How IBM Platform LSF Architecture Accelerates Technical Computing

Sponsored by IBM

Srini Chari, Ph.D., MBA
September, 2012

mailto:chari@cabotpartners.com

Executive Summary

Advances in High Performance Computing (HPC) have resulted in dramatic improvements in application processing performance across a wide range of disciplines ranging from engineering, manufacturing, finance - risk analysis and revenue management to geological, life and earth sciences. This mainstreaming of technical computing has driven solution providers towards innovative solutions that are faster, scalable, reliable, and secure. But these mission critical technical computing solutions are further challenged with reducing cost and managing complexity.

Further, data explosion has morphed application workloads in traditional technical computing environments from compute intensive to both compute and data intensive. Also, there continues to be an unrelenting appetite and need to increasingly solve problems that evolve, grow large and are complex, further challenging the scale of technical computing processing environments. These newer domains cover a broad range of emerging applications including fraud detection, anti-terrorist analysis, social and biological network analysis, semantic analysis, financial and economic modeling, drug discovery and epidemiology, weather and climate modeling, oil exploration, and power grid management[1]*.*

Although most technical computing environments are quite sophisticated, several IT organizations are still challenged to fully take advantage of available processing capacity to adequately address newer business needs. For these organizations, effective resource management and job submission that meets stringent service level agreement (SLA) requirements across multiple departments because sharing IT infrastructure is an extremely complex process. They require higher levels of shared infrastructure utilization and better application processing throughput, while keeping costs lower. It's hard to optimize the execution of the wide range of applications using clusters and ensure high cluster utilization given diverse workloads, business priorities and application resource needs.

To address complex technical computing requirements, IBM Platform Load Sharing Facility (LSF) is IBM's flagship solution that is successfully deployed across many industries and is continuously evolving to address contemporary challenging technical computing needs. As a powerful workload manager, IBM Platform LSF provides comprehensive, intelligent, policy-driven scheduling features that enable users to fully utilize all their IT infrastructure resources while ensuring optimal application performance.

This whitepaper describes the key aspects of the IBM Platform LSF underlying architecture and how it is tuned to addresses the business challenges faced by technical computing environments to optimize the use of shared clusters. The target audience includes chief technical officers (CTO), technical evaluators and purchase decision makers, who need to understand LSF's architectural capabilities and relate them to business benefits such as how LSF can help in containing operational and infrastructure costs while increasing scale, utilization, productivity and resource sharing in technical computing environments.

[1] Big Data in HPC – Back to the future http://blogs.amd.com/work/2011/04/13/big-data-in-hpc-back-to-the-future/

Introduction

Advances in High Performance Computing (HPC) and technical computing have resulted in dramatic improvements in application processing performance across a wide range of disciplines. Although most technical computing environments are quite sophisticated, several IT organizations find it challenging to maximize productivity from available processing capacity and meet newer business needs adequately.

HPC clusters deployed today across several verticals typically consist of hundreds or thousands of compute servers, storage and network interconnect components. These require substantial investment and drive up capital, personnel and operating costs. For maximum Return on Investment (ROI), these technical computing environments must be shared across several users and departments within an organization. The ever increasing computing demands in a continuously growing compute cluster requires fair sharing and effective utilization of raw clustered compute capability. Sharing is made possible through intelligent workload management that involves job scheduling and controlling shared resources, in a way that boosts cluster resource utilization and quality of service (QoS) to meet business priorities and SLAs.

As business needs evolve, technical compute environments must manage existing deployed applications as well as address newer business requirements. Maximizing throughput[2] and maintaining optimal application performance are the two key challenges in technical computing environments that are hard to address simultaneously. High throughput requires elimination of load imbalance among constituent compute nodes in a cluster. Optimal application performance necessitates reduction in communication overhead by appropriately mapping application workload to the best-fit available compute resources in the cluster. Such technical computing needs are addressed by workload management solutions that typically consist of a resource manager and job scheduler which together prevent jobs from competing with each other for limited shared resources.

