes a list of categories (with a brief description of each category) and the options belonging to each category.
- To view more details of the **bjobs** command, run **bjobs -h description**.
- To a brief description of all **bjobs** options, run **bjobs -h all**.
- To view a description of the **bjobs** filter category, run **bjobs -h filter**. The description includes a list of options with a brief description of each option.
- To view a detailed description of the **bjobs -app** option, run **bjobs -h -app**. The description includes the categories to which the option belongs, a detailed synopsis/usage of the option, and any conflicts the option has with other options.

LSF support for Gold v.2.2

Gold is a dynamic accounting system that tracks and manages resource usage in a cluster. LSF is integrated with Gold v2.2. The LSF integration allows dynamic accounting in Gold. The following Gold features are supported:

- Job quotations at the time of job submission
- Job reservations at the start time of jobs
- Job charges when jobs are completed

Gold v2.2 (or newer) is supported on Linux and UNIX. Complete the steps in `LSF_INSTALLDIR/9.1/misc/examples/gold/readme.txt` to install and configure the Gold integration in LSF

New and changed commands, options, and output

The following command options and output are new or changed for LSF 9.1.3

bacct

- A new termination reason, `TERM_ORPHAN_SYSTEM`, shows that an orphan job was automatically terminated by LSF.
- When an allocation shrinks, **bacct** shows
Release allocation on `<num_hosts>` Hosts/Processors `<host_list>` by user or administrator `<user_name>`
Resize notification accepted;
- Output includes number of slots and host names that a job has been allocated to, based on number of tasks in the job:
`bacct -l -aff 6`

Accounting information about jobs that are:

- submitted by all users.
- accounted on all projects.
- completed normally or exited
- executed on all hosts.
- submitted to all queues.
- accounted on all service classes.

```
-----
Job <6>, User <user1>, Project <default>, Status <DONE>, Queue <normal>, Command <myjob>
Thu Feb 14 14:13:46: Submitted from host <hostA>, CWD <${HOME}>;
Thu Feb 14 14:15:07: Dispatched <num_tasks> Task(s) on Host(s) <host_list>,
Allocated <num_slots> Slot(s) on Host(s) <host_list>;
Effective RES_REQ <select[type == local] order[r15s:pg]
rusage[mem=100.00] span[hosts=1] affinity[core(1,same=
socket,exclusive=(socket,injob))*1:cpubind=socket:membind
=localonly:distribute=pack] >;
Thu Feb 14 14:16:47: Completed <done>.
```

bapp

- Added -alloc option. Shows counters for slots in RUN, SSUSP, and USUSP. The slot allocation will be different depending on whether the job is an exclusive job or not.
- Changes to output. The following fields have changed from a slot-based to a task-based concept: NJOBS, PEND, RUN, SSUSP, USUSP, and RSV.

bhist

- If a job was submitted or modified with a -hostfile option, to point to a user-specified host file, **bhist -l** will show the show the user-specified host file path. **bhist -l -hostfile** will also show the user-specified host file contents.
- For JOB_RESIZE_NOTIFY_START event, **bhist** displays:

```
Added <num_tasks> tasks on host <host_list>, <num_slots> additional slots
allocated on <host_list>
```

- For JOB_RESIZE_RELEASE event, **bhist** displays

```
Release allocation on <num_hosts> Hosts/Processors <host_list> by user or
administrator <user_name>
Resize notification accepted;
```

- Output includes number of tasks in the job submitted and number of slots and host names that a job has been allocated to, based on number of tasks in the job:

```
bhist -l 749
```

```
Job <749>, User <user1>;, Project <default>, Command <my_pe_job>
```

```
Mon Jun  4 04:36:12: Submitted from host <hostB>, to Queue <priority>,
CWD <${HOME}>, 2 Task(s), Requested
Network <type=sn_all:protocol=mpi:mode=US:usage=
shared:instance=1>
```

```
Mon Jun  4 04:36:15: Dispatched <num_tasks> Task(s) on Host(s) <host_list>,
Allocated <num_slots> Slot(s) on Host(s) <host_list>;
Effective RES_REQ <select[type == local] rusage[nt1=1.00] >,
PE Network ID <1111111> <2222222> used <1> window(s)
```

```
Mon Jun  4 04:36:17: Starting (Pid 21006);
```

bhosts

- Added -alloc option. Shows counters for slots in RUN, SSUSP, and USUSP. The slot allocation will be different depending on whether the job is an exclusive job or not.

