Improved Infrastructure Accessibility and Control with LSF for LS-DYNA

Author:

Bernhard Schott
Christof Westhues
Platform Computing GmbH, Ratingen, Germany

Presenter:

Merten Slominsky
Christoph Reichert
Platform Computing GmbH, Ratingen, Germany

Correspondence:

Platform Computing GmbH
Europa-Ring 60
D-40878 Ratingen
Germany

Tel: +49-(0)2102 610 39 0
Fax: +49-(0)2102 610 39 29
e-mail: {bschott, cwesthue, mslominsky, creichert}@platform.com

Keywords:
LSF, LSF-MultiCluster, LSF-Parallel, MPP, MPI, PVM, compute infrastructure access across multiple sites, topology aware scheduling, LSF-Scheduler, LSF-Scheduling Plug-In, MAUI, open source

ABSTRACT & INTRODUCTION – LSF-4-LS-Dyna

Tighter markets and tougher competition forces users of IT infrastructure to get the maximum speed and shortest time-to-result out of existing and cost-optimized hardware.
With the introduction of LS-Dyna MPP version, new ways to use cost-saving compute infrastructure are at hand – as well as new questions to answer:

- How to find the best hosts suitable for my MPP jobs ?
- How to control my MPP application across various hosts?
- Do those hosts need to be in my local cluster?
- How to cope with topology questions?
- How to customize LSF to my cluster topology?

Platform Computing Corporation recently introduced LSF5.1 addressing these issues.

- MPP support with LSF-Parallel – full control on all MPP children
- Transparent access to local and logically and/or physically remote compute resources with LSF-MultiCluster
- Open scheduler plug-ins, supporting custom extensions like topology aware scheduling

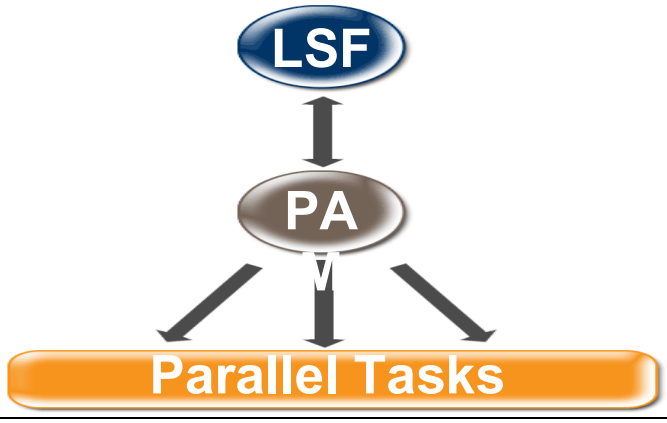
To facilitate the adoption of advanced LSF technology to solve customizing problems, Platform Computing Corporation supplies relevant parts of its source code as open source.

LSF-4-LS-Dyna : LSF-Parallel

Platform LSF-Parallel is fully integrated with the Platform LSF, to provide load sharing in a distributed system and batch scheduling for compute-intensive jobs. Platform LSF-Parallel provides support for:

- Dynamic resource discovery and allocation (resource reservation)
- Parallel batch job execution
- Transparent invocation of the distributed job processes across different platforms such as AIX, HP, Linux, SGI, and Solaris
- Full job-level control of the distributed processes to ensure no processes will become un-managed. This effectively reduces the possibility of one parallel job causing severe disruption to an organization's computer service
- The standard MPI interface(s)
- All major UNIX operating systems
- Full integration with Platform LSF, providing heterogeneous resource-based batch job scheduling including job-level resource usage enforcement

