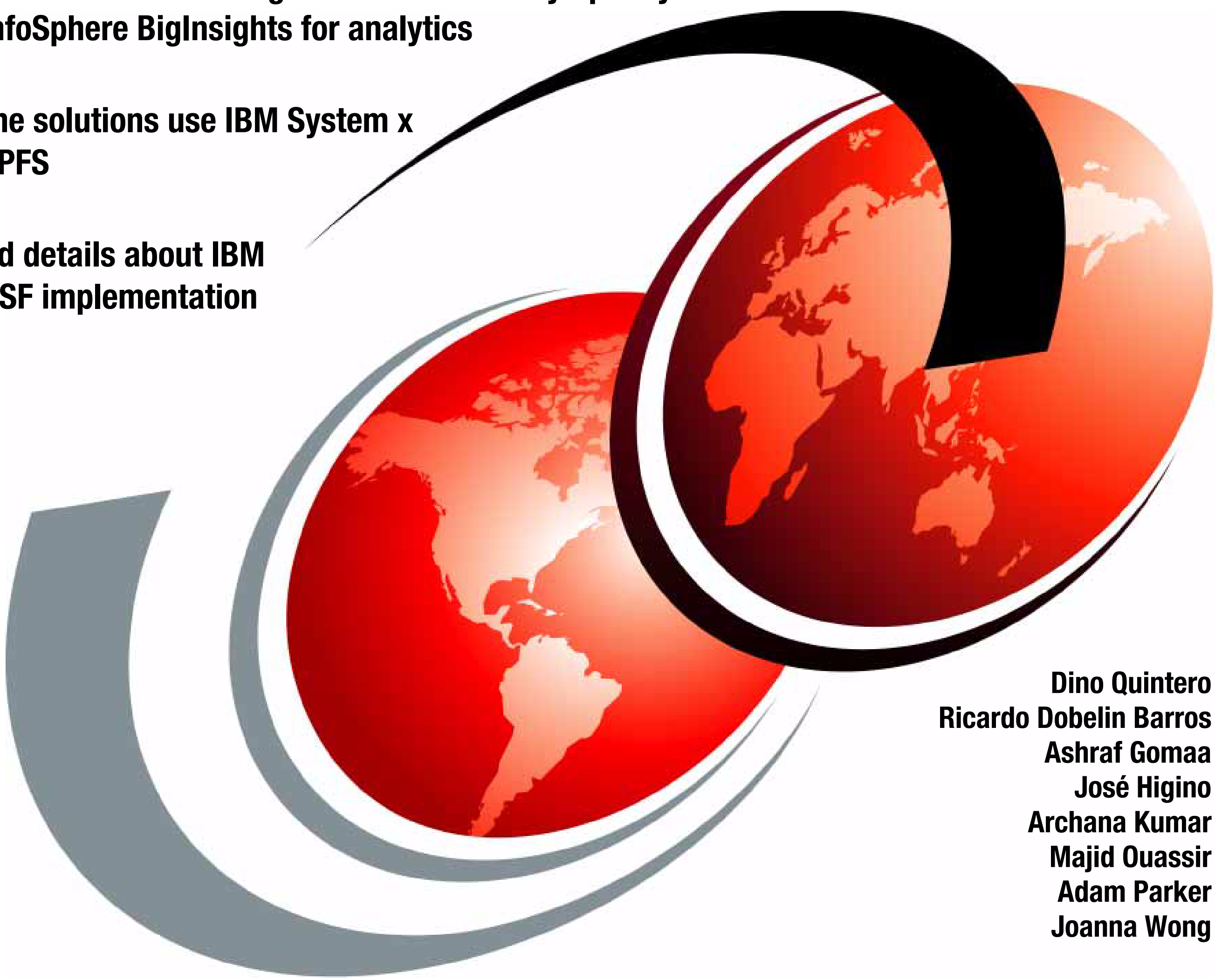
IBM Platform Computing Integration Solutions

Learn about solutions that integrate IBM Platform Symphony and IBM InfoSphere BigInsights for analytics

See how the solutions use IBM System x and IBM GPFS

Understand details about IBM Platform LSF implementation



Dino Quintero
Ricardo Dobelin Barros
Ashraf Gomaa
José Higino
Archana Kumar
Majid Ouassir
Adam Parker
Joanna Wong

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM Platform Computing Integration Solutions

March 2013

SG24-8081-00

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (March 2013)

This edition applies to IBM Platform Symphony 6.1, IBM Platform Symphony 5.2, IBM Platform LSF 8.3, IBM InfoSphere BigInsights 1.4, Red Hat Enterprise Linux (RHEL) 6.2 x86_64, and RHEL 6.3 x86_64.

Contents

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.

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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

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 measurements 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.

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 illustrate 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.

Trademarks

Preface

This IBM® Redbooks® publication describes the integration of IBM Platform Symphony® with IBM BigInsights™. It includes IBM Platform LSF® implementation scenarios that use IBM System x® technologies. This IBM Redbooks publication is written for consultants, technical support staff, IT architects, and IT specialists who are responsible for providing solutions and support for IBM Platform Computing solutions.

This book explains how the IBM Platform Computing solutions and the IBM System x platform can help to solve customer challenges and to maximize systems throughput, capacity, and management. It examines the tools, utilities, documentation, and other resources that are available to help technical teams provide solutions and support for IBM Platform Computing solutions in a System x environment.

In addition, this book includes a well-defined and documented deployment model within a System x environment. It provides a planned foundation for provisioning and building large scale parallel high-performance computing (HPC) applications, cluster management, analytics workloads, and grid applications.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Poughkeepsie Center.

Dino Quintero is an IBM Senior Certified IT Specialist and a Complex Solutions Project Leader with the ITSO in Poughkeepsie, New York (NY). His areas of knowledge include enterprise continuous availability, enterprise systems management, system virtualization, technical computing, and clustering solutions. He is currently an Open Group Distinguished IT Specialist. Dino holds a Master of Computing Information Systems degree and a Bachelor of Science degree in Computer Science from Marist College.

Ricardo Dobelin Barros is an IBM Certified IT Specialist in Brazil. Ricardo joined IBM in 2002 and has a total of 15 years of experience in the IT industry. He works in the capacity and performance group for IBM Strategic Outsourcing Delivery. Ricardo has expertise in availability management, service-levels management, sizing, and performance tuning in operating systems. He also has expertise in capacity planning, modeling, measurement, and infrastructure management for many IBM clients in Brazil. Ricardo holds a bachelor degree in systems analysis and post-graduate teaching degree in higher education.

Ashraf Gomaa is a Technical Project Manager and Team Leader at the Cairo Technology Development Center for IBM Egypt. He has worked for IBM for over 7 years with experience in software development and testing. His areas of expertise include web development, database development, globalization, IBM AIX® administration, and Linux administration. Ashraf graduated from Modern Academy in Maadi with a bachelor degree in computer science. He also has a diploma in software engineering from the Information Technology Institute in Egypt.

José Higino is an Infrastructure IT Specialist who provides AIX and Linux support and services for IBM Portugal. His areas of knowledge include System x, IBM BladeCenter®, and IBM Power Systems™ planning and implementation, management, virtualization, consolidation and clustering (HPC and high availability) solutions. José completed the Red

Hat Certified Technician level in 2007, became a CiRBA Certified Virtualization Analyst in 2009, and completed certification in KT Resolve methodology as a subject matter expert in 2011. José holds a Master of Computers and Electronics Engineering degree from Universidade Nova de Lisboa, Faculdade de Ciências e Technologia, in Portugal.

Archana Kumar is an Advisory Software Engineer for IBM and has over 10 years of experience in software development and testing. Before joining IBM, she worked in application domain at ORACLE India Pvt Ltd. Since joining IBM in March 2004, she has worked in grid computing, security, and systems management. Her areas of interest include distributed computing, IT management, virtualization, security, and open source projects. In 2002, Archana graduated with a degree in computer science engineering from National Institute of Engineering, which is affiliated to Visvesvaraya Technological University.

Majid Ouassir is a Senior Technical HPC Consultant for IBM for over 12 years. He holds Platform product certifications in IBM Platform LSF HPC, IBM Platform Symphony administrator, Platform Cluster Manager, and Platform HPC. Majid has written three publications about computer science. He holds a doctorate degree with honors in computer science from UTC: French National Higher University for Sciences and Technologies.

Adam Parker is a Senior IT Specialist in the United Kingdom. He has worked at IBM for 14 years, including 6 years in HPC. His areas of expertise include IBM Extreme Cloud Administration Toolkit, IBM General Parallel File System (GPFS™), and Linux cluster management. He co-wrote the Redbooks publication *Managing AIX Server Farms*, SG24-6606. He holds an honors degree in computer science from Wolverhampton University.

Joanna Wong is an Executive IT Specialist for the IBM STG Worldwide Client Centers focusing on IBM Platform Computing solutions. She has experience in HPC application optimization and large systems scalability performance on x86-64 and IBM Power architecture. Joanna also has industry experience in engagements with Oracle database server and Enterprise Application Integration. Joanna has an Artium Baccalaureatus degree in Physics from Princeton University and a Master of Science degree and a doctorate degree both in Theoretical Physics from Cornell University. She also has a Master of Business Administration degree from the Walter Haas School of Business with the University of California at Berkeley.

A special thank you goes to QLogic Corporation for the InfiniBand fabric equipment that we used during the residency. In addition, we thank the following people for their contributions to this project:

Ella Buslovic
ITSO, Poughkeepsie Center

Greg Geiselhart
Magnus Larsson
Oleg Litvinov
Gregory Rodgers
IBM USA

Michael Feiman
Eric Fiala
Peng Hu
Zane Hu
William Lu
Gord Sissons
IBM Canada

Bill McMillan
IBM UK

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Products and portfolio for IBM Platform Computing solutions

IBM Platform Computing offers the technology to help you gain the most from your IT investment. From pooling your technical computing resources to managing them efficiently to improving the performance of your applications, IBM Platform Computing solutions help IT organizations with the daily challenges of the data center. IBM Platform Computing solutions include the following products, which are highlighted in this chapter:

- IBM Platform LSF
- IBM Platform HPC
- IBM Platform Symphony
- IBM Platform Cluster Manager

These products provide a high-performance computing (HPC) cloud, workload management, big data analytics, and cluster management.

1.1 IBM Platform LSF

The IBM Platform LSF (Load Sharing Facility) product family is a workload management platform for demanding, distributed, and mission-critical HPC environments. It provides a set of intelligent, policy-driven scheduling features so that you can fully take advantage of all of your compute infrastructure resources and to ensure optimal application performance. By using the highly scalable and available architecture, you can schedule complex workloads and manage petaflop-scale resources.

For more information about the IBM Platform LSF product, see the product page at:

http://www.ibm.com/systems/technicalcomputing/platformcomputing/products/lsf/index.html

1.1.1 IBM Platform LSF editions

IBM Platform LSF has the following editions:

- IBM Platform LSF Express Edition
- IBM Platform LSF Standard Edition
- IBM Platform LSF Advanced Edition

For information about the editions, see the “IBM Platform LSF V9.1 family of products delivers excellent performance, scalability, and utilization” announcement letter at:

http://www.ibm.com/common/ssi/ShowDoc.wss?docURL=/common/ssi/rep_ca/8/897/ENUS213-018/index.html&lang=en&request_locale=en

Table 1-1 shows the operating systems that are supported by the editions of IBM Platform LSF.

Table 1-1 Supported operating systems

Editions		
Express	**Standard**	**Advanced**
Red Hat Enterprise Linux (RHEL) 4, 5, and 6	SLES 9, 10, and 11	SLES 9, 10, and 11
SUSE Linux Enterprise Server (SLES) 9, 10, and 11	Other Linux distributions at V2.6 or later	Other Linux distributions at V2.6 or later
Other Linux distributions at V2.6 or later	IBM AIX 6 and 7	IBM AIX 6 and 7
	HP B11.31	HP B11.31
	Oracle Solaris 10 and 11	Oracle Solaris 10 and 11
	Windows Server 2003	Windows Server 2003
	Windows Server 2003 R2 (32 or 64 bit)	Windows Server 2003 R2 (32 or 64 bit)
	Windows Server 2008	Windows Server 2008
	Windows Server 2008 R2 (32 or 64 bit)	Windows Server 2008 R2 (32 or 64 bit)
	Windows Vista, Windows 7	Windows Vista, Windows 7

Editions		
Express	**Standard**	**Advanced**
Other Linux distributions at version 2.6 or later (continued)	Windows HPC Server 2008	Windows HPC Server 2008
	Windows 8	Windows 8
	Windows Server 2012	Windows Server 2012

Table 1-2 shows the processors that are supported by the editions of IBM Platform LSF.

Table 1-2 Supported processors

Editions	Express	Standard	Advanced
Processors	AMD and Intel x86-64	AMD and Intel x86-64	AMD and Intel x86-64
		IBM POWER®	IBM POWER
		Oracle SPARC	Oracle SPARC

Optional add-ons extend IBM Platform LSF to provide a complete set of workload management capabilities that work together to address your HPC needs.

Table 1-3 shows the LSF add-on editions for IBM Platform LSF V9.1.

Table 1-3 LSF add-ons

Editions	Express	Standard	Advanced
IBM Platform Application Center	✓	✓	✓
IBM Platform RTM (Reporting, Tracking and Monitoring)	✓	✓	✓
IBM Platform License Scheduler	✓	✓	✓
IBM Platform Analytics		✓	✓
IBM Platform Process Manager		✓	✓
IBM Platform Session Scheduler		✓	✓
IBM Platform Dynamic Cluster		✓	✓

Table 1-4 shows the licensing method.

Table 1-4 Licensing LSF

Product	Unit
IBM Platform License Scheduler	Per core resource value unit (RVU)
IBM Platform Application Center	Per concurrent user
IBM Platform Process Manager	Per concurrent user
IBM Platform RTM	Per server
IBM Platform RTM Data Pullers	Per core RVU
IBM Platform MPI (Message Passing Interface)	Per core RVU

Product	Unit
IBM Platform Analytics Express Edition (10 named users, 0.5 TB of data)	Per installation
IBM Platform Analytics Standard Edition (10 named users, 1 TB of data)	Per installation
IBM Platform Analytics Advanced Edition (20 named users, 3 TB of data)	Per installation
IBM Platform Analytics Data Collectors for LSF	Per core RVU

1.1.2 IBM Platform LSF add-ons

A set of optional add-ons is offered for IBM Platform LSF to assist with workload management and so that users can become more productive. The following add-ons are available:

- IBM Platform Application Center
- IBM Platform RTM
- IBM Platform License Scheduler
- IBM Platform Analytics
- IBM Platform Process Manager
- IBM Platform Session Scheduler
- IBM Platform Dynamic Cluster

For information about these add-ons, see the IBM Platform LSF Product Family sheet at:

http://public.dhe.ibm.com/common/ssi/ecm/en/dcb03003usen/DCB03003USEN.PDF

1.1.3 New in the IBM Platform LSF V9.1 family

The IBM Platform LSF V9.1 family has the following enhancements:

- Improved performance, scalability, usability, manageability, and scheduling
- IBM Platform Session Scheduler for high-throughput batch scheduling
- IBM Platform Dynamic Cluster for workload-aware cloud computing

For more information about these enhancements, see the “Description” section in “IBM Platform LSF V9.1 delivers significant performance and scalability advances” at:

http://www.ibm.com/common/ssi/ShowDoc.wss?docURL=/common/ssi/rep_ca/3/897/ENUS212-423/index.html&lang=en&request_locale=en#h2-descx

1.2 IBM Platform HPC

IBM Platform HPC is an easy-to-use, comprehensive management software for high performance technical computing clusters and clouds. Its robust cluster and workload management capabilities are accessible by using the latest design in web-based portals. It simplifies the application integration process so that you can focus on developing your applications, instead of managing your cluster.

For more information about IBM Platform HPC and it advantages, see the IBM Platform HPC Software brochure at:

http://public.dhe.ibm.com/common/ssi/ecm/en/dcb03013usen/DCB03013USEN.PDF

IBM Platform HPC includes the following capabilities:

- Comprehensive, easy-to-use cluster management
- Next generation, easy-to-use interface
- Integrated application support
- Topology-aware workload management
- Robust workload and system monitoring and reporting
- Dynamic operating system multiboot
- Graphics processing unit (GPU) scheduling, management, and monitoring
- Robust commercial MPI library

For more information about these capabilities, see the IBM Platform HPC data sheet at:

http://public.dhe.ibm.com/common/ssi/ecm/en/dcd12354usen/DCD12354USEN.PDF

1.3 IBM Platform Symphony

IBM Platform Symphony delivers powerful enterprise-class management to run distributed applications and big data analytics on a scalable, shared grid. It accelerates dozens of parallel applications, for faster results and better utilization of all available resources.

IBM Platform Symphony has the following editions:

- IBM Platform Symphony Developer Edition
- IBM Platform Symphony Express Edition
- IBM Platform Symphony Standard Edition
- IBM Platform Symphony Advanced Edition

For information about IBM Platform Symphony, its advantages, and its editions, see the following documents:

- IBM Platform Symphony Software Family brochure

  http://public.dhe.ibm.com/common/ssi/ecm/en/dcb03002usen/DCB03002USEN.PDF

- IBM Platform Symphony data sheet

  http://public.dhe.ibm.com/common/ssi/ecm/en/dcd12359usen/DCD12359USEN.PDF

1.3.1 IBM Platform Symphony edition features

Table 1-5 lists the features for the editions of IBM Platform Symphony V6.1.

Table 1-5 IBM Platform Symphony features

Features	**Develop**	**Deploy**	**Scale**	**Converge**
	IBM Platform Symphony Editions			
	Developer	**Express**	**Standard**	**Advanced**
Low latency HPC SOA	✓	✓	✓	✓
Agile service and task scheduling	✓	✓	✓	✓
Dynamic resource orchestration		✓	✓	✓
Standard and custom reporting			✓	✓
Server, VM, desktop harvesting capability			✓	✓

Features	Develop	Deploy	Scale	Converge
	IBM Platform Symphony Editions			
	Developer	Express	Standard	Advanced
Data affinity				✓
MapReduce framework	✓			✓
Multi-cluster management				✓
Maximum number of hosts and cores	2 hosts	240 cores	5K hosts, 40K cores	5K hosts, 40K cores
Maximum number of applications		5	300	300
Desktop harvesting			✓	✓
Server and VM harvesting			✓	✓
GPU			✓	✓
Platform analytics			✓	✓
GPFS			✓	✓

1.3.2 IBM Platform Symphony add-on tools

You can use the following add-on resource harvesting tools with IBM Platform Symphony Standard and Advanced Editions to help you do more and spend less:

- IBM Platform Symphony Desktop Harvesting
- IBM Platform Symphony Server/VM Harvesting
- IBM Platform Symphony GPU Harvesting
- IBM Platform Analytics

For more information about the add-on tools, see the IBM Platform Symphony Software Family brochure at:

http://public.dhe.ibm.com/common/ssi/ecm/en/dcb03002usen/DCB03002USEN.PDF

1.3.3 New in IBM Platform Symphony V6.1

Table 1-6 lists the enhancements in IBM Platform Symphony V6.1.