- Changes to Host-based default output. The following fields have changed from a slot-based to a task-based concept: NJOBS, RUN, SSUSP, USUSP, RSV.

bjdepinfo

- This command can now show if a job dependency condition was not satisfied.

bjobs

- If a job was submitted or modified with a `-hostfile` option, **bjobs -l** or **bjobs -UF** will show the user-specified host file path. **bjobs -l -hostfile** or **bjobs -UF -hostfile** will show the user-specified host file contents.
- The **bjobs -o** option lets you specify the following new **bjob** fields:
 - `immediate_orphan_term` indicates that an orphan job was terminated immediately and automatically.
 - `host_file` displays the user-specified host file path used for a job.
 - `user_group` (alias `ugroup`) indicates the user group to which the jobs are associated (submitted with **bsub -G** for the specified user group).
- Behavior change for **bjobs -l**: Predicted start time for PEND reserve job will not be shown with **bjobs -l**. LSF does not calculate predicted start time for PEND reserve job if no back fill queue is configured in the system. In that case, resource reservation for PEND jobs works as normal, and no predicted start time is calculated.
- Behavior change for **bjobs -help**: Displays the description of the specified category, command option, or sub-option to stdout and exits. You can now abbreviate the `-help` option to `-h`.

Run **bjobs -h** (or **bjobs -help**) without a command option or category name to display the **bjobs** command description.

- The output of **bjobs -l** has changed:
 - Number of Processors Requested has changed to task concept, displaying number of Tasks.
 - For started jobs, slot allocation is shown after the number of Tasks started on the hosts.

For example:

```
bjobs -l 6

Job <6>, User <user1>, Project <default>, Status <RUN>, Queue <normal>, Command <myjob1>
Thu Feb 14 14:13:46: Submitted from host <hostA>, CWD <${HOME}>, 6 Tasks;
Thu Feb 14 14:15:07: Started 6 Task(s) on Host(s) <hostA> <hostA> <hostA> <hostA> <hostA> <hostA>, Allocated 6 Slots on Hosts <hostA> <hostA> <hostA> <hostA> <hostA>, Execution Home </home/user1>, Execution CWD </home/user1>;
```

bmod

- The `-hostfile` option allows a user to modify a PEND job with a user-specified host file. A user-specified host file contains specific hosts and slots that a user wants to use for a job:


```
bmod -hostfile "host_alloc_file" <job_id>
```

 To remove a user-specified host file specified for a PEND job, use the `-hostfilen` option:


```
bmod -hostfilen <job_id>
```
- The `-hl` option enables per-job host-based memory and swap limit enforcement on hosts that support Linux cgroups. The `-hln` option disables host-based memory and swap limit enforcement. `-hl` and `-hln` only apply to pending jobs.

LSB_RESOURCE_ENFORCE="memory" must be specified in `lsf.conf` for host-based memory and swap limit enforcement with the `-hl` option to take effect. If no memory or swap limit is specified for the job (the merged limit for the job, queue, and application profile, if specified), or **LSB_RESOURCE_ENFORCE="memory"** is not specified, a host-based memory limit is not set for the job.

- The `-ti` option of `-w` enables immediate automatic orphan job termination at the job level. The `-tin` option cancels the `-ti` option of a submitted dependent job, in which case the cluster-level configuration takes precedence.
- The **bmod -n** command has changed from `-n num_processors` to `-n num_tasks`.

bqueues

- Added `-alloc` option. Shows counters for slots in RUN, SSUSP, and USUSP. The slot allocation will be different depending on whether the job is an exclusive job or not.
- Changes to output. The following fields have changed from a slot-based to a task-based concept: NJOBS, PEND, RUN, SUSP, SSUSP, USUSP, and RSV.

brestart

- The `-ti` option allows users to indicate that a job is eligible for automatic and immediate termination by the system as soon as the job is found to be an orphan, without waiting for the grace period to expire.

brsvadd

- The `new-unit [slot | host]` option specifies whether an advance reservation is for a number of slots or hosts. If `-unit` is not specified for **brsvadd**, the advance reservation request will use the slot unit by default.