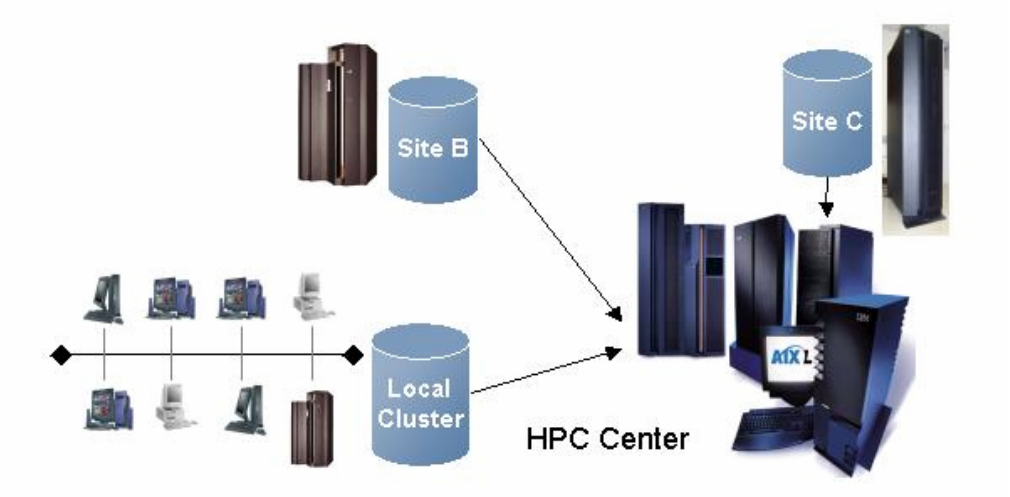
LSF-Parallel



Resource consumption control
LSF invokes an instance of the PAM = Parallel Application Manager to have full control on all children spilled off by the launching process. This includes resource consumption information – useful for accounting but also for true fairshare scheduling policy as well as for some QA: is every child consuming as much resources as expected?
Signaling service
Enabled by LSF-Parallel, LSF is always in control on all MPP children. This can be used to send signals to these MPP children, e.g. “kill”
This is particularly important in case the launching application / process had died: without LSF-Parallel, how would you find the runaway children and send termination signals?
Checkpointing MPP applications
LSF-Parallel allows for application enhancements like checkpointing. Since all MPP children are supplied with the appropriate signals, independent of the host they are actually running on, checkpointing is relatively easy to implement.

LSF-4-LS-Dyna : LSF-Multicluster

How to use resources for our LS-Dyna-MPP problems that are not local or belong to another logical / business entity?
The shown setup is in use at the IBM Grid Innovation Center in Montpellier [2].



Users are submitting their jobs in the local cluster. Depending on resource requests of the job and availability of free and suitable resources, jobs are forwarded to another cluster, e.g. another HPC-Center.
This procedure is completely transparent to the user – the user does not need to care or control availability. Input and output files are automatically forwarded as well as potential user-ID mismatches are cared for. And: yes, it runs across FireWalls.

LSF-MultiCluster allows different sites as well as different organizational entities to cooperate seamlessly without giving up their local autonomy. The local control of each participating cluster stays at the administrator assigned to. No provide resources to remote users does not mean, giving away administrative rights.

Job Forwarding Model ←→ Lease Model

In the Job-Forwarding model, the cluster that is starving for resources sends jobs over to the cluster that has resources to spare. Job status, pending reason, and resource usage are returned to the submission cluster. When the job is done, the exit code returns to the submission cluster. The scheduling of the job, once it has been received from a remote cluster is done by the local LSF-scheduler.

The lease model works differently: Two clusters agree that one cluster will borrow resources from the other, taking control of the resources. Both clusters must change their configuration to make this possible, and the arrangement, called a “lease”, does not expire, although it might change due to changes in the cluster configuration.
With this model, scheduling of jobs is always done by a single cluster. When a queue is configured to run jobs on borrowed hosts, LSF schedules jobs as if the borrowed hosts actually belonged to the cluster.
Still, the local administrators are in charge to allow this resource leasing – and could enforce the end of a lease on their own.