Table 1-6 New enhancements in IBM Platform Symphony V6.1

Features	Improvements
Performance enhancements	Single service instance for multiple tasks (MTS)
	Parallel EGO service start (30,000 services in under two minutes)
	Shared memory logic for MapReduce (reduce data movement)
Data management	Improved data access with multi-task service instances (MTS)
	Data affinity improvements

Features	Improvements
Manageability	Improved graphical user interface
	Disk and network I/O monitoring
	Support for IBM PowerLinux™
	Support for IBM Platform Analytics (optional add-on)
	Capture task-level resource usage, consumer demand, and metadata
	Improvements to integrated reporting
Workload management	Recursive workload support, *n* levels deep
	Stacking workload to the same host
Scalability	IBM Platform Symphony MultiCluster as part of the IBM Platform Symphony Advanced Edition
	Support for 100,000 cores with MultiCluster
	Push deployment for repository services
	Session director and GUI scalability enhancements
IBM Platform Symphony Developer Edition	Combines Symphony and MapReduce APIs
	SDK/API for EGO layer phase I (Enterprise Grid Orchestrator)

The following figures show examples of GUI improvements in IBM Platform Symphony V6.1.

Figure 1-1 shows the dashboard, which provides a quick overview of the health of your cluster. The dashboard shows a summary of the workload in the cluster, a summary of utilization and status for all hosts, and links to key pages in the Platform Management Console.

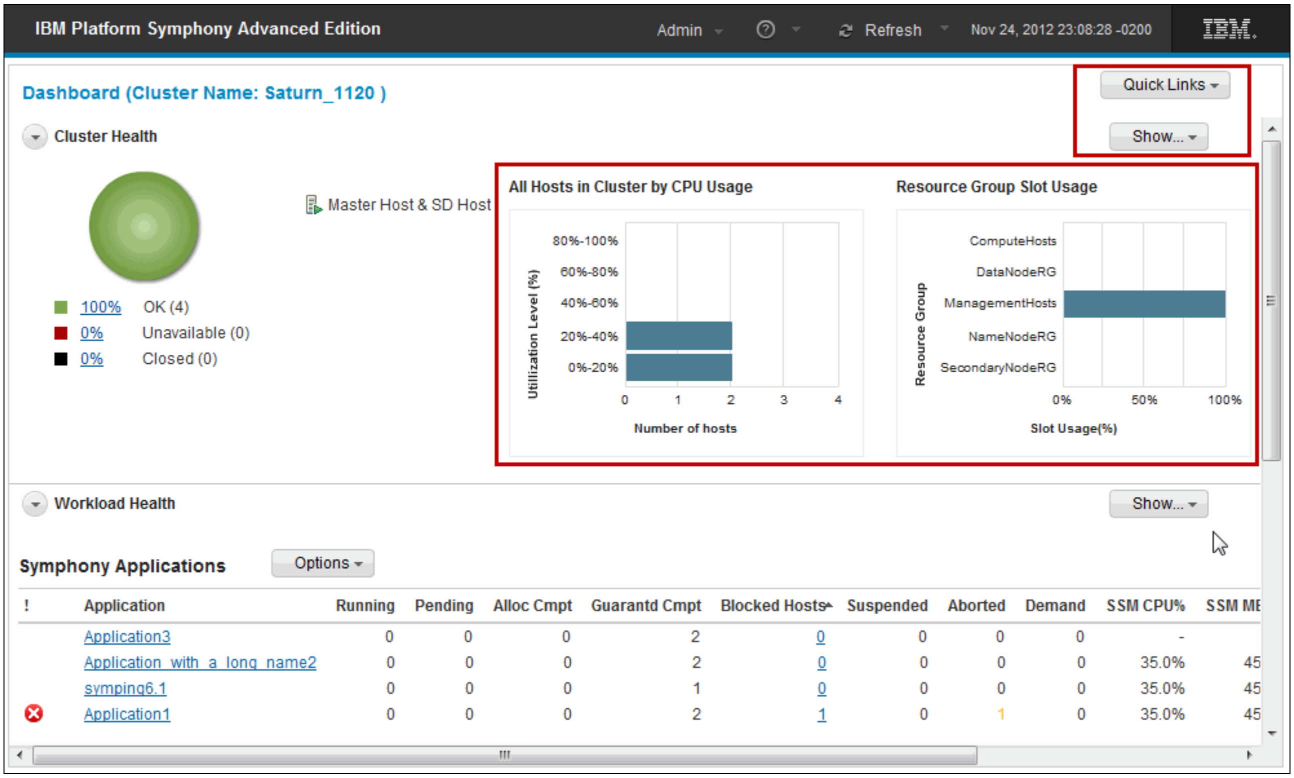


Figure 1-1 Main dashboard

Figure 1-2 shows links to the key pages in the Platform Management Console. The key pages include Symphony Workload, MapReduce Workload, System services, Resources, Cluster Settings, System Logs, Reports, Harvesting, and Add and Remove Symphony Applications.

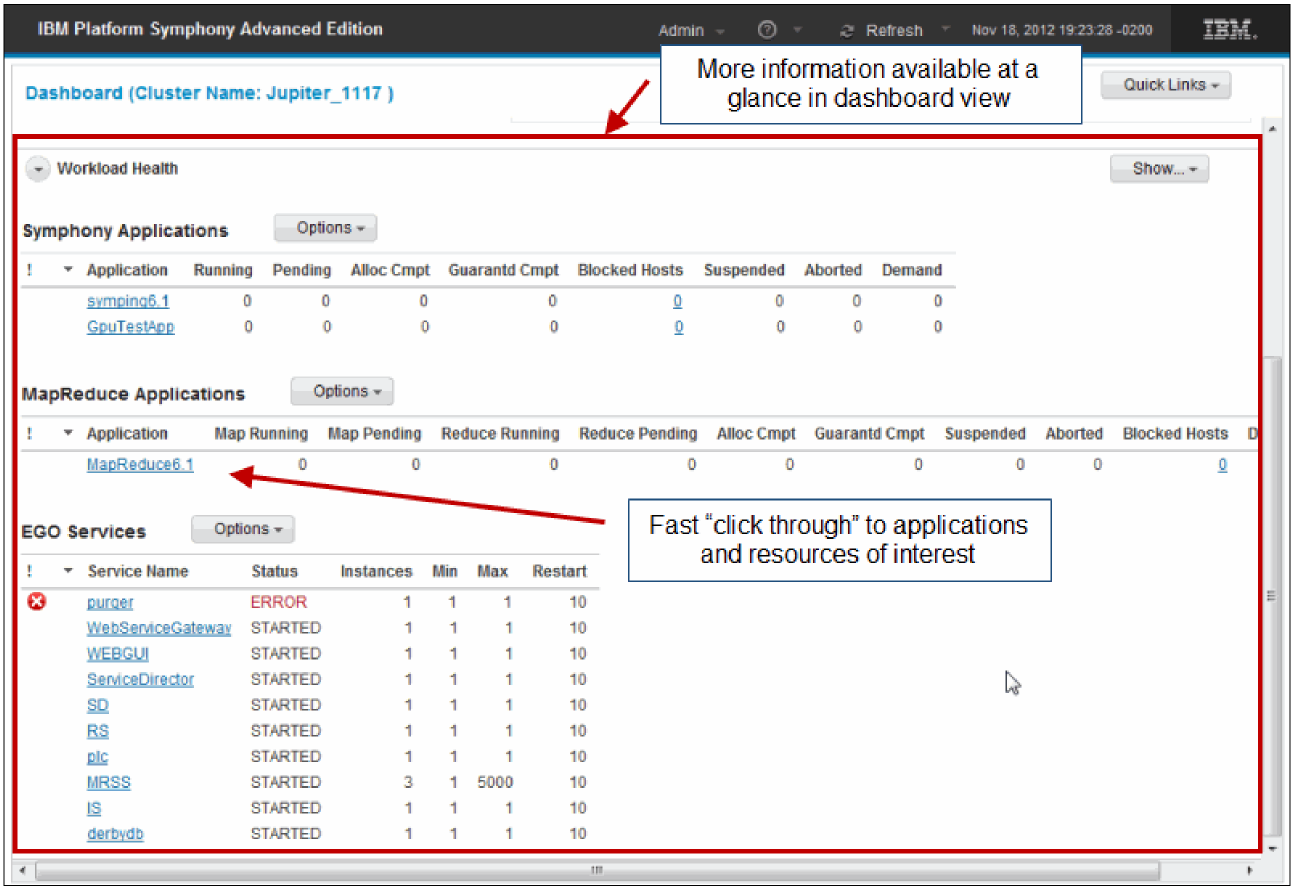


Figure 1-2 Dashboard view

Individual users can tailor application views to see applications that are specific to their department or line of business as shown in Figure 1-3.

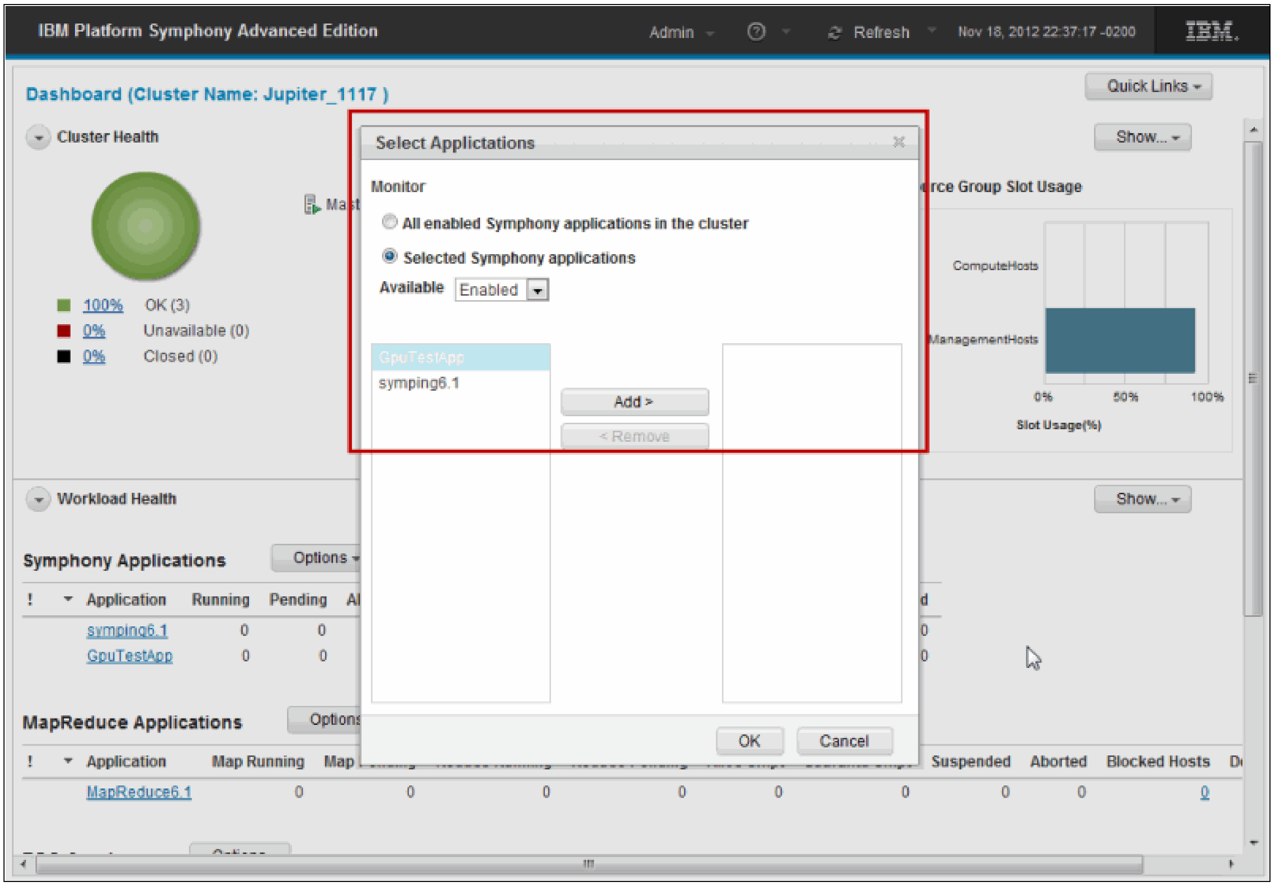


Figure 1-3 Selectively monitoring applications

Figure 1-4 shows how you can use the flexible interface of this version to tailor visual warnings and critical alerts by application.

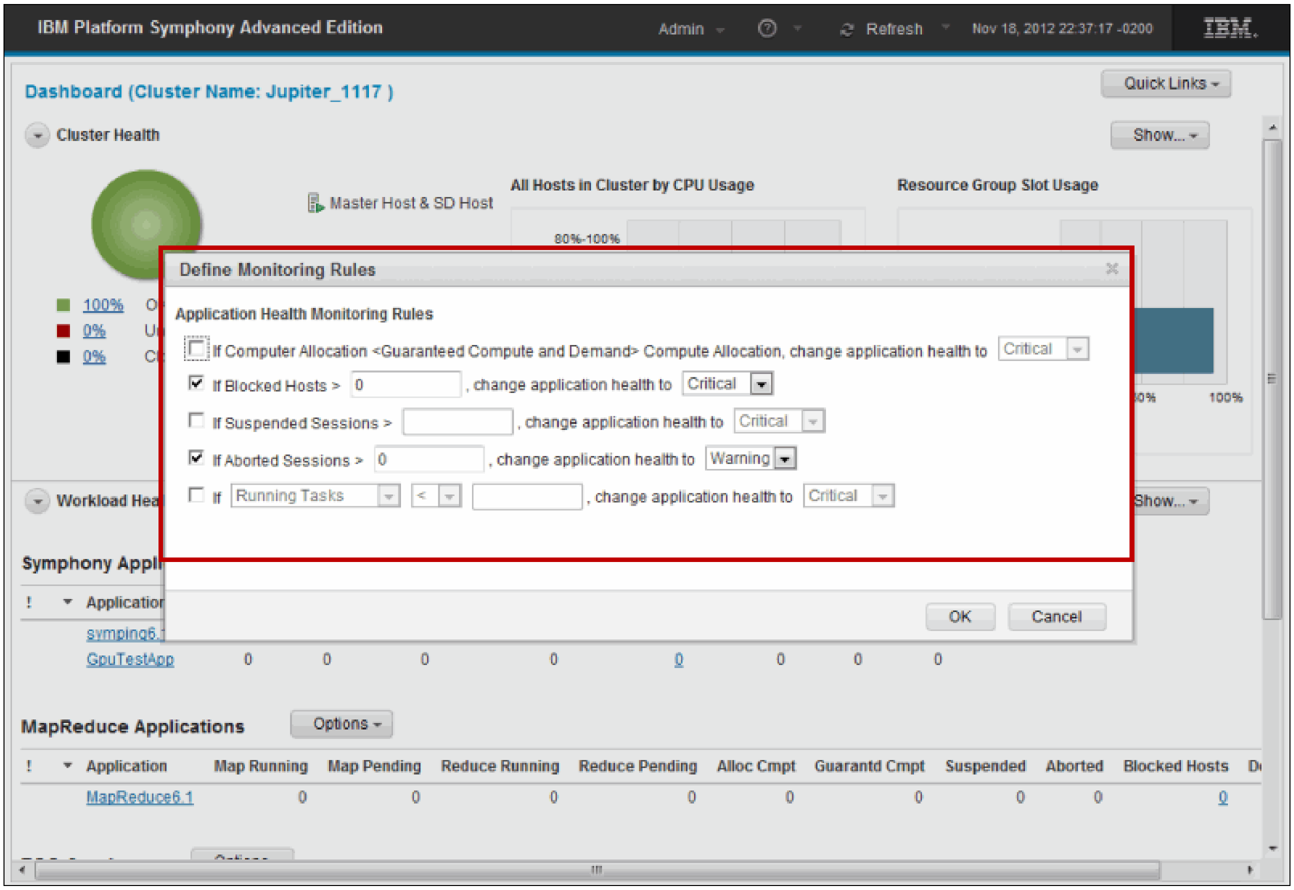


Figure 1-4 Customized visual alerts

You can use more intuitive interfaces to monitor SOA application sessions, to filter applications by consumer and type, and drag to reorder column headings as shown in Figure 1-5.

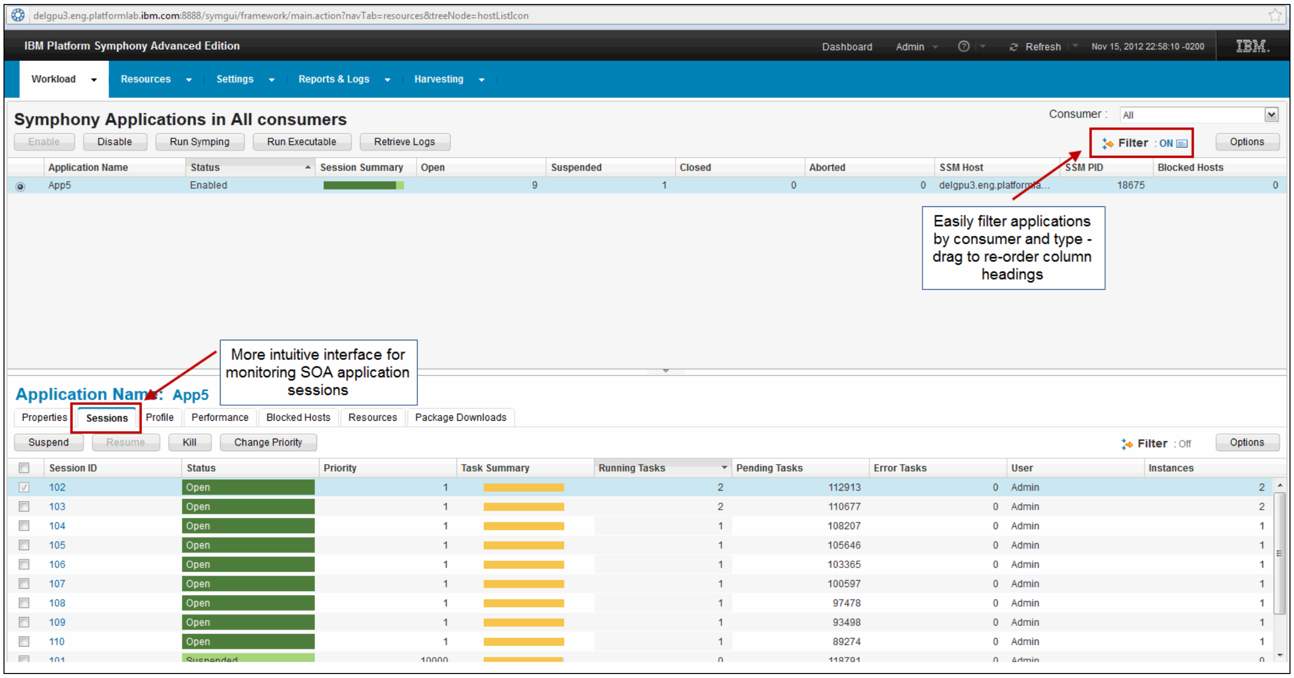


Figure 1-5 Intuitive workload monitoring interface

A new **Profile** tab (Figure 1-6) provides easy access to all application profile settings.