IBM Platform LSF is a powerful and comprehensive technical computing workload management platform that supports various applications and diverse workloads across several industry verticals on a computationally distributed system. It has proven capabilities such as the ability to scale to thousands of nodes, built-in high availability, intelligent job scheduling and sophisticated yet simple to use resource management capabilities that provide shared cluster manageability. LSF also provides effective monitoring and fine-grained control over workload scheduling policies that are well suited for multiple lines of business users within an organization. LSF ensures that resource allocation is always aligned to business priorities by making the most of heterogeneous shared resources in a shared cluster infrastructure. It helps improve cluster utilization and QoS by boosting job throughput, optimizing application performance thereby reducing cycle times and maximizing productivity in mission critical HPC environments.

In this whitepaper we cover the key aspects of IBM Platform LSF architecture in the context of business challenges faced by technical computing organizations. Its objective is to empower the CTOs, technical evaluators and purchase decision makers with a perspective on how LSF's architectural capabilities are well equipped to address today's HPC challenges specific to their business case. We also highlight some current LSF features and benefits and how they help in containing operational and infrastructure costs while increasing scale, utilization, productivity and resource sharing in technical computing environments.

[2] Throughput – number of jobs completed per unit of time

LSF Architecture

The IBM Platform LSF provides resource-aware scheduling and monitoring capabilities through its highly scalable and reliable architecture with built-in availability features. Its comprehensive set of intelligent, policy-driven scheduling capabilities enable full utilization of distributed cluster compute resources. The LSF architecture is geared to address technical computing challenges faced by users as well as administrators. It allows users to schedule complex workloads through easy to use interfaces. It also allows administrators to manage shared cluster resource up to petaFLOP-scale while increasing application throughput, maintaining optimum performance levels and QoS consistent with business requirements and priorities. LSF's modular architecture is unique for the workload management needs of technical computing environments by providing higher scalability and flexibility; clearly separating the following key elements of job scheduling and resource management:

- Policies that govern exchange of load information within cluster nodes and decision making for placement of tasks
- Mechanisms for transparent remote execution of scheduled jobs
- Interfaces that supporting load sharing applications, and
- Performance optimization of highly scalable HPC applications.

The following sections highlight the how LSF works, how users access various key features, what are the core elements of LSF and key associated functionality. It also covers the LSF installation architecture indicating where each LSF component is active within a cluster and how it helps in job scheduling and resource management tasks.

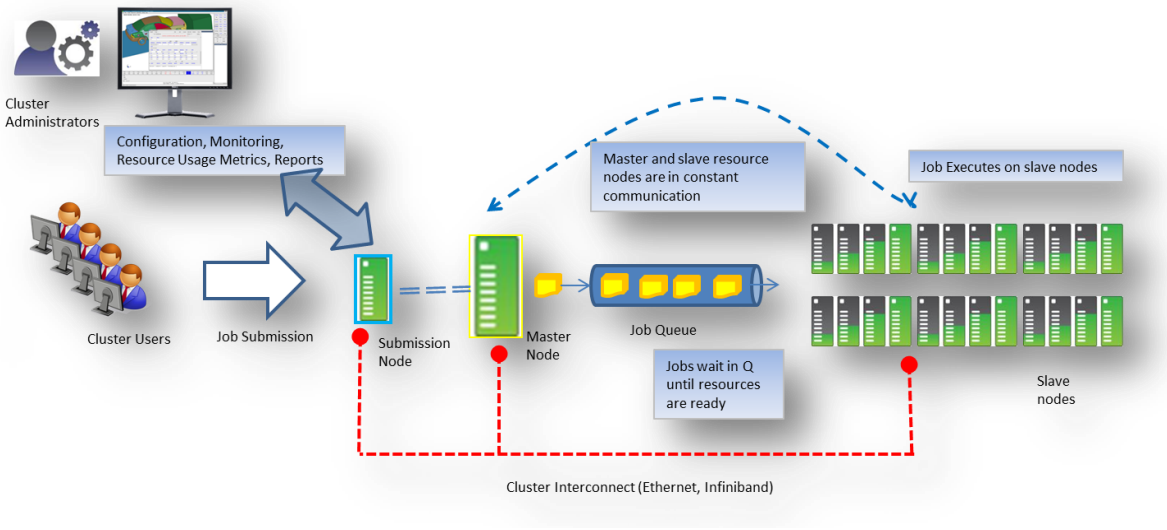


Figure 1: LSF Cluster Usage Model (source: IBM)