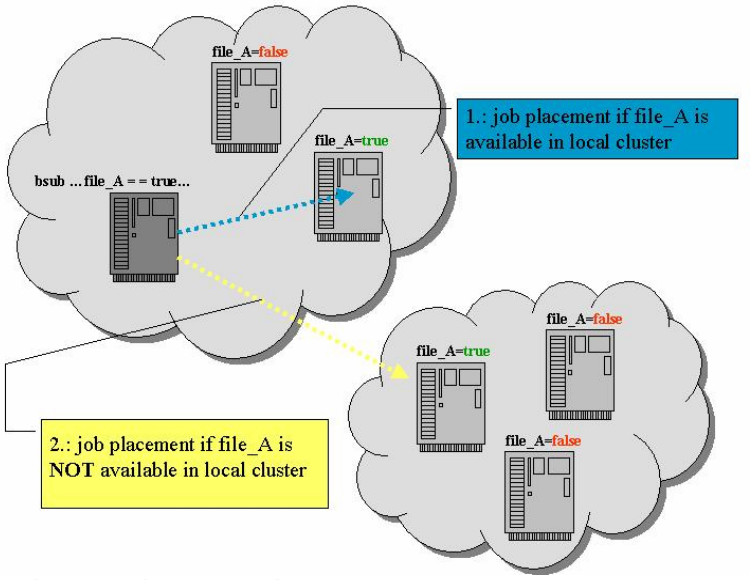
LSF-4-LS-Dyna : Topology-Awareness

The optimal placing of jobs inside a cluster or a larger machine with internal non-homogenous granularity has been supported by LSF since early 2002, detailed in [1].

The spread of workload across different organizational or geographic sites include the quest for topology aware scheduling between clusters.
Only one example should be discussed here:

Network-bandwidth translates to file transport time
What is the size of the input (and output) file of my job? Does it make sense to transport those files to a remote cluster to gain the advantage of using free CPU resources?
Answering this question would lead to “data-locality-based scheduling”.

Simplistic approach to data locality
Since LSF is a generic resource based scheduler and resources – dynamic or static – can easily be added by the administrator, one could define resources representing larger databases or other files that are supposed to be regarded as “stay local”.
When submitting a job, the resource requirement would state the need for resource “file A”. The job would get dispatched in the cluster (on a host) that has this resource “file A” available.