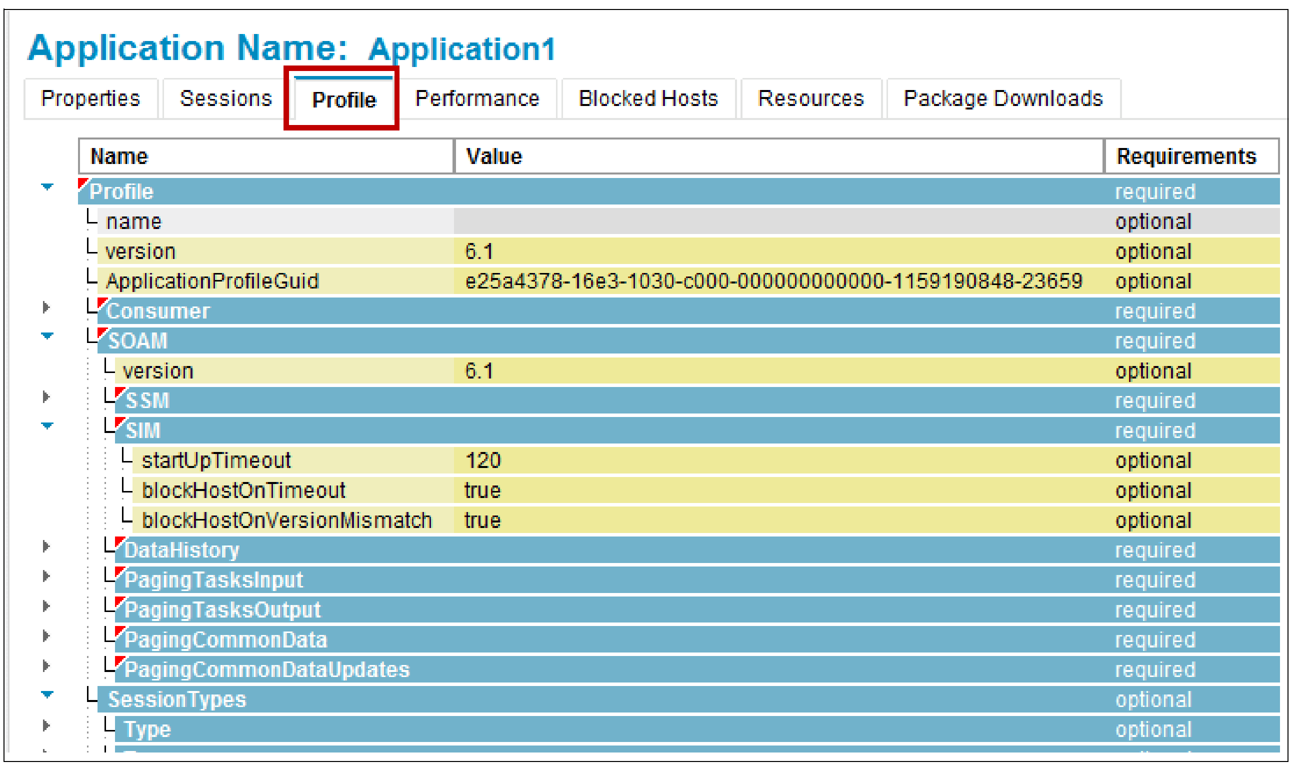


Figure 1-6 Easy access to application profiles

Figure 1-7 shows the new **Performance** tab, which provides configurable performance monitoring for SOA workloads.

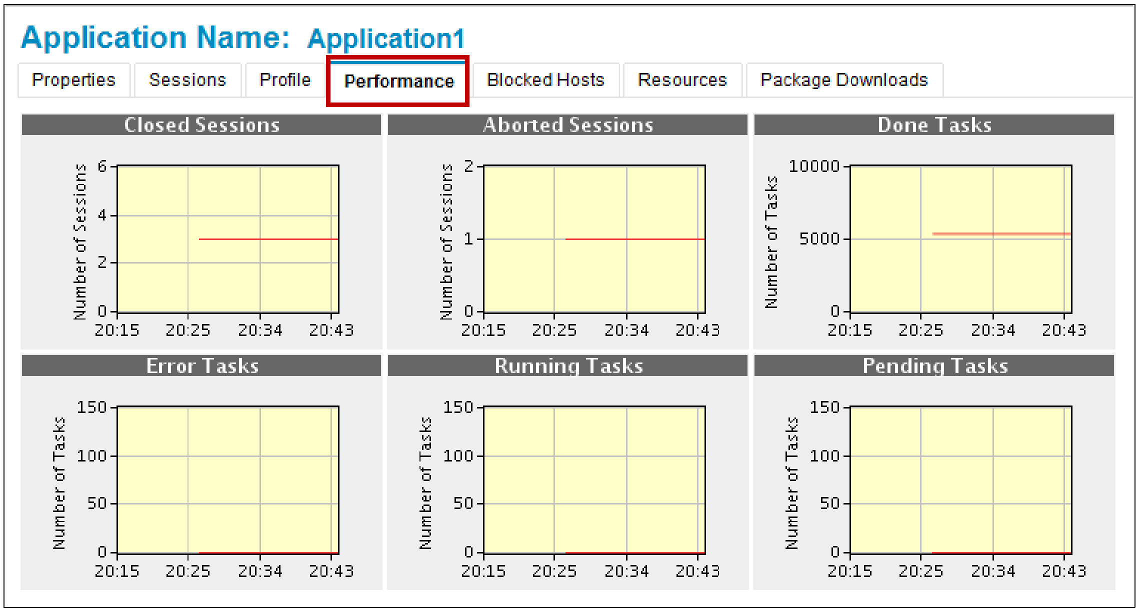


Figure 1-7 Integrated application level monitoring

Figure 1-8 shows a simple visual monitoring of multiple concurrent MapReduce workloads, including all relevant MapReduce statistics.

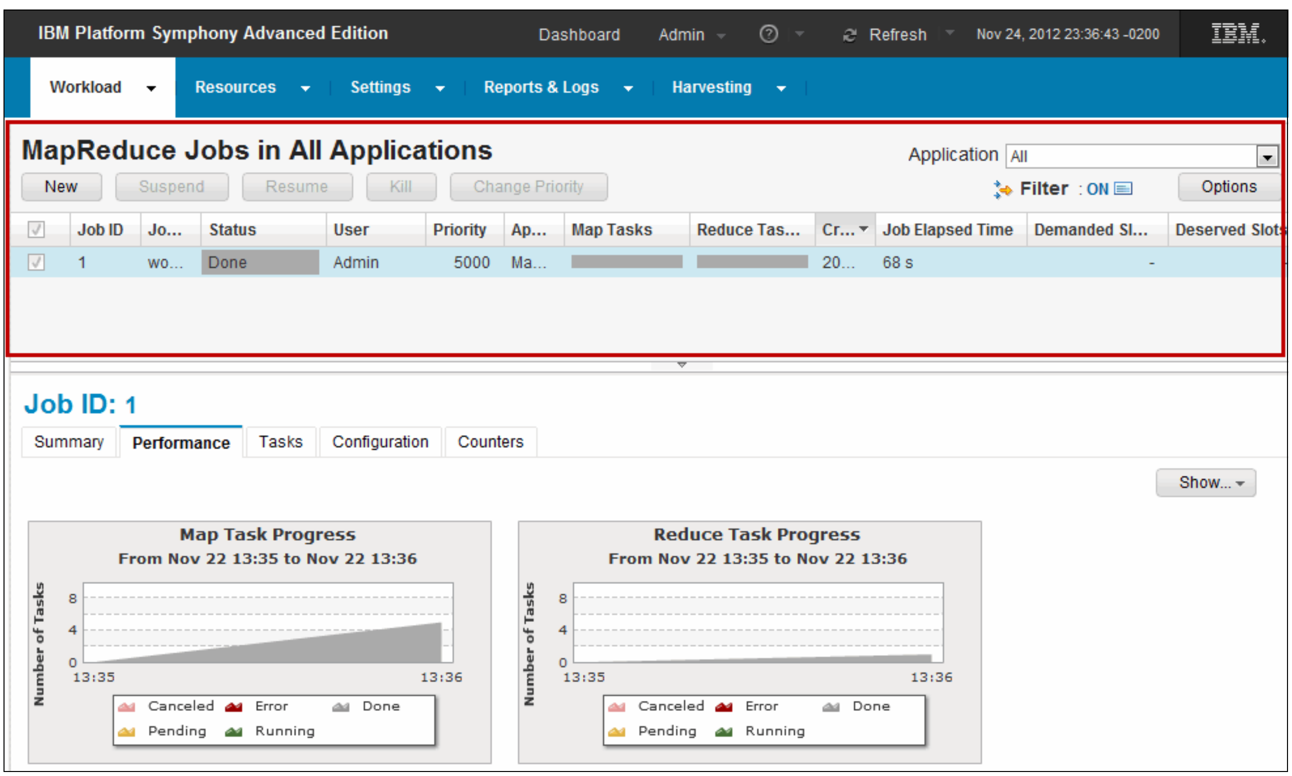


Figure 1-8 MapReduce workload monitoring

The flexible visual interface (Figure 1-9) shows dynamic sharing policies and resource loaning and borrowing at work. It shows applications that benefit from incremental resources.

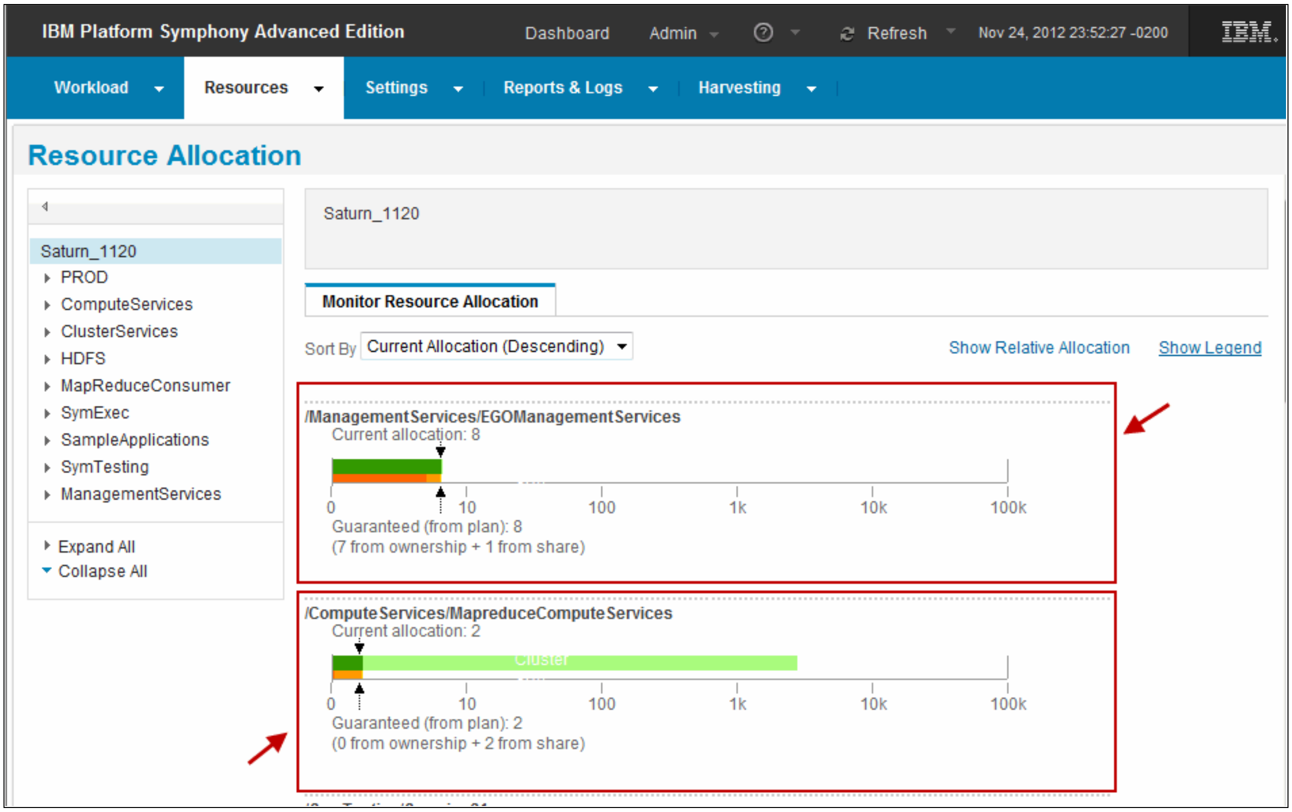


Figure 1-9 Dynamic monitoring of resource allocation

1.4 IBM Platform Cluster Manager

IBM Platform Cluster Manager quickly provisions, runs, manages, and monitors HPC clusters with unprecedented ease. It also helps to automate the assembly of multiple high-performance technical computing environments on a shared compute infrastructure for use by multiple teams.

IBM Platform Cluster Manager is available in the Standard Edition and the Advanced Edition. The next section shows a description of these products. For more information about each of the editions, see the following resources:

- IBM Platform Cluster Manager - Standard Edition data sheet

  `http://public.dhe.ibm.com/common/ssi/ecm/en/dcd12362usen/DCD12362USEN.PDF`

- IBM Platform Cluster Manager – Advanced Edition data sheet

  `http://public.dhe.ibm.com/common/ssi/ecm/en/dcd12349usen/DCD12349USEN.PDF`

1.4.1 IBM Platform Cluster Manager features

Table 1-7 shows a relation of features and versions of the IBM Platform Cluster Manager.

Table 1-7 IBM Platform Cluster Manager features

Features	Cluster Manager	
	Standard	**Advanced**
Physical provisioning	✓	✓
Server monitoring	✓	✓
Other hardware monitoring	✓	
IBM Platform HPC integration	✓	
VM provisioning		✓
Multiple cluster support		✓
Self service portal		✓
Storage management		✓
Network management		✓
Physical provisioning		✓

1.4.2 IBM Platform Cluster Manager supported environments

Table 1-8 shows the supported environments and the versions of the IBM Platform Cluster Manager.

Table 1-8 IBM Platform Cluster Manager supported environments

Supported environments	Standard	Advanced
IBM Platform LSF family	✓	✓
IBM Platform Symphony family		✓
Other workload managers		✓

2

Integration of IBM Platform Symphony and IBM InfoSphere BigInsights

In early 2012, the US Government announced the Big Data Research and Development Initiative to determine how big data can be used to address important problems that the government faces. Such problems include issues in healthcare, energy, and defense. Consumer companies, such as Amazon, are analyzing data to provide their customers with interactions that are individualized and more human. With all the attention that big data has received in the last few years, people are now realizing its value in providing deep insights into trends.

Hadoop can store big data and unlock the answers by analyzing them. IBM InfoSphere BigInsights is built on top of open source Hadoop and extends it with advanced analytic tools and other value-added capabilities. InfoSphere BigInsights helps organizations of all sizes to more efficiently manage the vast amounts of data that consumers and businesses create every day. At its core, Hadoop is a Distributed Computing Environment that manages the execution of distributed jobs and tasks on a cluster. As with any Distributed Computing Environment, the Hadoop software needs to provide facilities for resource management, scheduling, remote execution, and exception handling. Although Hadoop provides basic capabilities in these areas, IBM Platform Computing has been working on these problems and perfecting them for twenty years.

With use cases for MapReduce continuing to evolve, customers are increasingly encountering situations where scheduling efficiency is becoming important. This scenario is true from the standpoint of meeting performance and service-levels goals and from the perspective of using resources more efficiently to contain infrastructure costs. Also, as the number of applications for MapReduce and big data continues to grow, multitenancy and a shared services architecture become more critical. It is not feasible from a cost standpoint to create separately managed grid environments for every critical MapReduce workload.

This chapter describes the integration of IBM Platform Symphony and IBM InfoSphere® BigInsights. IBM Platform Symphony is a low-latency scheduling solution that supports true multitenancy and sophisticated workload management capabilities.

This chapter includes the following sections:

- IBM Platform Symphony
- Environment
- Configuring InfoSphere BigInsights
- Installing IBM Platform Symphony Advanced Edition
- Integrating IBM Platform Symphony and InfoSphere BigInsights
- Additional configuration for IBM Platform Symphony
- Benchmark tests
- Adding users
- Adding nodes
- Troubleshooting

2.1 IBM Platform Symphony

IBM Platform Symphony owes its name to its unique ability to orchestrate distributed services on a shared grid in response to dynamically changing workloads. It combines a fast service-oriented application middleware framework (SOAM), a low-latency task scheduler, and a scalable grid management infrastructure that is proven in some of the world's largest production grids. This unique design ensures application reliability and low-latency and high-throughput communication between clients and compute services.

2.1.1 How InfoSphere BigInsights works with IBM Platform Symphony

Figure 2-1 shows the various components that make up InfoSphere BigInsights Enterprise Edition.

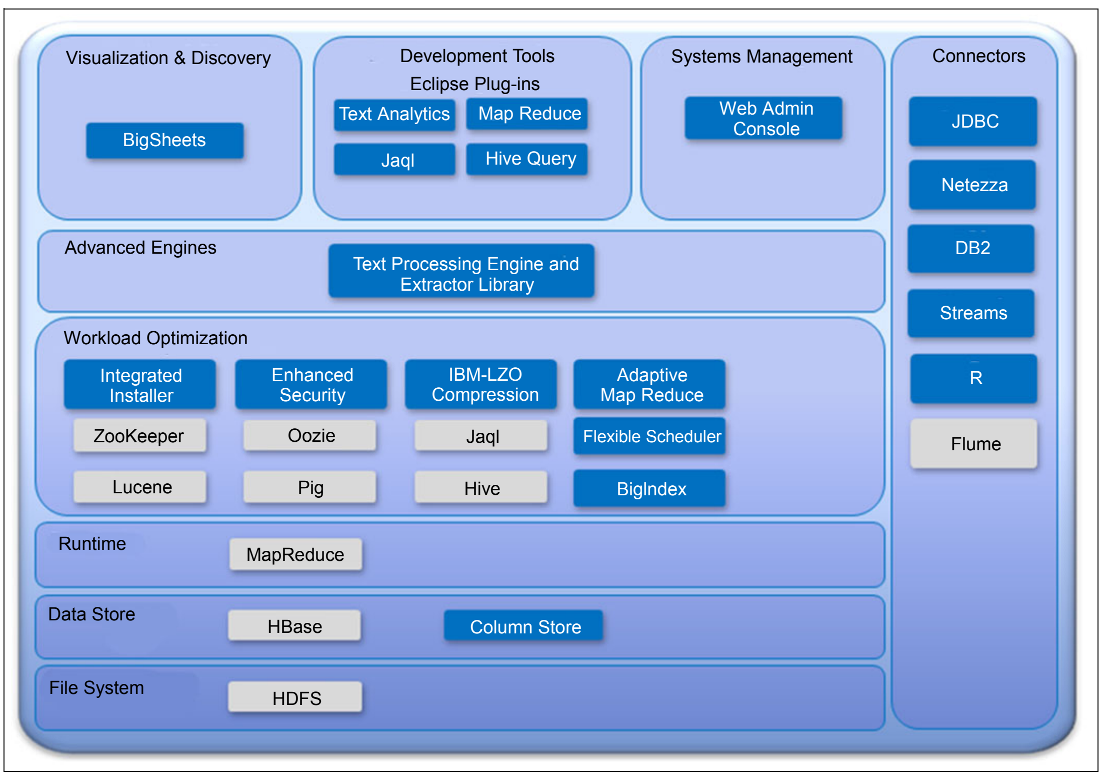


Figure 2-1 InfoSphere BigInsights components

This view makes it clear how InfoSphere BigInsights augments open source Apache Hadoop and provides more capabilities. These capabilities include visualization and query tools, development tools, management tools, and data connectors to external data stores.