LSF Cluster Use Model

Individual compute resources in technical computing environments are grouped into one or more clusters that are managed by IBM Platform LSF. Figure 1 shows how a typical cluster is used and the job management and resource management roles played by different nodes in a LSF-enabled cluster. One machine in the cluster is selected by LSF as the "master" node or master host. The other nodes in the cluster act as slave nodes and can be used by scheduling algorithms to execute jobs.

Master Nodes: When the cluster nodes start up, LSF uses intelligent, fault-tolerant algorithms for master node selection. During cluster operation, if the master node fails, LSF ensures that another node takes the place of the master, thus keeping the master node highly available and cluster services

accessible to users at all times. The master node plays a key role in resource management and job scheduling. The job scheduling decisions are governed by business priorities and policies set up by the cluster administrator.

Users connect to a cluster via a client and submit their jobs at the job submission node. As these user jobs queue up, the master decides where to dispatch the job for execution based on the resource required and current availability of the resources among slave nodes.

Slave Nodes: Each slave machine in the cluster collects its own "vital signs" or the load information periodically and reports them back to the master. Detailed information on the load index[3] for each node in the cluster is analyzed and used for scheduling decisions to reduce job turnaround time and cluster throughput. LSF has unique algorithms for smart information dissemination of the load index and resource usage status to optimize cluster scalability and reliability. These algorithms are proven to scale up to thousands of nodes in a cluster. These mechanisms help increase shared resource utilization and enable technical compute environments to manage resource demands efficiently.

Workload Execution: LSF has a remote execution component that starts or stops the jobs on the assigned slave node. Once the scheduled jobs complete on slave nodes, the completion results and job status are communicated to the user. LSF also generates reports on resource usage and detailed job execution logs. Users are able to obtain job execution results as if they were executing those on a local node. LSF frees the cluster users from having to decide which nodes are best for executing a job while allowing administrators to set up policies for job execution logic that are best suited for business needs.

LSF also provides options to checkpoint a job that is running on a slave node, move it to a different slave node and then resume execution. This feature can help to temporarily suspend running jobs, free up resources for any critical jobs and then resume jobs from the last execution point instead of having to restart them all over, thus improving cluster flexibility and utilization.

LSF Components

This section gives an overview of some of the core components of LSF and their key role in job scheduling and resource management functions. LSF is a layer of software services on top of UNIX and Windows operating systems that creates a single pool of networked compute and storage

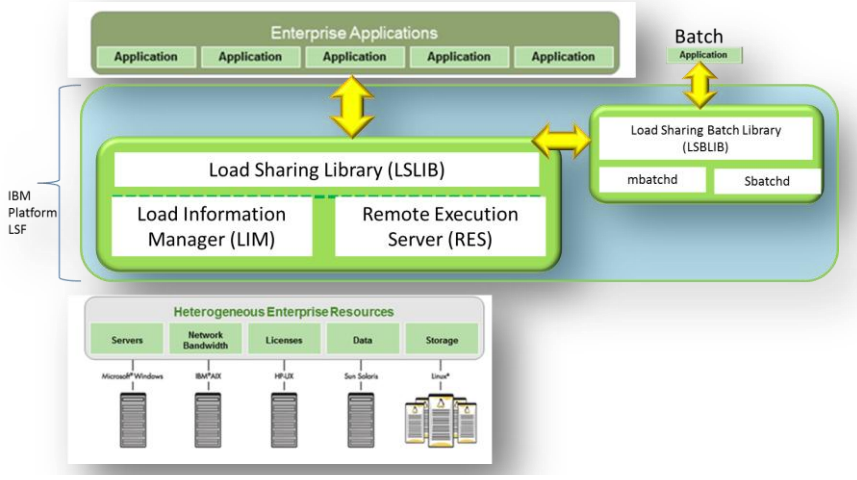


Figure 2: LSF Services - High Level architecture (source: IBM)