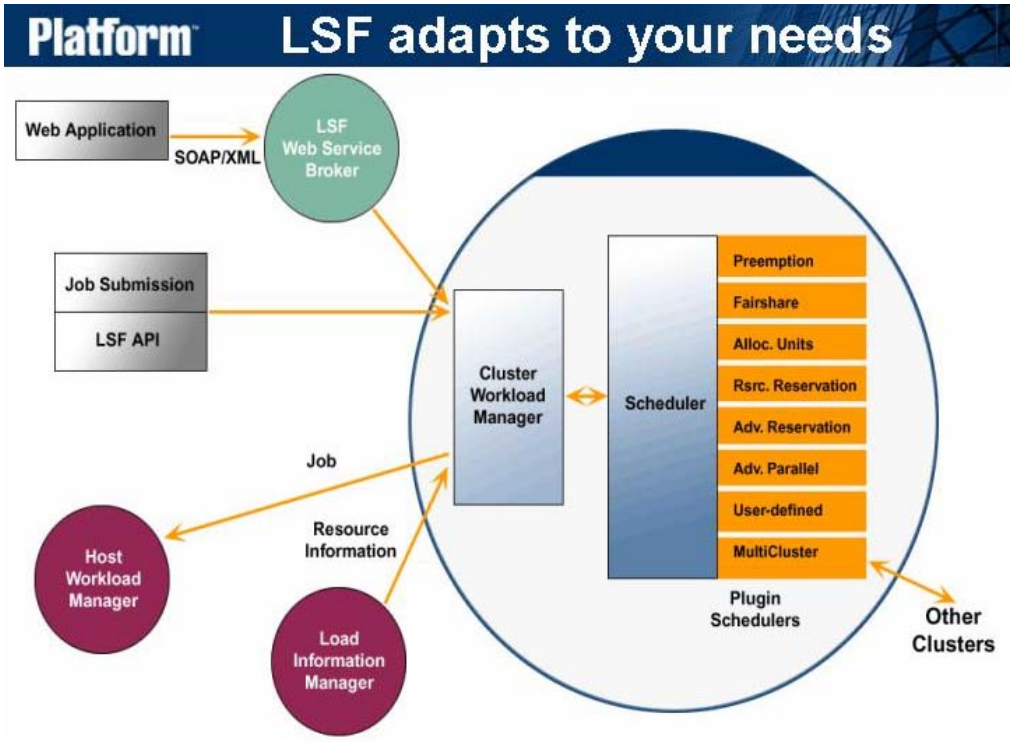


Practical implementation example
The practical implementation would care for automatic generation of the resource requirements string by using “esub” to evaluate and modify the job request submitted by the user.
Matching actual resource requirements for data with listed available data resources results in immediate dispatch decisions.
If no match has been found, the decision is based on the size of the requested data, whether transport to remote execution clusters is appropriate or not, and the job is forwarded to the responding queue.
By doing so, the list of available data-resources can be updated for reuse and data-caching is evolving.

LSF-4-LS-Dyna : Customizing the LSF-Scheduler

The above example for a very first approach to topology aware scheduling could perfectly be integrated into a scheduler-plug-in, by this being able to steer and control the full dispatching behavior of LSF.

Introduced with LSF5.0, the internal structure has changed significantly. The former monolithic LSF-Scheduler is now plug-in based, where your extensions may use the same plug-in mechanism as all standard LSF-Scheduler modules.



To facilitate the full usage of this open plug-in interface, an lsf_sdk_5.1.pdf is part of the standard documentation, featuring customization of LSF-scheduler with tutorials and example code.

The latest addition is the MAUI-Scheduler plug-in. Technically, it's a plug-in into LSF-Scheduler. Users see it as MAUI on top of LSF, where LSF is an execution layer to MAUI. At the same time, (other) users can still use the full functionality of LSF.

Additionally, an open source package has been made available, that includes the most important examples to start with.

The LSF Open Source Package

The LSF Open Source Package lsf5.1_open_source.tar.Z contains source code for the following:
cmd: The following LSF commands:
lsclusters, lsgrun, lshosts, lsload, lsmon, lsrcp, lsrun, ch, bbot, bclusters, bchkpnt, bhosts, bhpart, bjobs, bkill, bmod, bmgroup, bmig, bparams, bpeek, bpost, bqueues, bread, brestart, bresume, brun, bstop, bstatus, bsub, bswitch, btop, busers, bugroup,

esub: Master esub
melim: Master elim
plugins: LSF-Scheduler plugin examples:

plugins/fcfs:	First come first served plugin
plugins/external_plugins:	external plugin example

To access and download LSF Open Source Package lsf5.1, go to
ftp://anonymous@ftp.platform.com/pub/opensource/
pwd: <your-email-address>

Summary and Conclusions

Platform's product suite addresses the important issues and teams perfectly with LS-Dyna MPP.
LSF makes the most out of your LS-Dyna and investment and your IT infrastructure. It is reliable, easy to use and easy to implement and its through and through proven technology. It simply works.

Further on, Platform Computing has been opening its products to professional adopters by making APIs, plug-ins, and source code examples available. Customization is tremendously encouraged and supported by this, so we are looking forward to see plenty of new and interesting solutions.

References

1. Bill Mcmillan, Chris Smith, Ian Lumb & Allen Tran,
TOWARDS PREDICTABLE PERFORMANCE WITH TOPOLOGY AWARE SCHEDULING, ICPP2001,
Platform Computing Corporation,
{bmcmillan, csmith, ilumb, atran}@platform.com

2. Arend Dittmer, IBM Grid Innovation Center, LS-Dyna Demo
Platform Computing Corporation,
adittmer@platform.com