When IBM Platform Symphony is deployed with InfoSphere BigInsights, IBM Platform Symphony replaces the open source MapReduce layer in the Hadoop framework. (IBM Platform Symphony is not a Hadoop distribution.) IBM Platform Symphony relies on a Hadoop MapReduce implementation that is present with various open source components, such as Pig, Hive, HBase, and HDFS file systems.

As shown in Figure 2-2, IBM Platform Symphony replaces the MapReduce scheduling layer in the InfoSphere BigInsights software environment. As a result, IBM Platform Symphony provides better performance and multitenancy in a way that is transparent to InfoSphere BigInsights and its users.

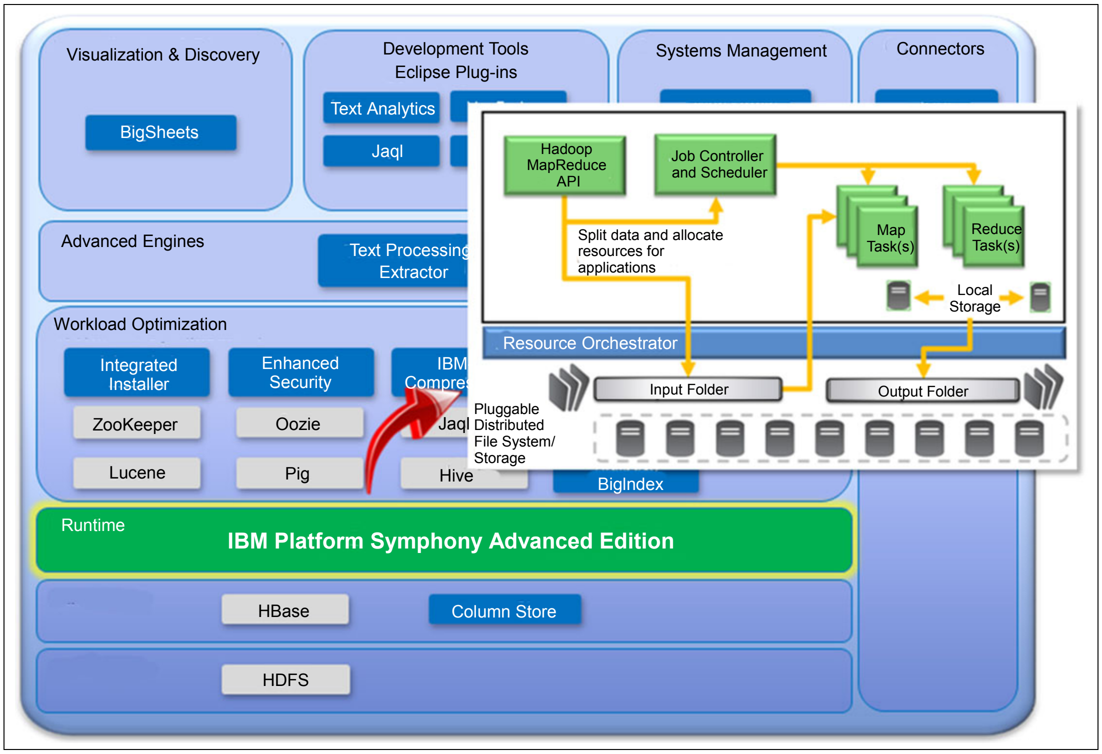


Figure 2-2 IBM Platform Symphony (Advanced Edition) runtime integration with InfoSphere BigInsights

Big data workloads can be submitted from the InfoSphere BigInsights graphical interface, from a command line, or from client applications that interact with the Hadoop MapReduce APIs. After you configure InfoSphere BigInsights to use the IBM Platform Symphony scheduler in place of the Hadoop scheduler, InfoSphere BigInsights workloads run seamlessly and are manageable from within the InfoSphere BigInsights environment.

Administrators must be aware that, when they run InfoSphere BigInsights on a shared IBM Platform Symphony grid, some cluster and service management that are accessible from within InfoSphere BigInsights become redundant. For example, it is no longer assumed that InfoSphere BigInsights has exclusive use of the grid. Therefore, IBM Platform Symphony provides capabilities for cluster node management, service management, and high availability features, for example, for such components as the NameNode, JobTrackers, and TaskTrackers.

2.1.2 The IBM Platform Symphony performance advantage

IBM continues to achieve record MapReduce performance results with IBM Platform Symphony. In a recent test that was conducted and audited by an independent third-party testing lab, IBM Platform Symphony ran a particular mix of social analytic workloads an average of 7.3 times faster than Hadoop MapReduce alone.

Performance improvements for MapReduce workloads that run on IBM Platform Symphony are a result of the following factors:

- Low-latency scheduling means that jobs start (and complete) faster.
- By design, IBM Platform Symphony monitors hosts dynamically and always schedules tasks preferentially to the host that can respond quickly to the workload.
- By avoiding the Hadoop heartbeat (polling) model, the task scheduling rate for Hadoop workloads is improved dramatically with IBM Platform Symphony and scheduling latency is reduced.
- If resources are idle on the cluster, IBM Platform Symphony MapReduce workloads can expand resource allocations dynamically to borrow unused nodes to maximize usage.
- IBM Platform Symphony uses generic slots rather than slots that are statically allocated to map and reduce functions so that slots can be shared between map and reduce tasks.
- The MapReduce shuffle phase is improved and attempts to keep data in memory while using the more efficient data transfer mechanisms of IBM Platform Symphony for moving data between hosts.
- Developers can optionally take advantage of API features and data handling enhancements that are unique to IBM Platform Symphony to achieve advantages beyond what can be achieved within MapReduce itself.

Customers who deploy InfoSphere BigInsights or other big data application environments can realize significant advantages by using IBM Platform Symphony as a grid manager:

- Better application performance
- Opportunities to reduce costs through better infrastructure sharing
- The ability to guarantee application availability and quality of service
- Ensured responsiveness for interactive workloads
- Simplified management by using a single management layer for multiple clients and workloads

IBM Platform Symphony is especially beneficial to InfoSphere BigInsights customers who are running heterogeneous workloads that benefit from low latency scheduling. The resource sharing and cost savings opportunities that re provided by IBM Platform Symphony extend to all types of workloads.

For Hadoop grid administrators who are looking for opportunities to improve performance, reduce cluster sprawl, and improve service levels at a lower cost, IBM Platform Symphony provides a powerful complement to InfoSphere BigInsights.

2.2 Environment

The integration and comparison tests took place on a group of machines at IBM Poughkeepsie Benchmark Center. The environment had the following setup:

- Hardware:
  - 12x IBM System xiDataPlex dx360 (Server Model 6391 and Type AC1) M3 nodes
    - 2x (6-core) Intel Xeon processor X5670 at 2.93 GHz

      A maximum of 4 GHz with Turbo Boost Technology and 24 logic cores with Hyper-Threading Technology (HTT) active
    - 48-GB RAM
    - 250-GB HDD (WD2502ABYS-23B7A; SATA 3 Gbps)
  - Intel (QLogic) True Scale QDR InfiniBand Switch and HCAs
  - 1-Gbps Ethernet Switch
- Software:
  - Red Hat Enterprise Linux 6.2 Server (x86_64)
  - EPEL repository
  - Intel (QLogic) InfiniBand OFED driver 7.0.1.0.43 (OFED version 1.5.3.2)
  - IBM InfoSphere BigInsights V1.4
  - IBM Platform Symphony Advanced Edition V5.2

This cluster configuration is appropriate only to validate the steps to integrate InfoSphere BigInsights with Platform Symphony MapReduce. It is not ideal for data-intensive performance benchmarks nor as a reference architecture for running IBM Platform Symphony Advanced Edition or the IBM InfoSphere BigInsights Map Reduce workload.

The cluster configuration has only one disk per node and is shared between partitions that are used for the operating system and for the Hadoop Distributed File System (HDFS). The HDFS that is configured in this example is not the recommended just a bunch of disks (JBOD) design. The JBOD design must have one disk per processor core on each data node, with each disk configured as a separate device. Also, the operating system drive was not mirrored for high availability.

The 10-Gbps network is used usually for performance configuration of the InfoSphere BigInsights cluster. This environment used the IP over InfiniBand (IPoIB) network that is configured over the Intel (QLogic) QDR InfiniBand interconnect in the cluster environment.

2.2.1 Preconfiguration on all nodes

Before we started the integration process, we ran or checked the preliminary configurations. Some of them are described in *Implementing IBM InfoSphere BigInsights on System x*, SG24-8077, which was used as a reference for this book. The details were based on the recommended settings for the machine environment and the software that were used in the integration. Therefore, the integration might change for other architecture and software configurations.

This section addresses configuration of the following areas:

- Cluster layout
- Cluster connectivity
- LVM-based file system for HDFS on all data nodes
- Customizing RHEL services
- Verifying whether all required packages are installed

- Creating users (same user configuration across all nodes)
- Temporary security changes

Cluster layout

When you build your cluster, you can install each server manually from a CD or DVD or by using a provisioning software such as IBM Extreme Cloud Administration Toolkit (xCAT). In either case, several combinations are possible when you size a cluster in terms of the number of networks and selecting which network to handle a specified workload. For InfoSphere BigInsights and Symphony integration, you can use for the user *homes*:

- Lightweight Directory Access Protocol (LDAP)
- Shared directories
  - General Parallel File System (GPFS)

    Consider using a separate network when you configure GPFS for optimal performance. GPFS can be used with the IBM Platform Symphony network (in high-performance networks such as InfiniBand). However, you must configure GPFS to guarantee less performance degradation on the IBM Platform Symphony network.
  - Network File System (NFS)

    Consider using a separate network when you configure NFS for optimal performance. If NFS is mixed with the same network as IBM Platform Symphony, performance can degrade.
- Local directories

  This integration uses the local home directories layout and uses the parallel distributed shell utility, **PDSH**, to distribute files across the nodes.

If the IBM Platform Symphony installation has more than one management node, a shared file system is required among the management nodes for failover and recovery of resource and workload runtime states. GPFS and NFS (on an external NFS server) are possible options for the shared file system.

Cluster connectivity

When you build the cluster, check the following areas to ensure connectivity:

1. Distribute SSH keys to set up passwordless authentication for root and other uses across all nodes:
   - SSH key setup for user root under a local host (Example 2-1)

Example 2-1 SSH key setup on management node (for root user)

```
# ssh-keygen -t rsa -N ""
# cat /root/.ssh/id_rsa.pub > /root/.ssh/authorized_keys
# chmod 600 /root/.ssh/authorized_keys
```

> **root SSH keys:** If you do not have a provisioning system that can distribute root keys for passwordless authentication, use **scp** on the `.ssh` directory to all nodes before you continue. Otherwise, each time you issue an SSH connection to a host, you are prompted for the password.

– SSH key setup for a specified user and for a group of nodes that use **PDSH** (Example 2-2)

Example 2-2 Generate and distribute SSH keys over the compute nodes with PDSH

```
management # su - <user>
<user>@management # ssh-keygen -t rsa -N ""
<user>@management # cat ~/.ssh/id_rsa.pub > ~/.ssh/authorized_keys
<user>@management # chmod 600 ~/.ssh/authorized_keys
<user>@management # exit
management # pdsh "scp -r management:/home/<user>/.ssh /home/<user>/"
```

2. Consider adding repository access to all nodes:

   a. Add the extra packages repository, from the Fedora project, for Red Hat Enterprise Linux (RHEL) on the management node and later on the compute nodes after you install and configure **PDSH** (Example 2-3).

Example 2-3 Repository for extra packages available in the Fedora project

```
# cat /etc/yum.repos.d/epel.repo
[epel]
name=Extra Packages for Enterprise Linux 6 - $basearch
#baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-6&arch=$base
arch
failovermethod=priority
enabled=1
priority=10
gpgcheck=0
gpgkey=
```

   b. Configure a distributed shell.

   c. Set up PDSH for the management node (Example 2-4).

Example 2-4 Setup PDSH and verify it on management host

```
management # yum install -y pdsh
management# cat /etc/dsh/all
host1
host2
management # cat /etc/hosts
1.1.2.1 management
1.1.1.1 host1
1.1.1.2 host2
management # cat /etc/profile.d/pdsh.sh
export WCOLL=/etc/dsh/all
management # pdsh hostname
host1: host1
host2: host2
```

d. Set up **PDSH** for the compute nodes (Example 2-5).

The *WCOLL* environment variable (Example 2-5) is set to the name of a file that includes a list of the host names of the compute nodes and the target nodes of the **PDSH** commands.

Example 2-5 PDSH setup on compute nodes from the management host (already with PDSH)

```
management # pdsh "scp 1.1.2.1:/etc/hosts /etc/hosts"
management # pdsh "cat /etc/hosts"
host1: 1.1.2.1 management
host1: 1.1.1.1 host1
host1: 1.1.1.2 host2
host2: 1.1.2.1 management
host2: 1.1.1.1 host1
host2: 1.1.1.2 host2
management # pdsh "scp management:/etc/yum.repos.d/epel.repo
/etc/yum.repos.d/epel.repo"
management # pdsh "yum install -y pdsh"
management # pdsh "scp management:/etc/dsh/all /etc/dsh/all"
management # pdsh "cat /etc/dsh/all"
host1: host1
host1: host2
host2: host1
host2: host2
management # pdsh "scp management:/etc/profile.d/pdsh.sh
/etc/profile.d/pdsh.sh"
management # pdsh "cat /etc/profile.d/pdsh.sh"
host1: export WCOLL=/etc/dsh/all
host2: export WCOLL=/etc/dsh/all
```

LVM-based file system for HDFS on all data nodes

The HDFS is created on the ext4 LVM-based file system. For better performance, the partition is mounted with the `noatime` option (disables `atime` updates for read access) to reduce disk I/O. The `noatime` option is added to the `/etc/fstab` path (Example 2-6) for persistent configuration.

Example 2-6 Adding the noatime option to /etc/fstab

```
# cat /etc/fstab
/dev/mapper/system-root /                   ext4 defaults        1 1
UUID=****************** /boot               ext3 defaults        1 2
/dev/mapper/system-data /data               ext4 defaults,noatime 1 2
tmpfs                   /dev/shm            tmpfs defaults       0 0
devpts                  /dev/pts            devpts gid=5,mode=620  0 0
sysfs                   /sys                sysfs defaults       0 0
proc                    /proc               proc    defaults       0 0
```

The settings are changed within the file system that is mounted (Example 2-7).

Example 2-7 Remounted file system with the noatime option

```
# mount |grep data
/dev/mapper/system-data on /data type ext4 (rw)
# mount -o remount,noatime /data
# mount |grep data
/dev/mapper/system-data on /data type ext4 (rw,noatime)
```

Customizing RHEL services

Disable some RHEL services (NTP, firewall, and SELINUX) in exchange for better performance:

1. Configure the NTP service as shown in Example 2-8.

   Example 2-8 NTP service configuration file

   ```
   # cat /etc/ntp.conf
   server firstserver
   driftfile /var/lib/ntp/drift
   disable auth
   restrict 127.0.0.1
   server secondserver
   ```

2. Disable SELINUX as shown in Example 2-9. Restart the system to make the changes take effect.

   Example 2-9 Disabling the SELINUX in the /etc/selinux/config file

   ```
   # cat /etc/selinux/config
   # This file controls the state of SELinux on the system.
   # SELINUX= can take one of these three values:
   #       enforcing - SELinux security policy is enforced.
   #       permissive - SELinux prints warnings instead of enforcing.
   #       disabled - SELinux is fully disabled.
   SELINUX=disabled
   # SELINUXTYPE= type of policy in use. Possible values are:
   #       targeted - Only targeted network daemons are protected.
   #       strict - Full SELinux protection.
   SELINUXTYPE=targeted
   ```

3. Disable the firewall for both IPv4 and IPv6 to improve performance (Example 2-10).

   Example 2-10 Disabling the firewall services (startup and currently running)

   ```
   # chhkconfig iptables off
   # chhkconfig ip6tables off
   # service iptables stop
   # service ip6tables stop
   ```

4. Disable the IPv6 feature that is required for the tests.

Verifying whether all required packages are installed

Run through all requirements on all nodes. To determine the software requirements for InfoSphere BigInsights, see *Implementing IBM InfoSphere BigInsights on System x*, SG24-8077. Also, install **`expect`**. For RHEL 6.2, we installed `expect-5.44.1.15-2.el6.x86_64`.

Creating users (same user configuration across all nodes)

Create the egoadmin user and group with the same ID on all nodes (Example 2-11). We use this user to install IBM Platform Symphony Advanced Edition and InfoSphere BigInsights.

Example 2-11 Creating the egoadmin user across all nodes

```
management # useradd -u 1000 -m egoadmin
management # pdsh "useradd -u 1000 -m egoadmin"
```

After the integration, you might need to create more users to work with the integrated software. For more information, see 2.8, “Adding users” on page 43.

Temporary security changes

During the InfoSphere BigInsights installation, complete the security changes to allow the egoadmin user to run programs with the security privileges of root before proceeding. Using the **visudo** command, make the following changes:

1. Add the following line:

   ```
   egoadmin ALL=(ALL) NOPASSWD: ALL
   ```

2. Change `Defaults requiretty` to `Defaults !requiretty`.

2.3 Configuring InfoSphere BigInsights

Set up InfoSphere BigInsights on 10 data nodes and 1 management node:

> **More information:** For a reference about the requirements and installation procedures for the InfoSphere BigInsights configuration, see the *Implementing IBM InfoSphere BigInsights on System x*, SG24-8077.

1. Start the web-based GUI installation setup on the management node, and generate the XML file based on the preferences chosen:
   a. In the Welcome window, click **Next**.
   b. Select **I accept the terms in the license agreement**, and click **Next**.
   c. Select **Create a response file without performing an installation**, and click **Next**.
2. Review the generated XML, and change it according to the desirable requirements.

   InfoSphere BigInsights is installed (from the management node) by using the modified XML without errors (highlighted in a bold font in Example 2-12 on page 26). The original XML file is generated by using the GUI (see Appendix A, “XML installation file for IBM InfoSphere BigInsights” on page 117).

   The host name of the cluster node in the test configuration is set to the high-performance network interface. For example, instead of setting i05i04 to use the Gigabit Ethernet network, set i05i04 to use IPoIB.