[3] Load Index: LSF defines a ***load -index*** for each type of resource. Load index quantifies each node's loading condition. Depending on the nature of the resource, some possibilities are queue length, utilization, or the amount of free resource. Reference: Utopia – a load sharing facility for a large scale heterogeneous system http://cse.unl.edu/~lwang/project/Utopia_A%20Load%20Sharing%20Facility%20for%20Large,%20Heterogeneous%20Distributed%20Computer%20Systems.pdf

resources. This layered service model provides a resource management framework to allocate, manage and use clusters as a single entity (Figure 2). LSF takes job requirements as inputs, finds the best resources to run the job, schedules and executes jobs and monitors its progress. Jobs always run according to host load and site policies. **LSF Base**, **LSF Batch** and **LSF Libraries** are its three basic components. Together, they help in distributing work across existing heterogeneous IT resources; creating a shared, scalable, and fault-tolerant infrastructure that delivers faster and more reliable workload performance.

LSF Base provides basic load-sharing services for the cluster such as resource usage information, host selection, job placement advice, transparent remote execution of jobs and remote file options. These services are provided through the following sub-components:

- Load Information Manager (LIM)
- Process Information Manager (PIM)
- Remote Execution Server (RES)
- LSF Base application programming interface (API)
- Utilities such as `lstools`, `lstcsh` and `lsmake`

LSF Batch extends LSF base services to provide a batch job processing system along with load balancing and policy-driven resource allocation control. To provide this functionality, LSF Batch uses the following LSF base services:

- Resource and load information from LIM to perform load balancing activities
- Cluster configuration information and master LIM election service from LIM
- RES for interactive batch job execution
- Remote file operation service provided by RES for file transfer.

LSF Libraries provide APIs for cluster application developers to get job scheduling and resource management functionality provided by LSF. There are two LSF libraries LSLIB and LSBLIB:

- LSLIB is the core library that provides basic workload management services to applications across a shared cluster and is a runtime library to easily develop load sharing applications
- LSLIB implements a high level procedural interface that allows applications to interact with LIM and RES. The other library, LSBLIB, is the batch library and it provides batch services that are required to submit, control, manipulate, and queue jobs on cluster nodes.

LSF Installation Architecture

LSF consists of a number of servers or daemon processes that run with root privileges on each participating host (Figure 3) in the cluster and a comprehensive set of utilities that are built on top of the LSF API.

LIM	Load Information Manager (**LIM**) runs on every cluster node, monitors its load and reports load information to LIM running on the master node. Master LIM (**MLIM**) collects information from all slave hosts running in the cluster and provides the same information to the applications.
RES	Remote Execution Service (**RES**) runs on every node in the cluster and accepts remote execution requests providing fast, transparent, and secure remote execution of tasks.
PIM	Process Information Manager (**PIM**) is started by **LIM** and runs on each node in the cluster. It collects information about job processes running on the host such as CPU and memory used by the job, and reports the information to **sbatchd**.
mbschd	Master Batch Scheduler (**mbschd**) daemon runs on the master host. It makes scheduling decisions based on job requirements, policies, and resource availability and communicates scheduling decisions to the **mbatchd** daemon.
mbatchd	Master Batch daemon (**mbatchd**) runs on the master host and is responsible for the overall state of jobs in the system. It receives job submission and information query requests. It manages jobs held in queues and dispatches jobs to hosts as determined by **mbschd**.
sbatchd	Slave Batch daemon (**sbatchd**) runs on each slave host. It receives request to run the job from **mbatchd** and manages local execution of the job. It is responsible for enforcing local policies and maintaining the state of jobs on the host. **It** forks a child **sbatchd** for every job and exits when job is complete. The child **sbatchd** runs an instance of **RES** to create the execution environment in which the job runs.

Figure 3: LSF daemons and their functions in scheduling & resource management (source: IBM)

There are multiple LSF processes running on each host in the cluster. The type and number of processes running depend on whether the host is a master host, a compute or slave host or one of the master node candidates as shown in Figure 4.