The following options are required for **brsvadd**, whether using the slot or host unit:

- The number of slots or hosts to reserve, using the `-n` option.
 - The list of candidate hosts, using `-m`, `-R`, or both.
 - Users or user groups that have permission to use the advance reservation, using `-u`.
 - A time period for the reservation, using either `-t` or `-b` and `-e` together.
- The option `-n` has been changed to specify either *job_slots* or *number_hosts*. The number of either job slots or hosts (specified by `-unit`) to reserve. For a slot-based advance reservation (`brsvadd -unit slot`), `-n` specifies the total number of job slots to reserve. For host-based advance reservation `brsvadd -unit host`, `-n` specifies the total number of hosts to reserve.
 - The option `-m` has been changed so that the number of slots specified by `-n <job_slots>` or hosts specified by `-n <number_hosts>` must be less than or equal to the actual number of hosts specified by `-m`.
 - The `-u` option has been expanded to include multiple users and user groups, in combination, if desired.
 - The `-g` option will be obsolete after LSF 9.1.3.

brsvmod

- The `-u "user_name... | user_group ..."` option has been changed so that it replaces the list of users or groups who are able to submit jobs to a reservation.
- The **adduser** subcommand has been added to add users and user groups to an advance reservation:
`adduser -u "user_name ... | user_group ..."] reservation_ID`

- The **rmuser** subcommand has been added to remove users and user groups from an advance reservation.
`rmuser -u "user_name ... | user_group ..."] reservation_ID`
- The option **-n** has been changed to specify *number_unit*. This option now changes the number of either job slots or hosts to reserve (based on the unit specified by `brsvadd -unit slot | host`. *number_unit* must be less than or equal to the actual number of slots for the hosts selected by **-m** or **-R** for the reservation.
- The option **-m** has been changed to modify the list of hosts for which job slots or number of hosts specified with **-n** are reserved.
- The **-g** option will be obsolete after LSF 9.1.3.

brsvs

- The output of **brsvs** is expanded to show:
 - whether the advance reservation was created with the user (**-u**) or group (**-g**) option, as shown under the TYPE heading.
 - the list of users or user groups specified for the advance reservation, under the USER heading.
 - (with the **-l** option), the Resource Unit (Slot or Host) specified for an advance reservation.

bslots

- Behavior change for **bslots**: LSF does not calculate predicted start times for PEND reserve jobs if no backfill queue is configured in the systemc. In that case, the resource reservation for PEND jobs works as normal, but no predicted start time is calculated, and **bslots** does not show the backfill window.

bsub

- The **-ti** suboption of **-w** enables automatic orphan job termination at the job level. If configured, the cluster-level orphan job termination grace period is ignored and the job is terminated as soon as it is found to be an orphan. This option is independent of the cluster-level configuration. If the LSF administrator did not enable **ORPHAN_JOB_TERM_GRACE_PERIOD** at the cluster level, you can still use automatic orphan job termination on a per-job basis.
- The **-hostfile** option allows a user to submit a job with a user-specified host file. A user-specified host file contains specific hosts and slots that a user wants to use for a job. The user-specified host file specifies the order in which to launch tasks, ranking the slots specified in the file. This command specifies the path of the user-specified host file:
`bsub -hostfile "host_alloc_file" ./a.out`
- The **-hl** option enables job-level host-based memory and swap limit enforcement on systems that support Linux cgroups. When **-hl** is specified, a memory limit specified at the job level by **-M** or by **MEMLIMIT** in `lsb.queues` or `lsb.applications` is enforced by the Linux cgroup subsystem on a per-job basis on each host. Similarly, a swap limit specified at the job level by **-v** or by **SWAPLIMIT** in `lsb.queues` or `lsb.applications` is enforced by the Linux cgroup subsystem on a per-job basis on each host. Host-based memory and swap limits are enforced regardless of the number of tasks running on the execution host. The **-hl** option only applies to memory and swap limits; it does not apply to any other resource usage limits.
LSB_RESOURCE_ENFORCE="memory" must be specified in `lsf.conf` for host-based memory and swap limit enforcement with the **-hl** option to take effect. If no memory or swap limit is specified for the job (the merged limit for the job,

queue, and application profile, if specified), or `LSB_RESOURCE_ENFORCE="memory"` is not specified, a host-based memory limit is not set for the job.

When `LSB_RESOURCE_ENFORCE="memory"` is configured in `lsf.conf`, and memory and swap limits are specified for the job, but `-hl` is *not* specified, memory and swap limits are calculated and enforced as a multiple of the number of tasks running on the execution host.