Example 2-12 InfoSphere BigInsights modified fullinstall.xml file parts used after GUI generation

```
<?xml version="1.0" encoding="UTF-8"?>
<cluster-configuration>
    <operation>install</operation>
    <vendor>ibm</vendor>
[...]
<hadoop>
         <datanode>
            <selection-type>Specified</selection-type>

<nodes>i05i06,i05i07,i05i08,i05i09,i05i10,i05i14,i05i15,i05i16,i05i17,i05i18</n
odes>
            <datanode-port>50010</datanode-port>
            <datanode-ipc-port>50020</datanode-ipc-port>
            <datanode-http-port>50075</datanode-http-port>
            <tasktracker-http-port>50060</tasktracker-http-port>
```

```
            <data-directory>/data/hdfs</data-directory>
         </datanode>
</hadoop>
[...]
<HBase>
        <configure>true</configure>
        <zookeeper-mode>shared</zookeeper-mode>
        <master-nodes>i05i04</master-nodes>
        <install-mode>fully</install-mode>

<region-nodes>i05i06,i05i07,i05i08,i05i09,i05i10,i05i14,i05i15,i05i16,i05i17,i0
5i18</region-nodes>
        <root-directory>/opt/ibm/bi/hbase</root-directory>
        <log-directory>/var/log/bi/hbase/logs</log-directory>
        <master-port>60000</master-port>
        <master-ui-port>60010</master-ui-port>
        <regionserver-port>60020</regionserver-port>
        <regionserver-ui-port>60030</regionserver-ui-port>
</HBase>
<node-list>
        <node>
            <name-or-ip>i05i04</name-or-ip>
            <password>{xor}</password>
            <rack/>
            <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
            <gpfs-node-designation/>
            <gpfs-admin-node/>
            <gpfs-rawdisk-list/>
        </node>
        <node>
            <name-or-ip>i05i06</name-or-ip>
            <password>{xor}</password>
            <rack/>
            <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
            <gpfs-rawdisk-list/>
        </node>
        <node>
            <name-or-ip>i05i07</name-or-ip>
            <password>{xor}</password>
            <rack/>
            <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
            <gpfs-rawdisk-list/>
        </node>
        <node>
            <name-or-ip>i05i08</name-or-ip>
            <password>{xor}</password>
            <rack/>
            <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
            <gpfs-rawdisk-list/>
        </node>
        <node>
            <name-or-ip>i05i09</name-or-ip>
            <password>{xor}</password>
            <rack/>
            <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
```

```
            <gpfs-rawdisk-list/>
        </node>
        <node>
            <name-or-ip>i05i10</name-or-ip>
            <password>{xor}</password>
            <rack/>
            <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
            <gpfs-rawdisk-list/>
        </node>
        <node>
            <name-or-ip>i05i14</name-or-ip>
            <password>{xor}</password>
            <rack/>
            <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
            <gpfs-rawdisk-list/>
        </node>
        <node>
            <name-or-ip>i05i15</name-or-ip>
            <password>{xor}</password>
            <rack/>
            <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
            <gpfs-rawdisk-list/>
        </node>
        <node>
            <name-or-ip>i05i16</name-or-ip>
            <password>{xor}</password>
            <rack/>
            <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
            <gpfs-rawdisk-list/>
        </node>
        <node>
            <name-or-ip>i05i17</name-or-ip>
            <password>{xor}</password>
            <rack/>
            <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
            <gpfs-rawdisk-list/>
        </node>
        <node>
            <name-or-ip>i05i18</name-or-ip>
            <password>{xor}</password>
            <rack/>
            <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
            <gpfs-rawdisk-list/>
        </node>
</node-list>
```

3. Run the **healthcheck.sh** script inside `$HADOOP_HOME/bin` directory, and validate whether all components are working as expected (no FAILED components). Example 2-13 shows the expected result at the end of the script.

Example 2-13 Expected end of output for the helthcheck.sh script

```
[...]
[INFO] Progress - 100%
```

```
[INFO] DeployManager - Start; SUCCEEDED components: [guardiumproxy, zookeeper,
hadoop, derby, hive, hbase, flume, oozie, orchestrator, jaqlserver, console,
sheets]; FAILED components: []
```

2.4 Installing IBM Platform Symphony Advanced Edition

To install IBM Platform Symphony Advanced Edition:

1. On the master node, edit the `/etc/bashrc` file, and add the following environment variables:

   ```
   export HADOOP_HOME=/opt/ibm/bi/IHC
   export JAVA_HOME=/opt/ibm/bi/jdk
   export PATH=$PATH:$HADOOP_HOME/bin
   export CLUSTERNAME=<cluster_name>
   export CLUSTERADMIN=egoadmin
   export HADOOP_VERSION=1_0_0
   export HDFS_URL=http://<HDF_NAME_NODE>:50070
   export SIMPLIFIEDWEM=N
   ```

   The HADOOP_HOME and JAVA_HOME environment variables are defined for the integration with InfoSphere BigInsights that is being installed in the `/opt/ibm/bi` directory. After you edit the file, copy it to all the nodes in the cluster (the host names that are listed in the `$WCOLL` file) using the **pdcp** command:

   ```
   pdcp /etc/bashrc /etc/bashrc
   ```

2. On all nodes, restart any currently open shells for the new environment variables to take effect.
3. On the master node, change directory (**cd**) to the directory where the IBM Platform Symphony installation images were downloaded. Install IBM Platform Symphony:

   ```
   ./symSetup5.2.0_lnx26-lib23-x64.bin --dbpath /opt/ibm/sym52/rpmdb --prefix
   /opt/ibm/sym52 --quiet
   ```

   The default port 8080 used by the Platform Management Consoles (PMC) is already being used by the BigInsight console.

4. On the master node, change the PMC port to the value `"18080"` (including the quotation marks) in the `$EGO_TOP/gui/conf/server.xml` file.

   Example 2-14 shows the original `server.xml` file.

   Example 2-14 Original server.xml file

   ```
   <!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
       <Connector port="8080" maxHttpHeaderSize="8192"
                  maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
                  enableLookups="false" redirectPort="8443" acceptCount="100"
                  connectionTimeout="20000" disableUploadTimeout="true" />
   ```

   Example 2-15 shows the edited `server.xml` file.

   Example 2-15 Changed server.xml file

   ```
   <!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
       <Connector port="18080" maxHttpHeaderSize="8192"
                  maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
                  enableLookups="false" redirectPort="8443" acceptCount="100"
                  connectionTimeout="20000" disableUploadTimeout="true" />
   ```

5. Initiate the master node, set the entitlement file, and start IBM Platform Symphony:

   > **Entitlement file:** Copy the entitlement file to a location where only the root user has access and it does not change when IBM Platform Symphony needs it for installation. In this example, we used the IBM Platform Symphony `$EGO_TOP` installation directory.

   ```
   su - egoadm
   . $EGO_TOP/profile.platform
   egoconfig join <clustername>
   egoconfig setentitlement $EGO_TOP/sym.entitlement
   exit
   . $EGO_TOP/profile.platform
   egosh ego start
   ```

6. On the compute nodes, install IBM Platform Symphony:

   ```
   /opt.db06b04/FILES/symcompSetup5.2.0_lnx26-lib23-x64.bin --dbpath
   /opt/ibm/sym52/pmrdb --prefix /opt/ibm/sym52 --quiet
   ```

7. On all nodes, change the IBM Platform Symphony configuration to enable **egosh** commands to use SSH connections (with no password) instead of RSH connections. Add the following line to the `$EGO_CONFDIR/ego.conf` file as shown in Example 2-16:

   ```
   EGO_RSH="ssh -o 'PasswordAuthentication no' -o 'StrictHostKeyChecking no'"
   ```

   Example 2-16 Line added at the end of the $EGO_CONFDIR/ego.conf file

   ```
   [...]
   EGO_VERSION=1.2.6

   EGO_RSH="ssh -o 'PasswordAuthentication no' -o 'StrictHostKeyChecking no'"
   ```

8. Initiate the compute nodes and start IBM Platform Symphony:

   ```
   su - egoadm
   . /opt/ibm/sym52/profile.platform
   egoconfig join <master_node_hostname>
   exit
   . /opt/ibm/sym52/profile.platform
   egosh ego start
   ```

   If the cluster administrator egoadmin is granted root privileges, the user egoadmin can start and shut down IBM Platform Symphony on any hosts in the cluster.

9. On all nodes, as root, run the **$EGO_ESRVDIR/egosetsudoers.sh** command

10. Confirm the command:

    ```
    cat /etc/ego.sudoers
    ```

11. On the master node, add the following line to the `/etc/bashrc` file:

    ```
    . /opt/ibm/sym52/profile.platform
    ```

12. Copy the modified file to the other nodes in the cluster:

    ```
    pdcp /etc/bashrc /etc/bashrc
    ```

2.5 Integrating IBM Platform Symphony and InfoSphere BigInsights

When you integrate IBM Platform Symphony and InfoSphere BigInsights, perform the following steps on the master node and on the computing nodes of the cluster.

> **Prerequisite:** Before you integrate IBM Platform Symphony and InfoSphere BigInsights, install the **pdsh** utility on all nodes so that you can run the same commands across multiple nodes.

To integrate IBM Platform Symphony and InfoSphere BigInsights:

1. Stop InfoSphere BigInsights and IBM Symphony on the cluster:

```
su - egoadmin
/opt/ibm/bi/bin/stop-all.sh
$EGO_BINDIR/egoshutdown.sh
exit
```

2. On the master node, edit the `/opt/ibm/sym52/soam/mapreduce/conf/pmr-env.sh` file. Check that the following values are in the file:

```
export JAVA_HOME=/opt/ibm/bi/jdk
export HADOOP_VERSION=1_0_0
export HADOOP_HOME=/opt/ibm/bi/IHC
export PMR_EXTERNAL_CONFIG_PATH=/opt/ibm/bi/hadoop-conf
export JVM_OPTIONS=-Xmx1024m
export PMR_SERVICE_DEBUG_PORT=
export PMR_MRSS_SHUFFLE_CLIENT_PORT=7879
export PMR_MRSS_SHUFFLE_DATA_WRITE_PORT=7881
export PYTHON_PATH=/bin:/usr/bin:/usr/local/bin
export PATH=${PATH}:${JAVA_HOME}/bin:${PYTHON_PATH}
export
USER_CLASSPATH=/opt/ibm/bi/IHC/lib/*:/opt/ibm/bi/jaql/*:/opt/ibm/bi/lib/*:/opt/
ibm/bi/jaql/modules/systemT/lib/systemT-jaql.jar:/opt/ibm/bi/textanalytics/lib/
text-analytics/systemT.jar:/opt/ibm/bi/text-analytics/lib/htmlparser-2.0/htmlpa
rser.jar
export
JAVA_LIBRARY_PATH=/opt/ibm/bi/IHC/lib/native/Linux-amd64-64/:/opt/ibm/bi/IHC/li
b/native/Linux-i386-32/
export CLOUDERA_HOME=/opt/ibm/bi/IHC
```

3. After you edit the file on the master node, copy the file to the other nodes in the cluster by using the **pdcp** command:

```
pdcp /opt/ibm/sym52/soam/mapreduce/conf/pmr-env.sh
/opt/ibm/sym52/soam/mapreduce/conf/pmr-env.sh
```

4. Apply the `BIIntegrationPatch-201206291122.tar.gz` patch by using the **patchHost** shell script. In this script, the *BI_Integration_Patch* variable defines the absolute path for the patch file. This example uses the `/tmp/BIIntegrationPatch-201206291122.tar.gz` path. Example 2-17 shows the **patchHost** script.

Example 2-17 The patchHost script

```
#!/bin/bash
#
# patchHost: Run the Big Insights / Symphony Integration patch
```

```
#
#  Change these values for your site
admin_id="egoadmin"
symdir="/opt/ibm/sym52"
bidir="/opt/ibm/bi"
BI_Integration_Patch="/tmp/BIIntegrationPatch-201206291122.tar.gz"

#  This needs to be run as root
if [ "$(id -un)" != "$admin_id" ]; then
   echo "This script must be run as $admin_id"
   exit 1
fi

if [ ! -f $symdir/profile.platform ] ; then
   echo "This symphony profile ($symdir/profile.platform) is missing "
   echo "Is symphony installed at $symdir?"
   exit 1
fi

. $symdir/profile.platform

if [ ! -f $BI_Integration_Patch ] ; then
   echo "This Symphony/BigInsights patch file($BI_Integration_Patch) is missing"
   exit 1
fi
if [ ! -d "$bidir" ] ; then
   echo "The Big Insights directory ($bidir)does not exist "
   exit 1
fi
if [ ! -d "$HADOOP_HOME" ] ; then
   echo "The HADOOP_HOME directory does not exist or HADOOP_HOME is not set"
   echo "HADOOP_HOME=$HADOOP_HOME"
   exit 1
fi
if [ -d "$symdir/bi_integration" ] ; then
   echo "The directory $symdir/bi_integration ALREADY exists. "
   echo "You cannot run this script twice"
   exit 1
fi
if [ ! -f "$HADOOP_HOME/hadoop-core-1.0.0.jar" ] ; then
  echo "File $HADOOP_HOME/hadoop-core-1.0.0.jar is missing"
  exit 1
fi
if [ -l "$HADOOP_HOME/hadoop-core-1.0.0.jar" ] ; then
  echo "File $HADOOP_HOME/hadoop-core-1.0.0.jar is a link"
  exit 1
fi

echo tar -C $symdir -xvzf $BI_Integration_Patch
tar -C $symdir -xvzf $BI_Integration_Patch
echo $EGO_TOP/bi_integration/jar_integration.sh
$EGO_TOP/bi_integration/jar_integration.sh
echo mv $HADOOP_HOME/hadoop-core-1.0.0.jar $HADOOP_HOME/hadoop-core-1.0.0.jar.ORIG
mv $HADOOP_HOME/hadoop-core-1.0.0.jar $HADOOP_HOME/hadoop-core-1.0.0.jar.ORIG
echo ln -sf $EGO_TOP/bi_integration/IBM-pmr-hadoop-1.0.0.jar $HADOOP_HOME/hadoop-core-1.0.0.jar
```

```
ln -sf $EGO_TOP/bi_integration/IBM-pmr-hadoop-1.0.0.jar $HADOOP_HOME/hadoop-core-1.0.0.jar
echo ln -sf $PMR_HOME/conf/pmr-site.xml $bidir/hadoop-conf/
ln -sf $PMR_HOME/conf/pmr-site.xml $bidir/hadoop-conf/
hlibdir=$symdir/soam/mapreduce/5.2/linux2.6-glibc2.3-x86_64/lib/hadoop-1.0.0

echo mv $hlibdir/jackson-core-asl-1.0.1.jar $hlibdir/jackson-core-asl-1.0.1.jar.bak
mv $hlibdir/jackson-core-asl-1.0.1.jar $hlibdir/jackson-core-asl-1.0.1.jar.bak
echo ln -sf $bidir/sheets/libext/jackson-core-asl-1.5.5.jar $hlibdir/jackson-core-asl-1.0.1.jar
ln -sf $bidir/sheets/libext/jackson-core-asl-1.5.5.jar $hlibdir/jackson-core-asl-1.0.1.jar
echo mv $hlibdir/jackson-mapper-asl-1.0.1.jar $hlibdir/jackson-mapper-asl-1.0.1.jar.bak
mv $hlibdir/jackson-mapper-asl-1.0.1.jar $hlibdir/jackson-mapper-asl-1.0.1.jar.bak
echo ln -sf $bidir/sheets/libext/jackson-mapper-asl-1.5.5.jar
$hlibdir/jackson-mapper-asl-1.0.1.jar
ln -sf $bidir/sheets/libext/jackson-mapper-asl-1.5.5.jar $hlibdir/jackson-mapper-asl-1.0.1.jar
```

5. On the master node, run the script as follows:

   a. Switch to the egoadmin user:

   ```
   su - egoadmin
   ```

   b. Run the script:

   ```
   ./patchHost
   ```

6. After the script runs, run it on the other nodes in the cluster by running the following commands on the master node:

   a. Copy the script (patchHost) and the patch to the other hosts:

   ```
   pdcp /tmp/patchHost /tmp/patchHost
   pdcp /tmp/Install/Symphony/patch/BIIntegrationPatch-201206291122.tar.gz
   /tmp/Install/Symphony/patch/BIIntegrationPatch-201206291122.tar.gz
   ```

   b. Run the **`patchHost`** script:

   ```
   pdsh /tmp/patchHost
   ```

7. Run the following commands on the master node to apply control shims:

   ```
   ln -sf $PMR_HOME/conf/pmr-site.xml
   /opt/ibm/bi/console/wtiruntime/dscomponents/enterpriseconsole/eclipse/plugins/c
   om.ibm.hadoop_1.0.0.v20120605_0341/
   ln -sf $EGO_TOP/bi_integration/IBM-pmr-hadoop-1.0.0.jar
   /opt/ibm/bi/console/wtiruntime/dscomponents/enterpriseconsole/eclipse/plugins/c
   om.ibm.hadoop_1.0.0.v20120605_0341/hadoop-core-1.0.0.jar
   ln -sf /opt/ibm/sym52/bi_integration/IBM_MANIFEST.MF
   /opt/ibm/bi/console/wtiruntime/dscomponents/enterpriseconsole/eclipse/plugins/c
   om.ibm.hadoop_1.0.0.v20120605_0341/META-INF/MANIFEST.MF
   ```

8. Run the following commands to add the oozie integration override:

   ```
   ln -s /opt/ibm/bi/IHC/lib/servlet-api-2.5-6.1.14.jar \
   /opt/ibm/bi/oozie/wasce/lib/endorsed
   ```

9. Modify the `/opt/ibm/bi/hive/bin/hive` file on the master node, and then copy it to the other nodes:

   a. Back up the old file:

   ```
   cp /opt/ibm/bi/hive/bin/hive /opt/ibm/bi/hive/bin/hive.ORIG
   ```

   b. Open the file and search for the *`hadoop_version_re`* variable. Replace its value with the `"^([[:digit:]]+)[\.\_]([[:digit:]]+)([\.\_]([[:digit:]]+))?.*$"` value (including the quotation marks).

c. To confirm the changes, enter the **diff** command to see the results:

```
$ diff hive.ORIG hive
240c240
< hadoop_version_re="^([[:digit:]]+)\.([[:digit:]]+)(\.([[:digit:]]+))?.*$"
---
>
hadoop_version_re="^([[:digit:]]+)[\.\_]([[:digit:]]+)([\.\_]([[:digit:]]+))
?.*$"
```

d. Copy the modified file to the other nodes in the cluster:

```
pdcp /opt/ibm/bi/hive/bin/hive /opt/ibm/bi/hive/bin/hive
```

10.Start InfoSphere BigInsights and IBM Platform Symphony:

```
su - egoadmin
/opt/ibm/bi/bin/start-all.sh
$EGO_BINDIR/egosh ego start -f all
```

11.Stop Jobtracker and Tasktracker:

```
su - egoadmin
/opt/ibm/bi/IHC/bin/stop-mapred.sh
```

2.6 Additional configuration for IBM Platform Symphony

After the integration, you must make some additional changes. These changes depend on the cluster size and complexity. Therefore, the values might be different for larger clusters. You have the following options:

- Identifying the management nodes in the IBM Platform Symphony configuration
- TCP/IP performance tuning

2.6.1 Identifying the management nodes in the IBM Platform Symphony configuration

In the IBM Platform Symphony 5.2 installation, the management nodes are incorrectly identified. Therefore, the default configuration uses all nodes as compute nodes. There are two ways you can change this setting:

- Manually select which nodes are the compute nodes.
- Change the configuration and specify which nodes are the management nodes (recommended solution).