On each participating host in a LSF cluster an instance of LIM runs and exchanges load information with its peers on other hosts and advises applications and associated tasks to determine which hosts are best for execution. Multiple resources on each host and the resource demands of each application are considered in LIM placement decisions. In addition to help LSF make placement decisions, LIM also provides load information to those applications that make their own placement decisions.

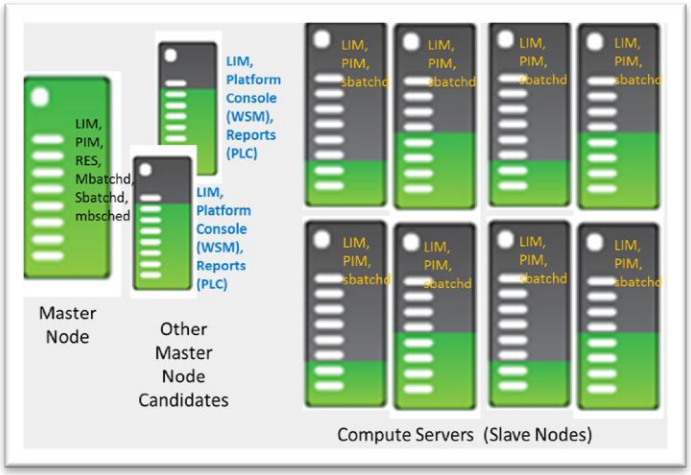


Figure 4: Installation architecture with various LSF processes running on different nodes in a LSF managed cluster (source: IBM)

Besides LIM, RES is another server or daemon on each host. RES provides the mechanisms for transparent remote execution of arbitrary tasks. Typically, after placement advice has been obtained from the LIM, a stream connection is established between the local application and its remote task through RES on the target host. This is followed by remote task initiation. LSF supports several models of remote execution to meet the diverse functional and performance requirements of applications. A LIM and a RES run on every Platform LSF server host. They interface with the host's operating system to give users a uniform, host-independent environment. Figure 5 shows sample job submission steps, for regular as well as batch jobs that are run on a LSF enabled cluster and various interactions between LSF components during the job submission and execution process.