- The `-env` option allows you to control the propagation of the specified job submission environment variables to the execution hosts. Specify a comma-separated list of environment variables to propagate to the execution hosts, or add `~` to the beginning of the variable name to block the environment variable from being propagated. The `none` keyword prevents all environment variables from being propagated, while the `all` keyword allows all environment variables to be propagated with their default values.
- The `bsub -n` command has changed from `-n min_processors[,max_processors]` to `-n min_tasks[,max_tasks]`
Submits a parallel job and specifies the number of tasks in the job. The number of tasks is used to allocate a number of slots for the job. Usually, the number of slots assigned to a job will equal the number of tasks specified. For example, one task will be allocated with one slot. (Some slots/processors may be on the same multiprocessor host).
- Behavior change for `bsub -help`: Displays the description of the specified category, command option, or sub-option to stdout and exits. You can now abbreviate the `-help` option to `-h`.
Run `bsub -h` (or `bsobs -help`) without a command option or category name to display the `bsub` command description.

busers

- Added `-alloc` option. Shows counters for slots in RUN, SSUSP, and USUSP. The slot allocation will be different depending on whether the job is an exclusive job or not.
- Changes to output. The following fields have changed from a slot-based to a task-based concept: NJOBS, PEND, RUN, SSUSP, USUSP, RSV.

lsmake

- The option `--no-block-shell-mode` has been added to allow `lsmake` to build customized Android 4.3 code. This option allows `lsmake` to perform child "shell" tasks without blocking mode. Without this parameter, blocking mode is used, making the build for Android 4.3 take a long time.

tssub

- The `-env` option allows you to control the propagation of the specified job submission environment variables to the execution hosts. Specify a comma-separated list of environment variables to propagate to the execution hosts, or add `~` to the beginning of the variable name to block the environment variable from being propagated. The `none` keyword prevents all environment variables from being propagated, while the `all` keyword allows all environment variables to be propagated with their default values.

New and changed configuration parameters and environment variables

The following configuration parameters and environment variables are new or changed for LSF 9.1.3

lsb.applications

- **PROCLIMIT** has been replaced by **TASKLIMIT**. It now represents the maximum number of tasks that can be allocated to a job. For parallel jobs, the maximum number of tasks that can be allocated to the job.
- **JOB_SIZE_LIST**: Defines a list of job sizes that are allowed in the specified application profile. The default job size is assigned automatically if there is no requested job size.
`JOB_SIZE_LIST=default_size [size ...]`
- When **MEMLIMIT** is defined and the job is submitted with `-h1`, memory limits are enforced on systems that support Linux cgroups for on a per-job and per-host basis, regardless of the number of tasks running on the execution host. **LSB_RESOURCE_ENFORCE="memory"** must be specified in `lsf.conf` for host-based memory limit enforcement with the `-h1` option to take effect.
- When **SWAPLIMIT** is defined and the job is submitted with `-h1`, swap limits are enforced on systems that support Linux cgroups for on a per-job and per-host basis, regardless of the number of tasks running on the execution host. **LSB_RESOURCE_ENFORCE="memory"** must be specified in `lsf.conf` for host-based swap limit enforcement with the `-h1` option to take effect.
- **LOCAL_MAX_PREEEXEC_RETRY_ACTION**: Defines the action to take on a job when the number of times to attempt its pre-execution command on the local cluster (**LOCAL_MAX_PREEEXEC_RETRY**) is reached.
`LOCAL_MAX_PREEEXEC_RETRY_ACTION=SUSPEND | EXIT`
 - If set to **SUSPEND**, the job is suspended and its status is set to **PSUSP**.
 - If set to **EXIT**, the job status is set to **EXIT** and the exit code is the same as the last pre-execution fail exit code.
- In the MultiCluster job forwarding model, the local cluster now considers **TASKLIMIT** on remote clusters before forwarding jobs. If the **TASKLIMIT** in the remote cluster cannot satisfy the job's processor requirements for an application profile, the job is not forwarded to that cluster.