To change the configuration and specify the management nodes:

1. Edit the `$EGO_CONFDIR/ego.cluster.`*`cluster_name`* file.
2. For all management nodes, add the `mg` tag inside the resources column as illustrated in Example 2-18. The *`cluster_name`* is the name of the symphony cluster. You can check the name by using the **`egosh ego info`** command (under the user egoadmin).

Example 2-18 The ego.cluster.<cluster_name> file with i05i04 as management node

```
# $Id: TMPL.ego.cluster,v 1.3.28.3.2.1.2.1.92.4.2.1.18.4 2012/05/22 09:38:44
yaoliu Exp $
#-----------------------------------------------------------------------
# T H I S   I S   A   O N E   P E R   C L U S T E R   F I L E
#
```

```
# This is a sample ego cluster definition file.  This file's name
# should be ego.cluster.<cluster-name>.
#

Begin   ClusterAdmins
Administrators = egoadmin
End    ClusterAdmins

Begin Host
HOSTNAME  model    type        r1m  mem  swp  RESOURCES    #Keywords
i05i04    !        !            -    -    -   (linux mg)
#lemon    PC200    LINUX86     3.5  1    2   (linux)
#plum     !        NTX86       3.5  1    2   (nt)
End     Host
```

Example 2-19 shows the output of this command.

Example 2-19 Checking the cluster name and master node for IBM Platform Symphony

```
[egoadmin@i05i04 ~]$ egosh ego info
Cluster name            : bisym
EGO master host name    : i05i04
EGO master version      : 1.2.6
```

3. After you identify all management nodes and change the required lines in the `$EGO_CONFDIR/ego.cluster.`*`cluster_name`* file, bring down all changed management nodes.

2.6.2 TCP/IP performance tuning

Some TCP/IP parameters can help tune the IBM Platform Symphony environment and possibly improve workload performance. The values that are shown were tested in the cluster for the tests that are documented in this book. As a result, you might need to adjust them for the cluster sizes, architectures, and operating system in your environment.

Change the **`sysctl`** attributes on all nodes, and add them to the `/etc/sysctl.conf` file:

```
sysctl -w net.ipv4.ipfrag_low_thresh=1048576
sysctl -w net.ipv4.ipfrag_high_thresh=8388608
```

If you have InfiniBand adapters and installed the InfiniBand drivers, you might have the **`/sbin/sysctl_perf_tuning`** script. The package name that was installed in the test environment was named `kernel-ib-1.5.3.2-2.6.32_220.el6.x86_64.x86_64` and represents the InfiniBand driver and ULP kernel modules.

If you have the **`/sbin/sysctl_perf_tuning`** script:

1. Run the RHEL 6.2 **`sysctl_perf_tuning`** script on all nodes. The script persistently changes the values (Example 2-20).

Example 2-20 System parameters that changed with the /sbin/sysctl_perf_tuning command

```
/sbin/sysctl -q -w net.ipv4.tcp_timestamps=0
/sbin/sysctl -q -w net.ipv4.tcp_sack=0
/sbin/sysctl -q -w net.core.netdev_max_backlog=250000
/sbin/sysctl -q -w net.core.rmem_max=16777216
/sbin/sysctl -q -w net.core.wmem_max=16777216
/sbin/sysctl -q -w net.core.rmem_default=16777216
```

```
/sbin/sysctl -q -w net.core.wmem_default=16777216
/sbin/sysctl -q -w net.core.optmem_max=16777216
/sbin/sysctl -q -w net.ipv4.tcp_mem="16777216 16777216 16777216"
/sbin/sysctl -q -w net.ipv4.tcp_rmem="4096 87380 16777216"
/sbin/sysctl -q -w net.ipv4.tcp_wmem="4096 65536 16777216"
```

2. After you run the script, if you are not satisfied with the performance results, use the script example lines to manually change the value again through all nodes. Then, check for improvements.

2.7 Benchmark tests

This section presents the benchmark results before and after we integrated IBM Platform Symphony Advanced Edition 5.2 with InfoSphere BigInsights 1.4. All benchmark results were obtained by using the following test environment for both hardware and software:

- Eleven hardware machines were used as described in 2.2, “Environment” on page 21. One machine was for the management node, and ten machines were for the data nodes of InfoSphere BigInsights (and the same ten for compute nodes on IBM Platform Symphony).
- All systems run RHEL 6.2 (x86_64) with InfoSphere BigInsights 1.4 and IBM Platform Symphony Advanced Edition 5.2.
- All configurations after this section used InfiniBand (IPoIB) as the network for InfoSphere BigInsights (Hadoop data) and Ethernet for the IBM Platform Symphony.

The following sections describe the tests and their specific configurations:

- Default configuration, which uses the tasks that are described in 2.6.1, “Identifying the management nodes in the IBM Platform Symphony configuration” on page 34, but *not* the changes in 2.6.2, “TCP/IP performance tuning” on page 35.
- Tuned configuration (with TCP tuning), which uses the tasks that are described in 2.6.1, “Identifying the management nodes in the IBM Platform Symphony configuration” on page 34, *and* the changes in 2.6.2, “TCP/IP performance tuning” on page 35.
- Preloaded configuration (with TCP tuning and prestart option enabled), which is the same as the tuned configuration, but with the Prestart option enabled on IBM Platform Symphony.

> **Tuned and preloaded configurations:** To reduce the debug level, you can optionally add the following line (Example 2-21 on page 39):
>
> `-Dmapred.map.child.log.level=WARN`

2.7.1 Test environment and monitoring tools

This section highlights the tuning options and benchmark results from the tests in the stand-alone setup of InfoSphere BigInsights 1.4 and after its integration with IBM Platform Symphony 5.2.

During the benchmarking tests, the following tools and web pages were used to extract the results and to help monitor the progress of each test:

- NMON
- The **`time`** command from RHEL 6.2 (**`time-1.7-37.1.el6.x86_64`**)
- The **`script`** command from RHEL 6.2 (**`util-linux-ng-2.17.2-12.4.el6.i686`**)

- InfoSphere BigInsights cluster web page:
  - HDFS usage:

    `http://i05n04.pbm.ihost.com:50070/dfshealth.jsp`
- IBM Platform Symphony cluster web pages:
  - Cluster information

    `http://i05n04.pbm.ihost.com:18080/platform/dealUserLogin.do`
  - MapReduce information

    `http://i05n04.pbm.ihost.com:18080/pmrgui/framework/main.action`

2.7.2 Results of the stand-alone benchmark for InfoSphere BigInsights V1.4

The following benchmark tests were conducted over InfiniBand:

- Sleep benchmark
- Pi estimate benchmark

Sleep benchmark

The sleep benchmark (scheduling benchmark) tests the performance of the scheduler. It examines how quickly the scheduler can create tasks after each other and how efficiently the scheduler can detect where (which node) to create. In this case, the scheduler is the InfoSphere BigInsights scheduler. The following command is issued:

```
time hadoop jar $HADOOP_HOME/../pig/test/e2e/pig/lib/hadoop-examples.jar sleep -mt
1 -rt 1 -m <tasks> -r 1
```

Where `<tasks>` is the number of map tasks to be created.

Table 2-1 shows the execution time for the sleep tests.

Table 2-1 Execution time for the sleep tests

Number of map tasks	Number of runs	Time observed[a]		Performance in tasks/second	
		Minimum	Maximum	Minimum	Maximum
1000	5	28s	33s	30/30	35/71
5000	5	1m 31s	1m 38s	51/02	54/95
10000	1		2m 53s	57/80	
50000	1		13m 41s	60/90	
100000	2	Crashed or hung twice			

a. +/- 0.5 seconds error

Pi estimate benchmark

Table 2-2 on page 38 shows the execution time for the Pi estimate tests (mixed I/O read and processor intensive benchmarks). The PI estimate benchmark uses the following command:

```
time hadoop jar $HADOOP_HOME/hadoop-examples-1.0.0.jar pi <nMaps> <nSamples>
```

Where `<nMaps>` is the number of map tasks to be created, and `<nSamples>` is the number of iterations for each task (will consume more processor time).

Table 2-2 Execution time for the Pi estimate tests

Number of map tasks	Number of samples per map	Pi estimate time in seconds (without input generation)	Complete job time observed (minimum[a])	Pi result
10000	10000000	221.669	7m 35s	3.14159265584
1000	10000000	36.538	1m 03s	3.1415926452
	100000000	80.404	1m 45s	3.14159265572
100	10000000	17.497	22s	3.141592736
	100000000	28.85	33s	3.1415926492
10	10000000	17.421	19s	3.14159256
	100000000	29.412	31s	3.141592744
1	100	17.592	19s	3.2
	10000	17.424	19s	3.1408
	100000	16.431	18s	3.1412
	1000000	16.448	18s	3.141552
	10000000	18.433	20s	3.1415844
	100000000	25.448	27s	3.14159256
	1000000000	87.621	1m29s	3.14159272

a. +/- 0.5 seconds error

2.7.3 Results of integration for IBM Platform Symphony V5.2 and InfoSphere BigInsights V1.4

After the integration of IBM Platform Symphony V5.2 with InfoSphere BigInsights V1.4, the same tests as in 2.7.2, “Results of the stand-alone benchmark for InfoSphere BigInsights V1.4” on page 37, were conducted. They measured the performance difference of the IBM Platform Symphony scheduler against the InfoSphere BigInsights scheduler.

The following benchmark tests were conducted:

- Sleep benchmark
- Pi estimate benchmark (mixed I/O read and processor-intensive benchmark)

Sleep benchmark

After the integration, the InfoSphere BigInsights scheduler is no longer used. Now the IBM Platform Symphony scheduler is used. Table 2-3 shows the execution time for the sleep tests (scheduling benchmark).

Table 2-3 Execution time for the sleep tests (default configuration)

Number of map tasks	Number of runs	Time observed[a]		Performance in tasks/second	
		Minimum	Maximum	Minimum	Maximum
1000	10	8s	12s	83/33	125
5000	10	13s	14s	357/14	384/62

Number of map tasks	Number of runs	Time observed[a]		Performance in tasks/second	
		Minimum	Maximum	Minimum	Maximum
10000	10	15s	22s	454/55	666/67
50000	10	1m 04s	1m 53s	442/47	781/25
100000	5	3m 01s	3m 38s	458/72	552/49
200000	1		18m 52s	176/68	

a. +/- 0.5 seconds error

After we change the values as shown in 2.6.2, “TCP/IP performance tuning” on page 35, the tests that are listed in Table 2-3 are repeated.

Table 2-4 shows the improvements.

Table 2-4 Execution time for sleep test (tuned configuration)

Number of map tasks	Number of runs	Time observed [a]		Performance in tasks/second	
		Minimum	Maximum	Minimum	Maximum
1000	10	7s	11s	90/91	142/86
5000	10	9s	12s	416/67	555/56
10000	10	12s	16s	625	833/34
50000	10	57s	1m 14s	675/68	877/19
100000	5	2m 49s	3m 14s	515/47	591/72
200000	2	19m 04s	22m 39s	147/17	174/83

a. +/- 0.5 seconds error

In these series of tests, the debug level was reduced by adding an optional flag attribute to the command line **`-Dmapred.map.child.log.level=WARN`** as shown in Example 2-21.

Example 2-21 Command line text to reduce debug level

```
# time hadoop jar $HADOOP_HOME/../pig/test/e2e/pig/lib/hadoop-examples.jar sleep
-Dmapred.map.child.log.level=WARN -mt 1 -rt 1 -m 5000 -r 1
```

Enabling the Prestart option

If the Prestart option is enabled for the MapReduce 5.2 application, it can improve the execution time. To enable the Prestart option:

1. Go to the following address:

   `http://<master_node_ip>:18080/platform`

2. Click the **MapReduce Workload** link on the left side of the window.
3. In the IBM Platform Symphony window, on the **Workload** tab, click **MapReduce Applications**, select the **MapReduce 5.2** application, and then, click the **Modify** button.

4. In the Application window (Figure 2-3), select the **Pre-start application** check box, and then click the **Save** button.

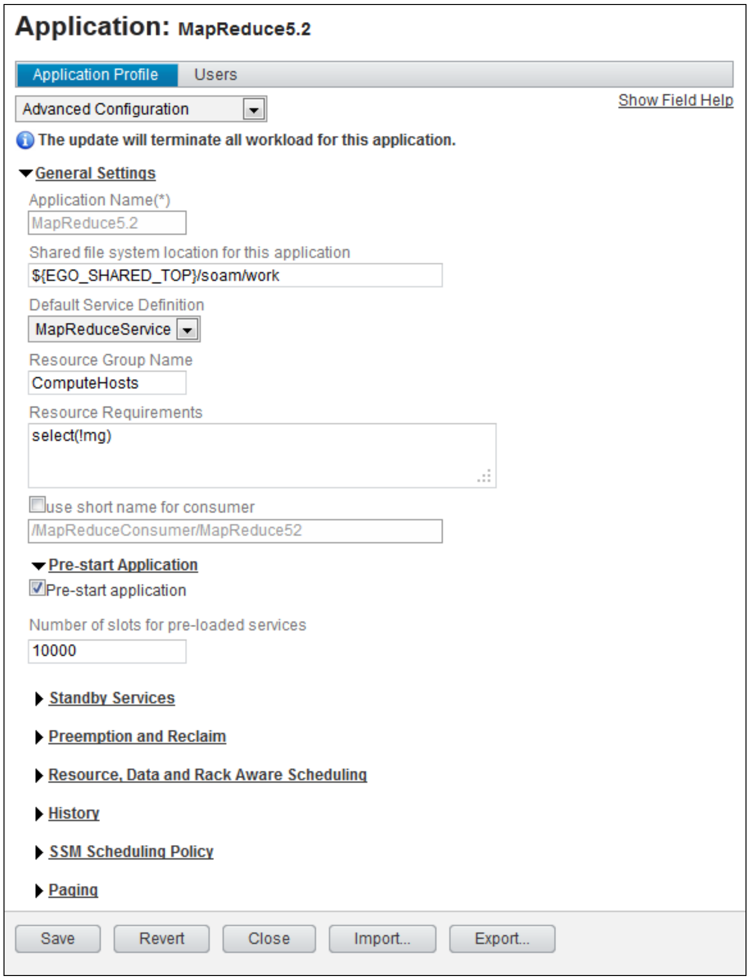


Figure 2-3 Pre-start option under MapReduce5.2 application properties

Table 2-5 shows the results of this change and of using the command line option to reduce the debug output as shown in Example 2-21 on page 39.

Table 2-5 Execution time for sleep test (tuned configuration and the pre-start option)

		Time observed[a]		Performance in tasks/second	
Number of map tasks	**Number of runs**	**Minimum**	**Maximum**	**Minimum**	**Maximum**
1000	10	5s	7s	142/86	200
5000	10	9s	9s	555/56	555/56
10000	10	13s	14s	714/29	769/23
50000	10	59s	1m 07s	746/27	847/46
100000	3	3m11s	3m 52s	431/03	523/56

a. +/- 0.5 seconds error

Pi estimate benchmark (mixed I/O read and processor-intensive benchmark)

After the integration, we no longer use the InfoSphere BigInsights scheduler. Now we use the IBM Platform Symphony scheduler. Table 2-6 shows the execution time for the Pi estimate tests.

Table 2-6 Execution time for Pi estimate tests (default configuration)

Number of map tasks	Number of samples per map	Pi estimate time in seconds (without input generation)	Complete job time observed (minimum[a])	Pi result
100000	10000000	519.835	50m 34s	3.141592655176
	100000000	3277.363	105m 45s	3.1415926543392
10000	10000000	54.429	5m 09s	3.14159265584
	100000000	351.059	10m 14s	3.141592654996
1000	10000000	17.0	44s	3.1415926452
	100000000	54.892	1m 22s	3.14159265572
100	10000000	8.159	12s	3.141592736
	100000000	18.055	22s	3.1415926492
	1000000000	94.077	1m 38s	3.1415926568
10	10000000	9.803	11s	3.14159256
	100000000	16.134	18s	3.141592744
	1000000000	83.071	1m 25s	3.1415926644
1	100	9.763	11s	3.2
	10000	8.771	10s	3.1408
	100000	9.744	11s	3.1412
	1000000	6.731	8s	3.141552
	10000000	5.745	7s	3.1415844
	100000000	14.748	16s	3.14159256
	1000000000	72.735	1m 14s	3.14159272

a. +/- 0.5 seconds error

2.7.4 Advantages of integrating IBM Platform Symphony V5.2 with InfoSphere BigInsights V1.4

This section compares the results from 2.7.2, “Results of the stand-alone benchmark for InfoSphere BigInsights V1.4” on page 37, and from 2.7.3, “Results of integration for IBM Platform Symphony V5.2 and InfoSphere BigInsights V1.4” on page 38. This section also highlights the advantages of performing this integration.

Three configurations were used for the sleep and the Pi estimated tests as follows:

- Default configuration

  This configuration uses the steps that are described in 2.6, “Additional configuration for IBM Platform Symphony” on page 34, without performing the changes in 2.6.2, “TCP/IP performance tuning” on page 35.

- Tuned configuration (with Ethernet tuning)

  This configuration uses the 2.6, “Additional configuration for IBM Platform Symphony” on page 34, and includes the changes in 2.6.2, “TCP/IP performance tuning” on page 35.

- Preloaded configuration (with Ethernet tuning and the **Prestart** option enabled)

  This configuration is the same as the tuned configuration, but with **Prestart** option enabled from IBM Platform Symphony.

Tuned and preloaded configurations: Optionally, you can add the following command-line option to reduce the debug level (Example 2-21 on page 39):

```
-Dmapred.map.child.log.level=WARN
```

The files are still created, but nothing is written in them.

Sleep benchmark results

For the sleep benchmark, Figure 2-4 compares the results that we obtained from the different test configurations.