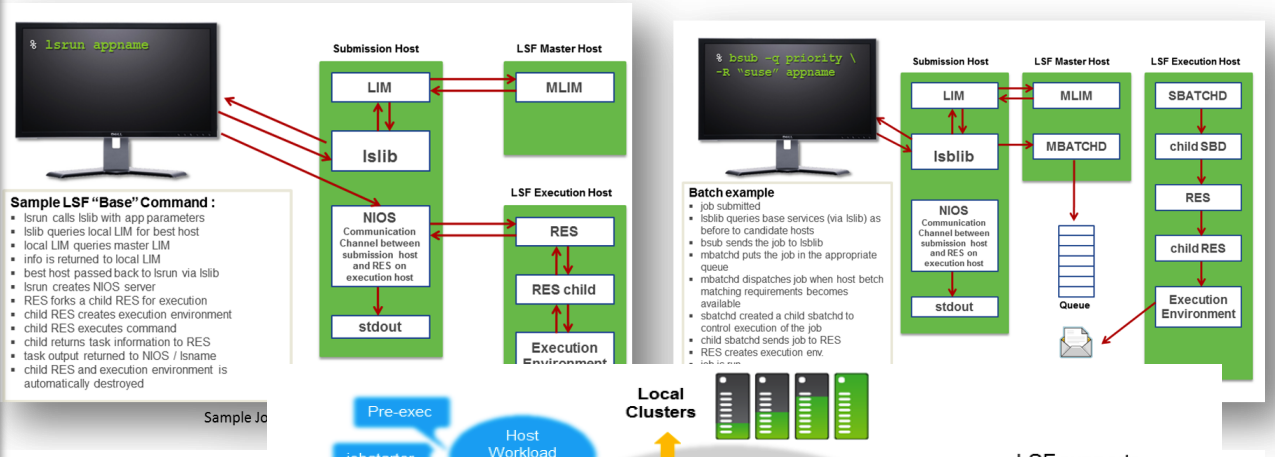


Figure 5: I

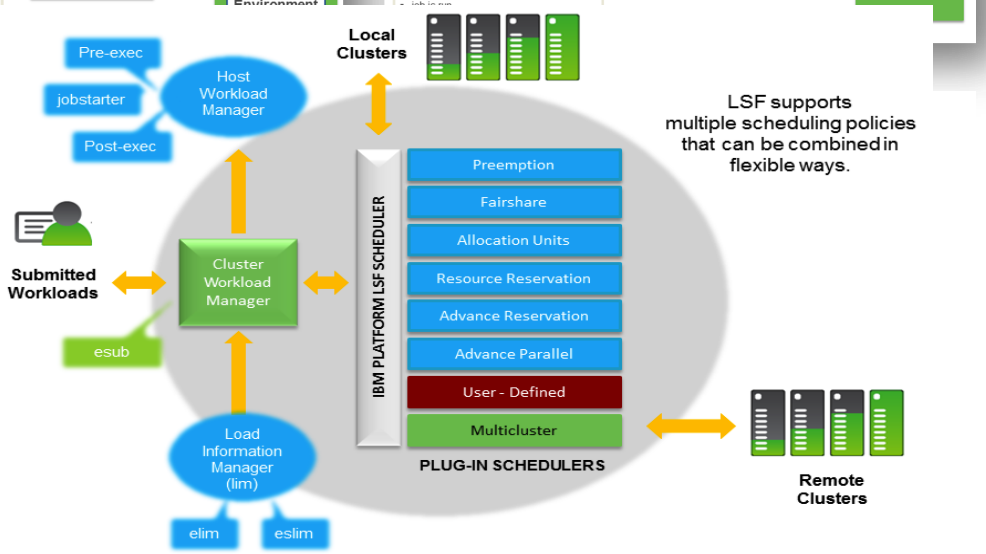


Figure 6: LSF Plug-in scheduler architecture (source: IBM)

Another unique architectural feature of LSF is that it allows multiple scheduling policies to coexist in the same cluster. Figure 6 shows the central component of the LSF scheduling architecture - the plug-in scheduler, that provides support for multiple scheduling policies. LSF makes scheduling decisions based on a flexible architecture that accommodates multiple scheduling approaches that can run concurrently and be used in combination, including user-defined scheduling approaches. The LSF scheduler plug-in API can be used to customize existing scheduling policies or implement new ones that can operate with existing LSF scheduler plug-in modules. These custom scheduling policies can influence, modify, or override LSF scheduling decisions empowering administrators to model the job scheduling decisions aligned with business priorities. The scheduler plug-in architecture is fully external and modular; new scheduling policies can be prototyped and deployed without having to change the compiled code of LSF.

Smarter scheduling Advantages:

- Higher throughput at a lower cost
- Flexibility to address changing business needs
- Better asset utilization & ROI
- Better service levels to end-users
- Increased automation & reduced manual intervention

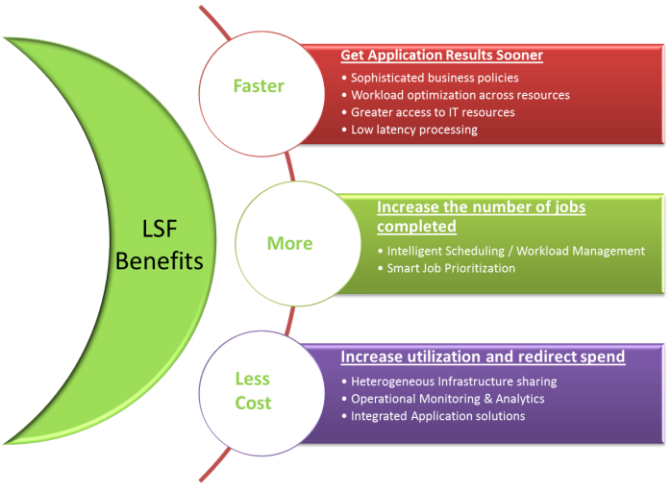
LSF Architectural Strengths

LSF's architectural strength results from its modular structure that even allows parts of the system to be used independent of other parts. For instance, a task can be executed on a remote host specified by the user so that LSLIB can contact the remote RES component, without needing the LIM component. Similarly, load information and placement advice from LIM may be obtained for purposes other than remote execution. Another advantage of the LSF architecture is that policies and mechanisms of load sharing may be changed independent of each other as well as independent of the applications running on the cluster. This provides significant fine grain control over resource sharing and job scheduling.