lsb.params

- **ORPHAN_JOB_TERM_GRACE_PERIOD**: If defined, enables automatic orphan job termination at the cluster level which applies to all dependent jobs; otherwise it is disabled. This parameter is also used to define a cluster-wide termination grace period to tell LSF how long to wait before killing orphan jobs. Once configured, automatic orphan job termination applies to all dependent jobs in the cluster.
 - **ORPHAN_JOB_TERM_GRACE_PERIOD = 0**: Automatic orphan job termination is enabled in the cluster but no termination grace period is defined. A dependent job can be terminated as soon as it is found to be an orphan.
 - **ORPHAN_JOB_TERM_GRACE_PERIOD > 0**: Automatic orphan job termination is enabled and the termination grace period is set to the specified number of seconds. This is the minimum time LSF will wait before terminating an orphan job. In a multi-level job dependency tree, the grace period is not repeated at each level, and all direct and indirect orphans of the parent job can be terminated by LSF automatically after the grace period has expired.
`ORPHAN_JOB_TERM_GRACE_PERIOD=seconds`
- **LOCAL_MAX_PREEEXEC_RETRY_ACTION**: Defines the default behavior of a job when it reaches the maximum number of times to attempt its pre-execution command on the local cluster (**LOCAL_MAX_PREEEXEC_RETRY**).
`LOCAL_MAX_PREEEXEC_RETRY_ACTION=SUSPEND | EXIT`

- If set to `SUSPEND`, the job is suspended and its status is set to `PSUSP`. This is the default action.
- If set to `EXIT`, the job status is set to `EXIT` and the exit code is the same as the last pre-execution fail exit code.

lsb.serviceclasses

- **EGO_RESOURCE_GROUP**: For EGO-enabled SLA service classes. A resource group or space-separated list of resource groups from which hosts are allocated to the SLA. List must be a subset of or equal to the resource groups allocated to the consumer defined by the **CONSUMER** entry. Guarantee SLAs (with `GOALS=[GUARANTEE]`) cannot have **EGO_RESOURCE_GROUP** set. If defined, it will be ignored. Default is undefined. In this case, **vemkd** determines which resource groups to allocate slots to LSF.

After changing this parameter, running jobs using the allocation may be re-queued.

```
EGO_RESOURCE_GROUP=mygroup1 mygroup4 mygroup5
```

lsb.queues

- **PROCLIMIT** has been replaced by **TASKLIMIT**. It now represents the maximum number of tasks that can be allocated to a job. For parallel jobs, the maximum number of tasks that can be allocated to the job.
- **JOB_SIZE_LIST**: Defines a list of job sizes that are allowed in the specified queue. The default job size is assigned automatically if there is no requested job slot size.

```
JOB_SIZE_LIST=default_size [size ...]
```

- When **MEMLIMIT** is defined and the job is submitted with `-h1`, memory limits are enforced on systems that support Linux cgroups for on a per-job and per-host basis, regardless of the number of tasks running on the execution host. **LSB_RESOURCE_ENFORCE="memory"** must be specified in `lsf.conf` for host-based memory limit enforcement with the `-h1` option to take effect.
- When **SWAPLIMIT** is defined and the job is submitted with `-h1`, swap limits are enforced on systems that support Linux cgroups for on a per-job and per-host basis, regardless of the number of tasks running on the execution host. **LSB_RESOURCE_ENFORCE="memory"** must be specified in `lsf.conf` for host-based swap limit enforcement with the `-h1` option to take effect.
- **LOCAL_MAX_PREEEXEC_RETRY_ACTION**: Defines the action to take on a job when the number of times to attempt its pre-execution command on the local cluster (**LOCAL_MAX_PREEEXEC_RETRY**) is reached.


```
LOCAL_MAX_PREEEXEC_RETRY_ACTION=SUSPEND | EXIT
```

 - If set to `SUSPEND`, the job is suspended and its status is set to `PSUSP`.
 - If set to `EXIT`, the job status is set to `EXIT` and the exit code is the same as the last pre-execution fail exit code.
- In the MultiCluster job forwarding model, the local cluster considers the receive queue's **TASKLIMIT** on remote clusters before forwarding jobs. If the receive queue's **TASKLIMIT** definition in the remote cluster cannot satisfy the job's processor requirements for a remote queue, the job is not forwarded to that remote queue in the cluster.

lsb.resources

- The `PER_SLOT` value in the `ReservationUsage` section has been changed to `PER_TASK` with the change to a task concept for job resource allocation.

lsf.conf

- **LSB_ENABLE_HPC_ALLOCATION**: When set to $Y|y$, this parameter changes concept of the required number of slots for a job to the required number of tasks for a job. The specified numbers of tasks (using **bsub**), will be the number of tasks to launch on execution hosts. The allocated slots will change to all slots on the allocated execution hosts for an exclusive job in order to reflect the actual slot allocation.

For new installations of LSF, **LSB_ENABLE_HPC_ALLOCATION** is set to Y automatically.