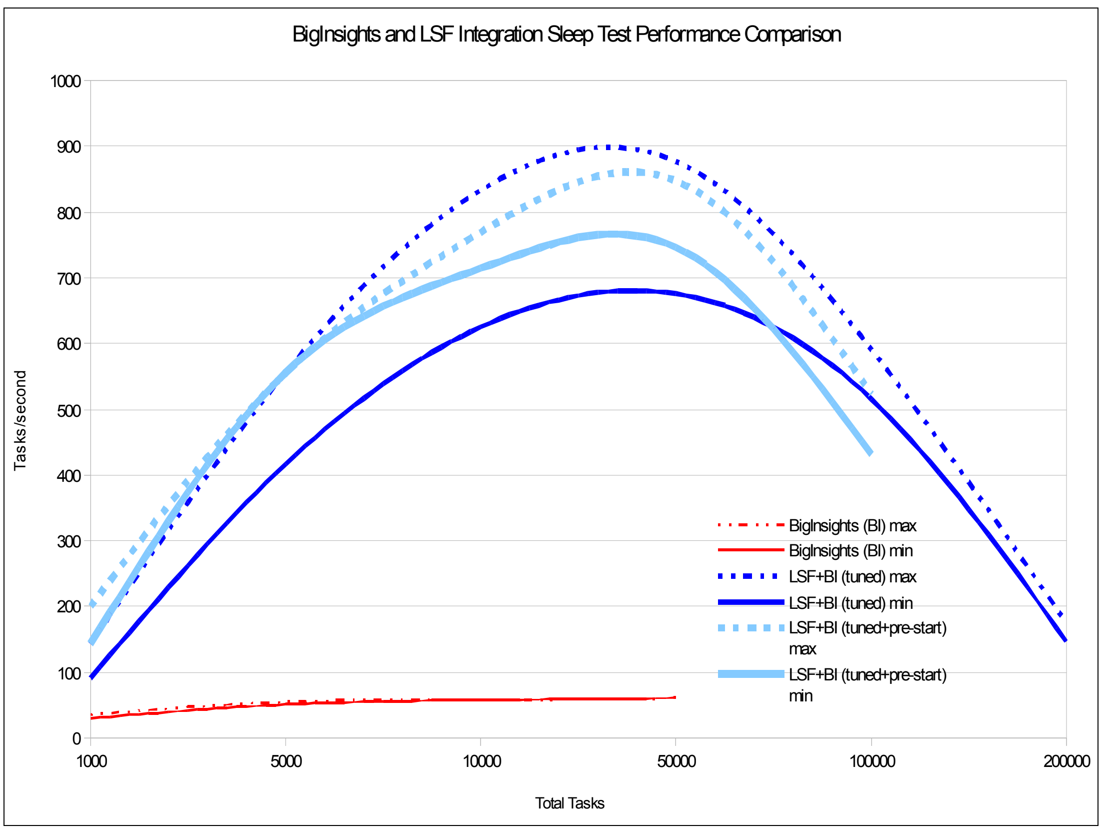


Figure 2-4 Sleep performance in tasks/second (default tuned, tuned, and preload)

The results show the following configurations:

- InfoSphere BigInsights only
- Integration of IBM Platform Computing LSF and InfoSphere BigInsights for tuned and preloaded configurations, with the last one accounting for the Prestart option that is enabled

Pi estimate benchmark results

For the Pi estimate benchmark, Figure 2-5 compares the results that we obtained through the two different test configurations.

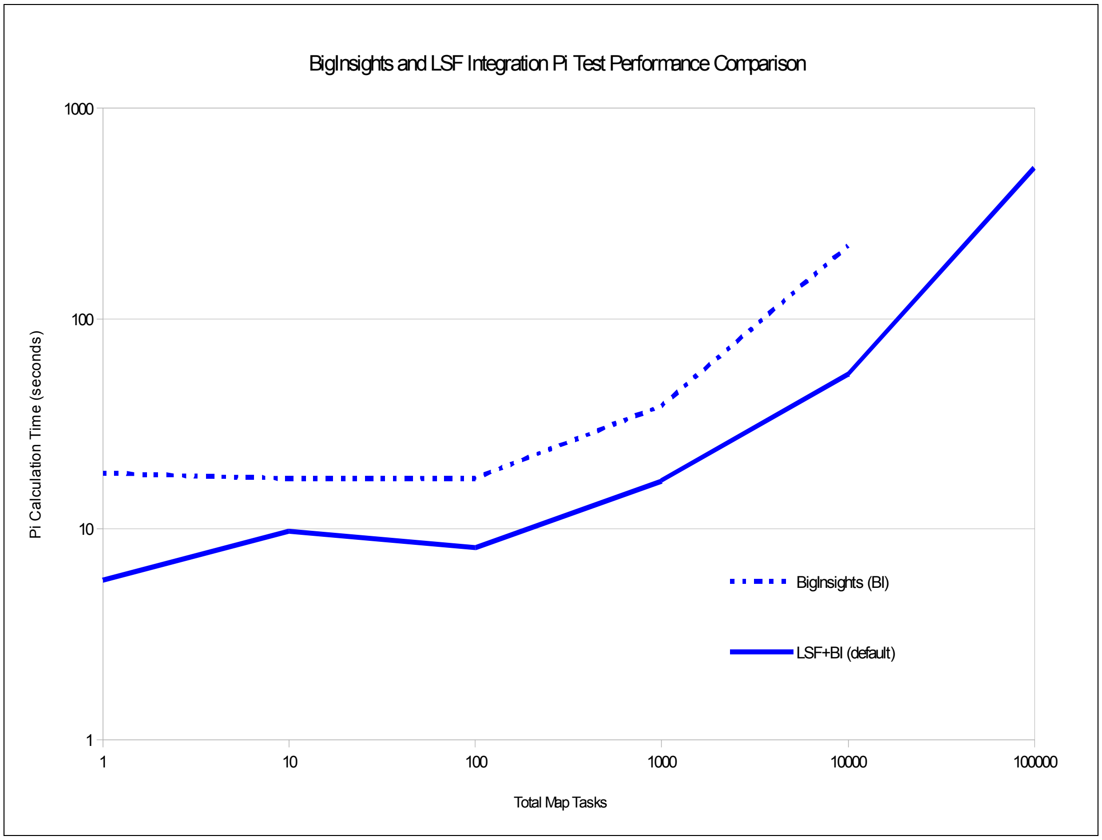


Figure 2-5 Pi estimate performance

The benchmark shows only the time that is needed to estimate the Pi value, discarding the time that is needed to write the maps into hard disk (that are equivalent between the two scenarios presented). The results are presented by using the following configurations:

- InfoSphere BigInsights only
- Integration of IBM Platform Computing LSF and InfoSphere BigInsights for a default configuration

2.8 Adding users

For the implementation steps described previously, the cluster has just one operational user who is the installation user egoadmin. This section describes how to add more users who can use the IBM Platform Symphony and the InfoSphere BigInsights integrated cluster.

2.8.1 Assumptions

Create a new supplementary user group to facilitate access to HDFS by the Hadoop users. This supplementary group is called *hdfsgrp*. This scenario uses the following assumptions:

- The user name to demonstrate the procedure is *testuser*.
- The hdfs data location is `/data/hdfs`.
- The **pdsh** command runs **ssh** to each data node and runs the command
- The **pdcp** command copies a file from the master to each of the data nodes.
- The home directories for new users are *not* shared across all the nodes. Adjust the directories as necessary if your home directory is shared.

2.8.2 Adding a user for the integrated cluster

To add a user for the integrated cluster:

1. Create a user on the master node and all other compute or data nodes (Example 2-22).

   Example 2-22 Creating a user on the master node and all compute or data nodes

   ```
   groupadd hdfsgrp
   pdsh groupadd hdfsgrp
   usermod –a –G hdfsgrp egoadmin
   pdsh usermod –a –G hdfsgrp egoadmin
   ```

2. Add the user to the master node and each node (Example 2-23). Run the commands as root from the master node.

   Example 2-23 Adding the user to the master node and all other nodes

   ```
   useradd testuser  -d /home/testuser
   id_on_master=`id –u testuser`
   pdsh adduser testuser –u $id_on_master  -d /home/testuser
   passwd testuser (set password for user on master)
   ```

3. Add the user to the hdfsgrp group (Example 2-24). Run the commands as root from the master node.

   Example 2-24 Adding the user to the hdfsgrp group

   ```
   usermod –a –G hdfsgrp testuser
   pdsh usermod –a –G hdfsgrp testuser
   ```

4. Create the hdfs user directory (Example 2-25). Run the commands as the egoadmin user from the master node.

   Example 2-25 Creating the hdfs user directory

   ```
   su – egoadmin
   hadoop dfs –mkdir /user/testuser
   hadoop dfs –chown testuser:hdfsgrp /user/testuser
   ```

5. Create a key for a new user for passwordless access to data nodes (Example 2-26).

   Example 2-26 Creating a key for passwordless access

   ```
   su – testuser
   ssh-keygen –t rsa –N “”
   ```

```
cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
exit
pdsh mkdir /home/testuser/.ssh
pdcp /home/testuser/.ssh/id_rsa /home/testuser/.ssh/id_rsa
pdcp /home/testuser/.ssh/id_rsa.pub /home/testuser/.ssh/authorized_keys
pdsh chmod 600 /home/testuser/,ssh/authorized_keys
```

6. Add environment variables to the user testuser (Example 2-27).

Example 2-27 Adding environment variables to testuser

```
/bin/sh -c 'echo ". /opt/ibm/sym52/profile.platform" >> /home/testuser/.bashrc
/bin/sh -c 'echo "source /opt/ibm/bi/conf/biginsights-env.sh" >>
/home/testuser/.bashrc
```

7. Run verifications steps to create data in hdfs (Example 2-28), which is a test of writing zeros to hdfs.

Example 2-28 Verification steps

```
[testuser@i05i04 ~]$ dd if=/dev/zero of=junk bs=1k count=10
10+0 records in
10+0 records out
10240 bytes (10 kB) copied, 9.05e-05 s, 113 MB/s
[testuser@i05i04 ~]$ hadoop dfs -put junk dfsjunk
[testuser@i05i04 ~]$ hadoop dfs -ls
Found 1 items
-rw-r--r-- 3 testuser hdfsgrp 10240 2012-11-06 13:55 /user/testuser/dfsjunk
```

8. Run a sleep job as testuser (Example 2-29).

Example 2-29 Running a sleep job as testuser

```
[testuser@i05i04 ~]$ time hadoop jar $HADOOP_HOME/hadoop-examples-1.0.0.jar
sleep 1 100
12/11/03 21:30:47 INFO internal.MRJobSubmitter: Connected to JobTracker(SSM)
12/11/03 21:30:47 INFO internal.MRJobSubmitter:  Job <Sleep job> submitted, job
id <37>
12/11/03 21:30:47 INFO internal.MRJobSubmitter: Job will verify intermediate
data integrity using checksum.
12/11/03 21:30:47 INFO mapred.JobClient: Running job: job__0037
12/11/03 21:30:48 INFO mapred.JobClient: map 0% reduce 0%
12/11/03 21:30:53 INFO mapred.JobClient: map 100% reduce 100%
12/11/03 21:30:53 INFO mapred.JobClient: Job complete: job__0037
12/11/03 21:30:54 INFO mapred.JobClient: Counters: 18
12/11/03 21:30:54 INFO mapred.JobClient:   Shuffle Errors
12/11/03 21:30:54 INFO mapred.JobClient:     WRONG_PATH=0
12/11/03 21:30:54 INFO mapred.JobClient:     CONNECTION=0
12/11/03 21:30:54 INFO mapred.JobClient:     IO_ERROR=0
12/11/03 21:30:54 INFO mapred.JobClient:   FileSystemCounters
12/11/03 21:30:54 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=35
12/11/03 21:30:54 INFO mapred.JobClient:   Map-Reduce Framework
12/11/03 21:30:54 INFO mapred.JobClient:     Reduce input groups=2
12/11/03 21:30:54 INFO mapred.JobClient:     Combine output records=0
12/11/03 21:30:54 INFO mapred.JobClient:     Map output records=1
12/11/03 21:30:54 INFO mapred.JobClient:     Shuffled Maps =1
```

```
12/11/03 21:30:54 INFO mapred.JobClient:     Reduce shuffle bytes=8
12/11/03 21:30:54 INFO mapred.JobClient:     Combine input records=0
12/11/03 21:30:54 INFO mapred.JobClient:     Spilled Records=2
12/11/03 21:30:54 INFO mapred.JobClient:     SPLIT_RAW_BYTES=0
12/11/03 21:30:54 INFO mapred.JobClient:     Map output bytes=4
12/11/03 21:30:54 INFO mapred.JobClient:     Reduce input records=1
12/11/03 21:30:54 INFO mapred.JobClient:     GC time elapsed (ms)=0
12/11/03 21:30:54 INFO mapred.JobClient:     Failed Shuffles=0
12/11/03 21:30:54 INFO mapred.JobClient:     Merged Map outputs=1
12/11/03 21:30:54 INFO mapred.JobClient:     Reduce output records=0

real    0m8.140s
user    0m3.681s
sys     0m0.441s
```

2.9 Adding nodes

After the integration, you add a node to InfoSphere BigInsights and IBM Platform Symphony.

Rejoining a machine to a cluster

Use these steps when a machine crashes and restarts, and you need to rejoin it to the cluster. You do not need to shut down or restart the entire cluster in this case.

1. Add the `/etc/hosts path` or DNS of the new node, depending on the name resolution method.
2. Add the same host name to the `conf/slaves` file on the master node.
3. Log in to the new subordinate node and run the following commands:

   ```
   $ cd path/to/hadoop
   $ bin/hadoop-daemon.sh start datanode
   $ bin/hadoop-daemon.sh start tasktracker
   ```

4. If you are using the function of the `dfs.include/mapred.include` file, add the node to the `dfs.include/mapred.include` file so that the NameNode and JobTracker can detect that the additional node was added. Enter the following commands:

   ```
   hadoop dfsadmin -refreshNodes
   hadoop mradmin -refreshNodes
   ```

5. Synchronize the Hadoop configuration files in the `$BIGINSIGHTS_HOME/hdm/hadoop-conf-staging/` directory on the management node with their counterparts on all of the nodes in the cluster. To synchronize these files, run the following command:

   ```
   $BIGINSIGHTS_HOME/bin/syncconf.sh hadoop force
   ```

Adding a data or compute node to the integrated software

To add a data or compute node to the integrated software for the InfoSphere BigInsights and the IBM Platform Symphony:

1. Add the new node host name to the DNS server or to the `/etc/hosts` path of all nodes.
2. Verify and complete as necessary all requirements that are described in 2.2.1, “Preconfiguration on all nodes” on page 21, according to the remaining nodes where the node will join.

3. Add the node manually to the current Hadoop configuration (the `$HADOOP_CONF_DIR/slaves` file) before you synchronize the configurations.
4. Verify that the InfoSphere BigInsights is started on all nodes (except the joining node) and that the HDFS as no errors. If possible, synchronize the cluster by running the following command as the egoadmin user:

   `$BIGINSIGHTS_HOME/bin/syncconf.sh hadoop force`
5. On the master node, run the following command as the egoadmin user:

   `addnode.sh hadoop <new_node_hostname>`

   > **Adding the force option:** The InfoSphere BigInsights software was copied to the new node, but none of the integration is present on the new node. For troubleshooting purposes, you might need to add the **`force`** option (at the end) to this command.

6. To install and complete the integration process on the new node, follow the steps only for compute nodes in the following sections:
   – 2.4, “Installing IBM Platform Symphony Advanced Edition” on page 29
   – 2.5, “Integrating IBM Platform Symphony and InfoSphere BigInsights” on page 31.
7. Restart the IBM Platform Symphony on all nodes for the new node to join correctly.
8. Before you apply that patch, stop the IBM BigInsigts and Hadoop.

2.10 Troubleshooting

This section describes troubleshooting hints and tips for working with IBM Platform Computing and InfoSphere BigInsights.

2.10.1 InfoSphere BigInsights

After you tune a sleep test with 10,000 maps and 10,000 prestarted services, remove all contents of the `/data/mapred/local` path on all nodes and then restart all nodes.

2.10.2 Increasing Java virtual memory

On all nodes, change the value of the variable *JVM_OPTIONS*, inside the `$PMR_HOME/conf/pmr-env.sh` file to `-Xms1024m -Xmx4096m` as shown in Example 2-30.

Example 2-30 Changes inside the $PMR_HOME/conf/pmr-env.sh file

```
[...]
# The JVM options
export JVM_OPTIONS="-Xms1024m -Xmx4096m"
[...]
```

2.10.3 Increasing system limits

Increase the limits for the root and egoadmin users by editing the `/etc/security/limits.conf` file and adding the lines as shown in Example 2-31.

Example 2-31 Maximizing the limit values for root and egoadmin user

```
egoadmin hard nofile 1000000
egoadmin soft nofile 1000000
egoadmin hard nproc unlimited
egoadmin soft nproc unlimited
root hard nofile 1000000
root soft nofile 1000000
root hard nproc unlimited
root soft nproc unlimited
```

Configured system limits: Even though some configured limits are already configured in InfoSphere BigInsights for the users, change these limits to bigger values. After the integration, both products use a much higher number of opened files and processes.

3

Implementation scenario for IBM Platform LSF

This chapter highlights the general challenges for high-performance computing (HPC) clusters. It includes some of the major features of IBM Platform LSF and the way these features can help to address the challenges of HPC clusters.

This chapter includes the following sections:

- Valid hardware resources to ensure that the setup can use IBM Platform LSF
- Sizing for I/O-intensive clusters
- Considerations for GPGPU intensive clusters
- Job queues
- Job scheduling
- Goal-oriented scheduling
- Job submission
- Lifecycle of jobs
- Compute units
- Application profiles
- Job submission prechecks and setup
- Job resizing
- Idle job detection
- Defining external resources (elims)
- Using advance reservations
- Hyper-Threading technology
- Changing the paradigm with guaranteed resources

3.1 Valid hardware resources to ensure that the setup can use IBM Platform LSF

The IBM Redbooks publication *IBM Platform Computing Solutions*, SG24-8073, presented the IBM Platform range of products. As explained in Chapter 3 of that book, these products were installed on an IBM iDataPlex® cluster. This environment in this chapter uses the same hosts, which are all iDataPlex M3 servers that are based on an Intel Xeon X5670 6-core processor. Each node has a dual socket for 12 cores per node. In IBM Platform LSF terms, each node has 12 slots. To change LSF to use threads rather than cores for slots, see 3.15, “Hyper-Threading technology” on page 92.

IBM Platform Computing Solutions, SG24-8073, also explained how to plan for IBM Platform LSF and how to install the product. This book highlights the features of IBM Computing LSF, such as queue configurations and scheduling policies, that you can use in your cluster to make the best usage of it.

3.2 Sizing for I/O-intensive clusters

Two ideologies of thought surround I/O-intensive clusters. You probably have some large central data storage device. This device might or might not be backed up to auxiliary storage, such as tape, but it is most likely protected by a RAID mechanism. This storage device is attached to one or more hosts that share the data. It might be shared by using a point-to-point protocol such as Network File System, (NFS). Alternatively, it might use a shared, parallel, distributed file system such as IBM General Parallel File System (GPFS).

Each compute node usually has a local disk, although you might have diskless compute nodes. In some industries, the administrators copy the files off the central storage to the local disk for processing before the job begins. Then, they copy the output back to the central storage after the job finishes. This copy can use NFS or `rsync`.

In some industries, the administrator chooses to work on the data, directly off the parallel distributed file system. This approach save time from copying the data from the central storage to the local nodes.