While LSF manages cluster sharing and job scheduling complexities with its smart architecture, it also provides easy-to-use and simple interfaces that improve user and administrator productivity and boost collaboration in technical compute environments. The highly available single master node concept for managing an entire cluster simplifies cluster management and frees up domain experts to focus on value added work instead of tedious job scheduling and cluster management tasks. At higher scale, LSF deploys a hierarchical master node concept internally but all that complexity is hidden and does not impact its simplified use model. Users can access resources from clusters with thousands of nodes that could be spread across geographies through additional LSF components such as LSF MultiCluster. LSF is architected to run on a variety of x86 hardware and operating environments including the latest generation of IBM System x servers. It is also certified on IBM Power Systems servers running the AIX and Linux operating systems.

IBM Platform LSF Benefits

LSF allows multiple users to share heterogeneous assets more effectively in a shared computing environment. Consequently, people are more productive, projects are completed earlier and because computer utilization is better, infrastructure costs are contained. By consolidating compute resources from multiple, distributed clusters, workload can be distributed more efficiently across an organization's technical computing assets that are geographically dispersed. With this capability, effective sharing of resources can be extended from a single cluster to enable flexible hierarchical or peer-to-peer workload distributions between multiple clusters.



LSF improves efficiency by removing the problem of underutilized compute resources by allowing local administrators to retain control of their own assets while still permitting remote clusters to tap idle capacity. LSF transparently extends cluster-level capabilities into the grid, making it exceptionally fast and cost-efficient to deploy, eliminating the need for sites to implement an expensive, customized scheduling layer to share resources between clusters. With simple interfaces and a plug-in modular architecture, LSF lowers the learning curve and increases cluster user productivity, reduces application integration and training costs, and speeds up job completion by eliminating manual job submission errors through automation. Technical computing users obtain faster results and complete more jobs using shared cluster resources at lower costs.

In short, LSF equips technical computing environments to achieve the following benefits:

- Obtain higher-quality results faster

- Reduce infrastructure and management costs, and
- Easily adapt to changing user requirements.

Conclusion

Technical computing environments typically run varied applications and workloads ranging from performance sensitive, compute- intensive, data-intensive or a combination, on clusters and large scale distributed systems. While such environments demand reliability as well as scalability from the underlying IT infrastructure, budgetary constraints and competitive pressures make it imperative to increase resource utilization and improve infrastructure sharing efficiencies to achieve better collaboration, productivity and faster time to results. Flexibility, scalability and agility are the key requirements of technical computing environments[4].

In such large scale distributed systems, vast amount of scattered computing resources are made available to users through dynamic and transparent load sharing provided by LSF. Through its transparent remote job execution, LSF harnesses powerful remote hosts to improve application performance and enables users to access resources from anywhere in the system. The IBM Platform LSF architecture is geared to create a scalable, reliable, highly utilized and manageable shared infrastructure for technical computing environments with powerful resource management and scheduling solutions cutting across cluster silos. Its modular architecture provides the much needed flexibility and fine-grained control while speeding up job turnaround times and improving productivity. Simple interfaces and easy customization features of LSF reduce complexity and management costs; facilitate better collaboration, tighter alignment of scheduling and resource management tasks with business objectives and priorities. LSF lowers operating costs smartly by matching the limited supply of shared resources with application demands and business priorities through features such as guaranteed resources, live re-configuration, fair-share and pre-emptive scheduling enhancements, better performance and scalability. IBM Platform LSF is architected to optimally place workloads not only based on the capability of a cluster machine to run a workload, but based on a determination of what host is best able to run the workload while ensuring broader business policies and requirements are met.