`LSB_ENABLE_HPC_ALLOCATION=Y|y|N|n`

- **LSB_MEMLIMIT_ENF_CONTROL**: This parameter further refines the behavior of enforcing a job memory limit. In the case that one or more jobs reach a specified memory limit (both the host memory and swap utilization has reached a configurable threshold) at execution time, the worst offending job will be killed. A job is selected as the worst offending job on that host if it has the most overuse of memory (actual memory usage minus memory limit of the job). You also have the choice of killing all jobs exceeding the thresholds (not just the worst).

`LSB_MEMLIMIT_ENF_CONTROL=<Memory Threshold>:<Swap Threshold>:<Check Interval>:[all]`

The following describes usage and restrictions on this parameter.

- `<Memory Threshold>`: (Used memory size/maximum memory size)
A threshold indicating the maximum limit for the ratio of used memory size to maximum memory size on the host. The threshold represents a percentage and must be an integer between 1 and 100.
 - `<Swap Threshold>`: (Used swap size/maximum swap size)
A threshold indicating the maximum limit for the ratio of used swap memory size to maximum swap memory size on the host. The threshold represents a percentage and must be an integer between 1 and 100.
 - `<Check Interval>`: The value, in seconds, specifying the length of time that the host memory and swap memory usage will not be checked during the nearest two checking cycles. The value must be an integer greater than or equal to the value of **SBD_SLEEP_TIME**.
 - The keyword `:all` can be used to terminate all single host jobs that exceed the memory limit when the host threshold is reached. If not used, only the worst offending job is killed.
 - If the cgroup memory enforcement feature is enabled (**LSB_RESOURCE_ENFORCE** includes the keyword "memory"), **LSB_MEMLIMIT_ENF_CONTROL** is ignored.
 - The host will be considered to reach the threshold when both Memory Threshold and Swap Threshold are reached.
 - **LSB_MEMLIMIT_ENF_CONTROL** does not have any effect on jobs running across multiple hosts. They will be terminated if they are over the memory limit regardless of usage on the execution host.
 - On some operating systems, when the used memory equals the total memory, the OS may kill some processes. In this case, the job exceeding the memory limit may be killed by the OS not an LSF memory enforcement policy.
In this case, the exit reason of the job will indicate "killed by external signal".
- **LSB_BJOBS_FORMAT** now lets you specify the `user_group` field (alias `ugroup`), which indicates the user group to which the jobs are associated (submitted with **bsub -G** for the specified user group).

Environment variables

- When a job is submitted with a user-specified host file, the `LSB_DJOB_RANKFILE` environment variable is generated from the user-specified host file. If a job is not submitted with a user-specified host file then `LSB_DJOB_RANKFILE` points to the same file as `LSB_DJOB_HOSTFILE`.
`LSB_DJOB_RANKFILE=file_path`
- The `esub` environment variable `LSB_SUB_MEM_SWAP_HOST_LIMIT` controls host-level memory and swap limit enforcement on execution hosts that support the Linux `cgroup` subsystem. The value is `Y` or `SUB_RESET`.
`LSB_RESOURCE_ENFORCE="memory"` must be specified in `lsf.conf` and a memory or swap limit must be specified for the job for host-based memory and swap limit enforcement to take effect.
- The `esub` environment variable `LSF_SUB4_SUB_ENV_VARS` is used to modify the `bsub -env` and `tssub -env` parameters.
- `LSB_PROJECT_NAME` can be used to set the project name. This enables the use of project names for accounting purposes inside third party tools that launch jobs under LSF using environment variables.
`LSB_PROJECT_NAME=project_name`

New and changed accounting and job event fields

The following job event fields are added or changed for LSF 9.1.3.

lsb.acct

- **JOB_FINISH:** The fields `numAllocSlots(%d)` and `allocSlots(%s)` have been added. `numAllocSlots(%d)` is the Number of allocated slots and `allocSlots(%s)` is the list of execution hosts where the slots are allocated.
- **JOB_RESIZE:** The fields `numAllocSlots(%d)`, `allocSlots(%s)`, `numResizeSlots(%d)`, and `resizeSlots(%s)` have been added. `numAllocSlots(%d)` is the Number of allocated slots and `allocSlots(%s)` is the list of execution hosts where the slots are allocated. `numResizeSlots(%d)` is the number of slots allocated for executing a resize and `resizeSlots(%s)` is a list of execution host names where slots are allocated for resizing.

lsb.events

- **JOB_START:** The fields `numAllocSlots(%d)` and `allocSlots(%s)` have been added. `numAllocSlots(%d)` is the Number of allocated slots and `allocSlots(%s)` is the list of execution hosts where the slots are allocated.
- **JOB_RESIZE_NOTIFY_START** and **JOB_RESIZE_RELEASE:** The fields `numResizeSlots(%d)` and `resizeSlots(%s)` have been added. `numResizeSlots(%d)` is the number of slots allocated for executing a resize and `resizeSlots(%s)` is a list of execution host names where slots are allocated for resizing.

Known issues

- LSF 9.1.3 has updated the `libstdc++.so.5` system library that is linked to Platform EGO binaries like **egosh** and **egoconfig**. EGO commands are now linked with `libstdc++.so.6`. You must install `libstdc++.so.6` on older systems that may not have this library. After you upgrade to LSF 9.1.3, Platform EGO commands fail without this library installed. For example, running **egosh** gives the following error:

```
egoshegosh: error while loading shared libraries: libstdc++.so.6:  
cannot open shared object file: No such file or directory
```

- Compatibility problem with parallel jobs using both a 9.1.1 and 9.1.3 execution host. The job will exit unsuccessfully.

After the 9.1.1 release of LSF, logic to handle the case that a task exits slowly on other execution nodes when LSF crashes on the first execution node was introduced. The **LSF_RES_ALIVE_TIMEOUT** parameter was introduced to control if those tasks exit directly on nodes other than the first node. LSF res report task usage is sent to the first node and it waits for the first node to reply. If the timeout exceeds the **LSF_RES_ALIVE_TIMEOUT** setting, LSF res on an execution node other than the first knows that the LSF daemons have crashed on the first node. LSF res exits directly on the non-execution node.

If LSF daemons on the first execution node are version 9.1.1, they do not include the **LSF_RES_ALIVE_TIMEOUT** parameter. Therefore, if 9.1.3 is on a subsequent execution node, it cannot always receive a reply. If LSF daemons on the first execution node detect that some tasks exited, they also exit and the entire job fails to run.

Solution: To run a parallel job in a mixed LSF 9.1.1 and 9.1.3 environment, set **LSF_RES_ALIVE_TIMEOUT=0** in job environment variables when submitting the job. The logic will be disabled.

- When LSF 9.1.3 is installed on RHEL Power8 chown fails to change ownership from root on NFSv4 file systems.

This is a known issue in RHEL NFSv4 (see <https://access.redhat.com/site/solutions/130783>).

On an NFS client with NFSv4 mount, an error may occur when attempting to chown a file in the mount directory: `chown: changing ownership of `filename': Invalid argument`

NFSv3, an NFS client, passes the UID number to chown and the NFS server accepts it. NFSv4 passes identities in the form of `<username>@<id-map-domainname>` to make sure client and server are working on the same domain. Chown fails when:

- The username is known to the client but not known to the server.
- The idmapd domain name is set differently on the client vs server.

This issue can be fixed by:

- Insuring that the NFS server and client are configured with the same domain name (`/etc/idmapd.conf`) and both have knowledge of the user accounts.
- If you cannot ensure that both NFS server and client have the same user account knowledge, disable the idmap feature:
 - Set the following parameter for the kernel NFS module:


```
nfs.nfs4_disable_idmapping=1
```
 - Or, it can be set to take effect during slightly later during boot with:


```
echo "options nfs nfs4_disable_idmapping=1" > /etc/modprobe.d/99-nfs.conf
```
 - Or, it can also be set on-the-fly with:


```
echo 1 > /sys/module/nfs/parameters/nfs4_disable_idmapping
```

 and remount the NFSv4 entry point.
- Compute Unit does not work with leased-in hosts. mbschd does not update with the leased-in hosts when they are assigned to a Compute Unit.

Also, when a Compute Unit member is a host group, then that host group cannot contain a wildcard. If you try to configure that case, LSF logs a warning and ignores the Compute Unit.

- Parallel restart cannot be used in Solaris Zone environment. **mbatchd** cannot start after initiating an **mbatchd** parallel restart on Sparc Solaris.
- Installer does not copy the JRE to LSF_TOP/9.1/install/1ap after installation when the host already has the JRE environment set. But if the JRE environment is no longer available, the LSF patch cannot be successfully applied.
- To use egosh.exe on Windows 2003 x86_64, "Microsoft Visual C++ 2005 SP1 Redistributable Package (x64)" (available at <http://support.microsoft.com>) must be installed first.
- Issue lim socket leaks on PE networks. The library functions do not handle fd correctly.
This occurs when /tmp/PNSD is deleted. In this case, nrt_command() leaves an open socket. This is a PNSD problem and occurs when PE integration is enabled but the node does not have PE installed or configured.