To support many technical computing environments, customers are challenged with manual and cumbersome tools integration and may require multiple dedicated personnel to develop and maintain custom integration between various tools and applications. This increases costs and business risks because some mission-critical functionality could be expensive or time-consuming to realize.

The IBM Platform LSF product family has the broadest set of capabilities in the industry with significant differentiation as many technical computing components (tools, applications, middleware, etc.) are tightly integrated and fully supported. With an integrated product family, LSF comes packaged with all the engineering, integration and processes so that technical computing users can be productive and focus on their core business, engineering or scientific tasks. This integrated family also reduces future strategic risk as the business evolves. IBM plans to enhance the capabilities of LSF and LSF-add on components in the future. Clients can expect IBM to deliver capabilities to deploy new LSF add on components on demand to keep up with ever changing requirements of the technical computing marketplace.

[4] Trends from the trenches: Bio IT World 2012 http://www.slideshare.net/chrisdag/2012-trends-from-the-trenches

Appendix: What's new in LSF 8.x

IBM Platform LSF virtualizes heterogeneous infrastructure and offers customers complete freedom of choice. Through fully integrated and certified applications, custom application integration, support for a wide variety of operating systems, it ensures that current investments are preserved while providing the strategic benefit of freedom of choice to run the best platform for the best job. The current LSF release delivers improvements in performance and scalability while introducing several new features listed below that simplify administration and boost productivity of cluster users.

New Features in LSF 8.x	Functional details	Business Benefits
Guaranteed resources	This feature guarantees resources (execution slots, entire hosts or user-defined shared resources such as software licenses) to groups of jobs. As an example, a business unit might guarantee that it has access to specific types of resources within ten minutes of a job being submitted, even while sharing resources between departments. Administrators can enable these guarantees without requiring that end-users change their job submission because jobs can be automatically attached to an SLA class via access controls procedures.	Guaranteed resources feature ensures that lower priority jobs using the needed resources can be pre-empted in order to meet the SLAs of higher priority jobs. Thus, it helps to simplify and align business SLAs with shared infrastructure configuration and use.
Live reconfiguration	Live reconfiguration feature allows changes to be made to clusters without having to re-start LSF daemons running on cluster nodes. This is useful to customers who need to add hosts, adjust sharing policies or re-assign users between groups "on the fly", without impacting cluster availability or running jobs. In cases where users are members of multiple groups, controls can be put in place so that a group administrator can only control jobs associated with their designated group rather than impacting jobs related to another group submitted by the same user.	With Live Reconfiguration, down-time is reduced, and administrators are free to make needed adjustments quickly rather than wait for scheduled maintenance periods or non-peak hours.
Delegation of administrative rights	LSF 8.x extends the concept of group administrators to enable project managers and line of business managers to dynamically modify group membership and fair-share resource allocation policies within their group. These capabilities can be delegated selectively depending on the group and site policy. Different group administrators can manage jobs, control sharing policies or adjust group membership.	Empowers line of business owners to take control of their own projects.
Fairshare & pre-emptive scheduling enhancements	This feature allows the algorithms used to tune dynamically calculated user priorities to be adjusted at the queue level. These algorithms can vary based on department, application or project team preferences. A new variable is introduced - "decay rate" to apply to currently running jobs either system-wide or at the queue level. This is useful for clusters with long running jobs. They now have more accurate view of real resource use which is useful for fair-share scheduling to consider.	Fine-grained tuning and customization of infrastructure sharing policies ensures flexibility and agility in resource sharing that matches closely with evolving business requirements.
Performance and Scalability	LSF has been extended to support an unparalleled scale of up to 100,000 cores and 1.5 million queued jobs for very high throughput EDA workloads. Even higher scalability is possible for more traditional HPC workloads. On very large clusters with large numbers of user groups employing fair-share scheduling, the memory footprint of the master batch scheduler in LSF has been reduced by approximately 70% and scheduler cycle time has been reduced by 25%.	Faster job turnaround times. Better performance and scalability.

To learn more about IBM Platform LSF, visit:
http://www-03.ibm.com/systems/technicalcomputing/platformcomputing/products/lsf/index.html