Limitations

- **lsmail** does not work with Exchange servers on Windows 2008 64bit.
- Processor number is not detected correctly on POWER7 Linux machines
- NUMA topology may be incorrect after bringing cores offline.

Bugs fixed

The July 2014 release (LSF 9.1.3) contains all bugs fixed before 30 May 2014. Bugs fixed between 8 October 2013 and 30 May 2014 are listed in the document *Fixed Bugs for Platform LSF 9.1.3*.

Fixed bugs list documents are available on Platform LSF's IBM Service Management Connect at www.ibm.com/developerworks/servicemanagement/tc/plsf/index.html. Search for the specific Fixed bugs list document, or go to the LSF Wiki page.

Chapter 2. Platform LSF product packages

The Platform LSF product consists of the following packages and files:

- Product distribution packages, available for the following operating systems:

Operating system	Product package
IBM AIX 6 and 7 on IBM Power 6, 7, and 8	lsf9.1.3_aix-64.tar.Z
HP UX B.11.31 on PA-RISC	lsf9.1.3_hppa11i-64.tar.Z
HP UX B.11.31 on IA64	lsf9.1.3_hpuxia64.tar.Z
Solaris 10 and 11 on Sparc	lsf9.1.3_sparc-sol10-64.tar.Z
Solaris 10 and 11 on x86-64	lsf9.1.3_x86-64-sol10.tar.Z
Linux on x86-64 Kernel 2.6 and 3.x	lsf9.1.3_linux2.6-glibc2.3-x86_64.tar.Z
Linux on IBM Power 6, 7, and 8 Kernel 2.6 and 3.x	lsf9.1.3_linux2.6-glibc2.3-ppc64.tar.Z
Windows 2003/2008/7/8/8.1 32-bit	lsf9.1.3_win32.msi
Windows 2003/2008/7/8.1/HPC server 2008/2012/ 64-bit	lsf9.1.3_win-x64.msi
Apple Mac OS 10.x	lsf9.1.3_macosx.tar.Z
Cray Linux XE6, XT6, XC-30	lsf9.1.3_lnx26-lib23-x64-cray.tar.Z
ARMv8 Kernel 3.12 glibc 2.17	lsf9.1.3_lnx312-lib217-armv8.tar.Z
ARMv7 Kernel 3.6 glibc 2.15	lsf9.1.3_lnx36-lib215-armv7.tar.Z

- Installer packages:

- lsf9.1.3_lsfinstall.tar.Z

This is the standard installer package. Use this package in a heterogeneous cluster with a mix of systems other than x86-64 (except zLinux). Requires approximately 1 GB free space.

- lsf9.1.3_lsfinstall_linux_x86_64.tar.Z

Use this smaller installer package in a homogeneous x86-64 cluster. If you add other non x86-64 hosts you must use the standard installer package. Requires approximately 100 MB free space.

- lsf9.1.3_no_jre_lsfinstall.tar.Z For all platforms not requiring the JRE. JRE version 1.4 or higher must already be installed on the system. Requires approximately 1 MB free space.

The same installer packages are used for LSF Express Edition, LSF Standard Edition, and LSF Advanced Edition.

- Entitlement configuration files:

- LSF Standard Edition: platform_lsf_std_entitlement.dat
- LSF Express Edition: platform_lsf_exp_entitlement.dat.
- LSF Advanced Edition: platform_lsf_adv_entitlement.dat.

- Documentation packages:

- lsf9.1.3_documentation.tar.Z
- lsf9.1.3_documentation.zip

Downloading the Platform LSF product packages

Download the LSF installer package, product distribution packages, and documentation packages from IBM Passport Advantage:

www.ibm.com/software/howtobuy/passportadvantage.

The following videos provide additional help downloading LSF through IBM Passport Advantage:

- YouTube
- IBM Education Assistant

Notices

This information was developed for products and services offered in the U.S.A.

IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web

sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Intellectual Property Law
Mail Station P300
2455 South Road,
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application

programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.



Java[™] and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

LSF[®], Platform, and Platform Computing are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software

Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.



Printed in USA

GI13-3413-04