Some jobs might require a large amount of preprocessing and post-processing. The systems that have this requirement are larger (although fewer) than the compute nodes. They normally have more memory and might even have their own local RAID arrays or graphics processing units (GPUs). Getting data to these nodes, in a timely manner, before the main compute jobs runs is critical. You do not want to spend half the job time copying the data to the nodes, when no other job is running. Similarly, when copying data from these nodes, complete other productive tasks; do not wait for the I/O to complete.

Sizing the cluster for I/O-intensive workloads varies depending on the data movement method that you use. Usually it depends on the amount of data that is being processed. If you have a few or smaller input files, you can copy them. However, if you have many or large input files, it might not be possible to copy them over. Creating multiple jobs with dependencies can help with this problem. Alternatively, implementing IBM Platform Process Manager can also help with the workflow.

The speed of the copy depends on the interconnect that is being used (Ethernet or InfiniBand), the number of nodes that are requesting the file, and the number of nodes that are serving it. The speed of the interconnect is fixed. You cannot vary this speed. However, you can vary the number of nodes that are requesting the file (job size) and the number of nodes that are serving the files.

Most users quote how many GB per second they require and how many GB or TB of space they need. Most vendors can calculate the hardware that is required based on this value. With a parallel distributed file system, the more disks you have, the faster the parallel file system will go.

3.3 Considerations for GPGPU intensive clusters

A massive growth has occurred in the use of General Purpose Graphics Processing Units (GPGPU). These devices can provide massive amounts of vector operations that certain codes can take advantage of. Recently, several of the top ten most powerful super computers have taken advantage of GPGPUs.

GPGPUs are devices that are attached to a general-purpose host system. This general-purpose system provides access to the file system and memory. This system then uses external libraries and system calls to use the facilities that are provided by the GPU.

When you use hosts with GPUs, you must inform IBM Platform LSF of these resources. Typically a system has one or two GPUs attached to it, depending on the number of sockets on the host. Probably more x86 cores than GPUs are attached to the system. Normally you do not share GPUs between jobs. For best performance, treat them as an exclusive resource.

To use nVidia GPGPUs with IBM Platform LSF, install the specific *elims* to identify and monitor these devices. The elims must match the version of CUDA (a parallel computing platform and programming model) that is installed on the host systems. For more information about elims, see 3.13, “Defining external resources (elims)” on page 85.

Alternatively, you can define a GPU queue and place the hosts with GPUs in that queue. For more information about defining a queue, see 3.4, “Job queues” on page 51. If you have many nodes with GPUs and only have a few jobs that can take advantage of these GPUs, this approach might not be an effective use of resources. Normally several more x86 cores are on a single host than GPU cores that are on the host.

3.4 Job queues

A *queue* is a cluster-wide or network-wide holding place for jobs. Each queue might implement different job scheduling and control policies. A system administrator designs the number of queues, the resources that are available to that queue, and the policies on that queue. Users then submit jobs against these queues for execution on hosts within the cluster.

3.4.1 Defining queues

When you define a queue, ask the following questions:

- Which users will access and possibly manage this queue?
- Which hosts and resources are defined against this queue?
- Does this queue have an exclusive execution restriction?
- What type of jobs will we run on these queues? How long will they execute for?

All definitions for queues are stored in the `lsb.queues` file. You create a queue by modifying the `lsb.queue` file and by adding a few lines as shown in Example 3-1.

Example 3-1 Modifying the lsb.queue file

```
Begin Queue
QUEUE_NAME = shortjobs
DESCRIPTION = for short running jobs
ADMINISTRATORS = sdenham
PRIORITY = 75
USERS = aortiz asasaki yjw tmgr1 tmgr2
CPULIMIT = 2
RUNLIMIT = 5 10
CORELIMIT = 0
MEM = 800/100
SWP = 500/50
HOSTS = i05n46 i05n47
End Queue

Begin Queue
QUEUE_NAME = exclusive
PRIORITY = 70
EXCLUSIVE = Y
HOSTS = i05n47 i05n48 i05n49 i05n50 i05n51
DESCRIPTION = used for exclusive MPI jobs
End Queue
```

In the scenario in this chapter, we added two new queues. The first queue is for short-running jobs, is administered by user sdenham, and is accessible by a few select users. Setting up user sdenham as a queue administrator should reduce the load on the administrator. This user can perform operations (such as starting, stopping, or canceling jobs) on other user jobs within the queue. However, this user cannot edit any IBM Platform LSF configuration files nor can this user start or stop IBM Platform LSF.

The second queue, which is called *exclusive*, is for larger jobs, such as Message Passing Interface (MPI) jobs that prefer not to share resources with other jobs. For more information about exclusive scheduling, see 3.5.4, “Exclusive scheduling” on page 59. This second queue is open to anyone to use. Host i05n47 is in both queues. A host can be a member of many queues. The more queues that you have, the more complex the scheduling is to understand and manage. Using application profiles can help reduce the number of queues. For information about application profiles, see 3.9, “Application profiles” on page 73.

After you add these new lines, run the **badmin reconfig** command to communicate the new queues to IBM Platform LSF. To see the new queues, you can enter the **bqueues** command. For a more detailed view of the queues and to verify their expected behavior, use the **-l** flag (Example 3-2).

Example 3-2 Verifying the queues

```
$ bqueues -l shortjobs

QUEUE: shortjobs
  -- for short running jobs

PARAMETERS/STATISTICS
PRIO NICE STATUS          MAX JL/U JL/P JL/H NJOBS  PEND   RUN SSUSP USUSP  RSV
 75    0  Open:Active       -    -    -    -     1     1     0     0     0    0
```

```
Interval for a host to accept two jobs is 0 seconds

DEFAULT LIMITS:
 RUNLIMIT
 5.0 min of i05n45.pbm.ihost.com

MAXIMUM LIMITS:
 CPULIMIT
 2.0 min of i05n45.pbm.ihost.com

 RUNLIMIT
 10.0 min of i05n45.pbm.ihost.com

 CORELIMIT
      0 M

SCHEDULING PARAMETERS
           r15s   r1m  r15m   ut      pg    io   ls    it    tmp    swp    mem
 loadSched   -     -     -     -       -     -    -     -     -     500M   800M
 loadStop    -     -     -     -       -     -    -     -     -      50M   100M

USERS: asasaki aortiz tmgr1 tmgr2 yjw
HOSTS:  i05n46 i05n47
ADMINISTRATORS:  sdenham
```

In Example 3-2 on page 52, a run limit is on the `shortjobs` queue. Any job that does not complete within the run limit is first warned and then ultimately stopped if it exceeds this limit. This queue is for short running jobs. You do not want any jobs to use all the slots.

3.4.2 Default queue

The default queue is defined in the `lsb.params` file. By default, the queue is the *normal queue* and the *interactive queue*. When a user submits a job, IBM Platform LSF chooses the best queue to send the job to. If the user does not specify a queue, the best queue is chosen based on the input parameters of the job. For example, if the job is an interactive job, the interactive queue is chosen automatically (Example 3-3).

Example 3-3 Automatic default queue selection

```
$ bsub -I ./Ijob
Job <4719> is submitted to default queue <interactive>.
<<Waiting for dispatch ...>>
<<Starting on i05n45.pbm.ihost.com>>
hi
i05n45
Type something in then:
exit

$ bsub  ./Ijob
Job <4720> is submitted to default queue <normal>.
```

Using the **`bsub -I`** command indicates that the job is an interactive job (the jobs takes input from `stdin`) and this job is placed on the interactive queue. However, if you resubmit the job without the **`-I`** flag, the job is sent to the normal queue.

To change the order or to choose different queues, update the DEFAULT_QUEUE setting in the `lsb.params` file and restart the master batch daemon (MBD) by using the **`breconfig`** command.

3.4.3 Managing queues

Queues can be in one of several states. A queue can be *open* and accepting jobs, or a queue can be *closed*, meaning that it rejects any job that is submitted. After jobs are on the queue, they are dispatched depending upon the status of the queue. If the queue is *active*, jobs are dispatched. If the queue is *inactive*, jobs are held on the queue. Table 3-1 shows how to modify the status of a queue.

Table 3-1 Modifying the status of a queue

Status	Command	Meaning
open	**`badmin qopen`** ***queuename***	Accept jobs
close	**`badmin qclose`** ***queuename***	Reject jobs
active	**`badmin qact`** ***queuename***	Dispatch jobs
inactive	**`badmin qinact`** ***queuename***	Hold jobs

To drain a queue, you change the queue status to *close*, but leave the queue active so that jobs can continue to be dispatched. Under normal circumstances, queues are open and active. However, you can specify periods in which queues are active or inactive, such as a queue that is run nightly. Users can submit jobs to this queue during the day, but the queue becomes active only at night when it dispatches jobs.

To remove a queue, comment the Begin Queue and End Queue lines (Example 3-4). You do not need to comment the queue configuration. Then, run the **`badmin reconfig`** command again.

Example 3-4 Removing a queue

```
# Begin Queue
QUEUE_NAME = interactive
DESCRIPTION = default for interactive jobs
PRIORITY = 80
INTERACTIVE = ONLY
NEW_JOB_SCHED_DELAY = 0
HOSTS = i05n45
# End Queue
```

3.5 Job scheduling

After you define the queues, consider the type of scheduling policy you want on these queues. IBM Platform LSF supports several scheduling policies. The following scheduling policies can coexist within the same cluster:

- First come, first served (FCFS)
- Preemption
- Fairshare
- Exclusive
- Backfill

The following sections describe the policy types, their advantages, and how to configure them.

3.5.1 First come, first served

First come, first served is the default scheduling policy. IBM Platform LSF attempts to dispatch jobs onto the run queue in the order they are received. FCFS is simple to understand and use. However, using this policy can cause long-running, low-priority jobs to force higher-priority, short-running jobs to go into a pending state for an unacceptable time.

Example 3-1 on page 52 shows the first queue definition, `shortjobs`, which is an FCFS queue.

3.5.2 Preemption scheduling

Preemption scheduling is similar to FCFS. Jobs are submitted and run in the order they are received. However, unlike FCFS, higher priority jobs can force other longer running jobs to be suspended while the higher priority jobs get a chance to run. After the higher priority jobs are finished, the lower priority jobs are will be resumed. Preemption scheduling includes the following types:

- Job slot preemption
- Advanced reservation (see 3.14, “Using advance reservations” on page 87)
- License preemption (by using License Scheduler)

This type of scheduling solves the problem of longer running, lower priority jobs that monopolize run queues. Example 3-5 shows the configuration of preemption scheduling on two queues.

Example 3-5 Preemption configuration and scheduling

```
Begin Queue
QUEUE_NAME = short
PRIORITY = 70
HOSTS = i05n45 i05n46 i05n47 # potential conflict
PREEMPTION = PREEMPTIVE[normal]
End Queue

Begin Queue
QUEUE_NAME = normal
PRIORITY = 40
HOSTS = i05n45 i05n46 i05n47 # potential conflict
PREEMPTION = PREEMPTABLE[short]
End Queue
```

In Example 3-5, jobs that run on the normal queue can be preempted by jobs that run on the short queue, if a resource constraint is on the three nodes in these queues.

> **Suspended jobs:** Jobs that are SUSPENDED are not stopped nor checkpointed (stopped, checked, and restarted). A job does not release its resources when it is suspended.

One of the potential issues with preemption scheduling is that large jobs, which use the entire memory of many nodes, can be difficult to suspend and resume. If a process uses all the memory of a machine, the process or job is then suspended. It does *not* release its resources. The operating system must free some memory by paging some of the memory pages of the larger, suspended job. When some memory becomes available, the higher priority jobs can run. After they run, the operating system has to page the memory pages back to resume the original job (possibly on *all* the effected nodes). This process can take a large amount of time. Before you use preemption, you must understand the characteristics of the jobs that are running on the cluster and whether they can be preempted.

Jobs that are selected for preemption are selected from the least-loaded hosts first. However, it is possible to change this characteristic. You can choose to preempt jobs that run for the least amount of time or to avoid preempting jobs that nearly finished their execution. You can also limit the number of times a job can be preempted. Otherwise, it might never finish on a busy system.

By adding the values that are shown in Example 3-6 to the `lsb.params` and `lsb.applications` files, you can influence the preemption.

Example 3-6 Limiting preemption behavior

```
NO_PREEMPT_FINISH_TIME = 10
MAX_JOB_PREEMPT = 2
NO_PREEMPT_RUN_TIME=60
```

These settings prevent jobs that have been running for over 60 minutes or jobs that have just 10 minutes left to run. They preempt only jobs that have a maximum of two times.

3.5.3 Fairshare scheduling

Fairshare scheduling divides access to the cluster resources by using *shares*. It prevents one user from monopolizing all the cluster resources by taking the following factors into consideration:

- Users share assignment
- Job slots that are reserved and in use
- Run time of running jobs
- Cumulative actual processor time
- Historical run time of finished jobs

When a user submits a job, the number of shares or priority decreases. The submitted job gets this priority in the queue. If a user submits many jobs, these jobs have a lower priority compared to another user who submits fewer jobs, providing all jobs use a similar amount of resources. However, after the job finishes, the priority increases.

The following types of fairshare scheduling can be implemented:

- Queue (equal share or user hierarchical)
- Cross queue
- Host partition

Queue equal share

Each user is given an equal number of shares. Example 3-7 shows a round-robin queue with equal shares given to all users.

Example 3-7 Round-robin queue

```
Begin Queue
QUEUE_NAME = roundrobin
PRIORITY = 40
FAIRSHARE = USER_SHARES[[default,1]]
USERS = parkera sdenham yjw akumar
End Queue
```

Figure 3-1 illustrates the definition of the round-robin queue. In this queue, the four users have an equal share. Theoretically, each user gets 25% of the resources that are available.

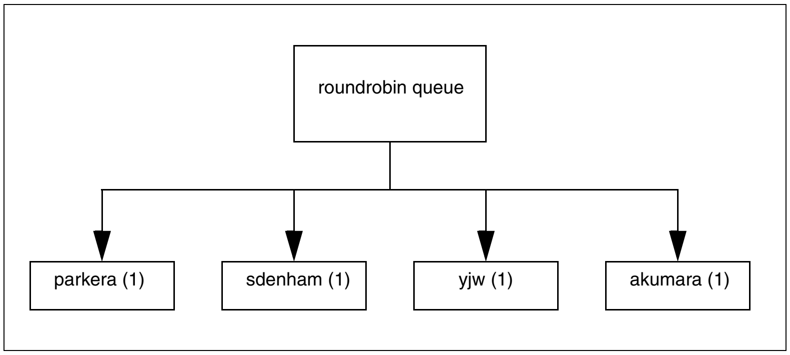


Figure 3-1 Resources available in the round-robin queue

Queue user hierarchical

You might have a situation where you do not want to allocate resources equally between users. You might want to give certain users or projects more shares than others. These groups might pay for more resources within the shared cluster, or their projects might have a higher priority than others. Therefore, you want to ensure that they can use more of the resources within the cluster but not monopolize the cluster.

You can then break down the shares within the cluster into groups and then into users within each group. Figure 3-2 shows this break down between two different groups STG and ITS. These two groups then have several users, each with their own numbers of shares.

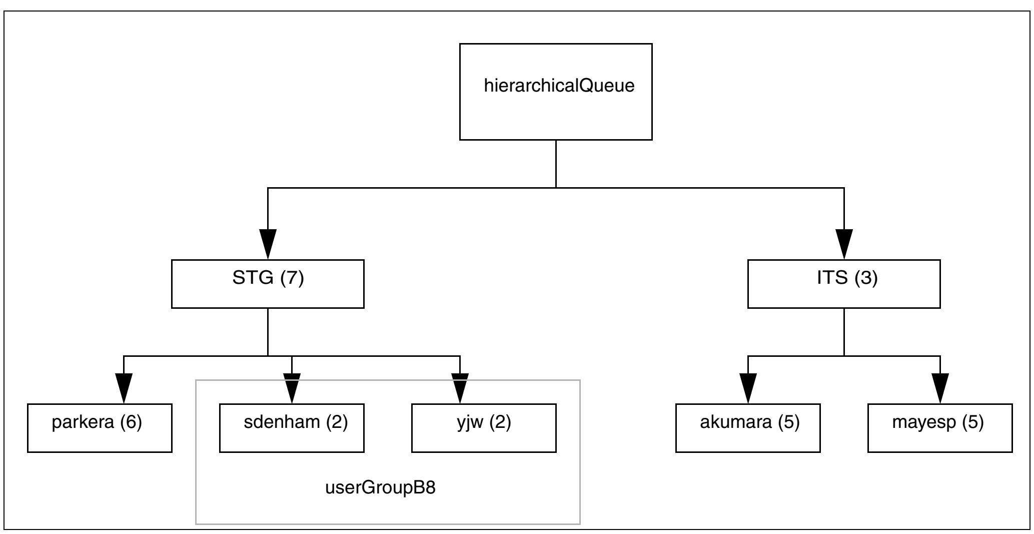


Figure 3-2 Hierarchical queues

The best way to calculate these shares is out of 10. In this example, the shares of STG (5) plus the shares of ITS (5) is 10. STG gets 70% of the resources, and ITS gets only 30% of the resources. The same breakdown applies to the users. Of the 70% of resources that STG receives, user parkera gets 60% of the resources, and users sdenham and yjw each get only 20% of the resources.

Example 3-8 shows how to define these shares in the `lsb.users` file.

Example 3-8 Defining shares

```
Begin UserGroup
GROUP_NAMEGROUP_MEMBERUSER_SHARES
userGroupB8(sdenham yjw)()
STG   (parkera userGroupB8@)([parkera,6] [userGroupB8@,2])
ITS   (akumara mayesp)([default,5])
End UserGroup
```

Example 3-9 shows how to configure the `lsb.queues` file. In Example 3-9, FAIRSHARE is set up against the queue. You can use different shares for different groups on different queues. You can have multiple hierarchy queues.

Example 3-9 Configuring the queue

```
Begin Queue
QUEUE_NAME = hierarchicalQueue
PRIORITY = 40
USERS = STG ITS
FAIRSHARE = USER_SHARES[[STG,7] [ITS,3]]
End Queue
```

To check the share allocation, enter the **`bugroup -l`** command as shown in Example 3-10.

Example 3-10 Checking the share allocation

```
$ bugroup -l
GROUP_NAME:  userGroupB8
USERS:       sdenham yjw

GROUP_NAME:  STG
USERS:       parkera userGroupB8@
SHARES:      [parkera, 6] [userGroupB8@, 2]

GROUP_NAME:  ITS
USERS:       mayesp akumara
SHARES:      [mayesp, 5] [default, 5] [akumara, 5]
```

You can use the **`bqueues -l`** command to discover the share allocation on the queue itself.

Cross queue

By using cross-queue fair sharing, you can apply the policy of prioritizing user or groups of users access to resources within a single queue across multiple queues. This way, if a user uses different queues, the priority is taken into account across all the queues. When you define cross-queue fairsharing, you create a master queue. Within this master queue, you define subordinate queues that inherit their policy from the master queue. You can create multiple subordinate queues for each master. However, a subordinate queue cannot be a member of multiple master queues.

Example 3-11 shows the definition of a master queue, called *master*, and two subordinate queues.

Example 3-11 Showing the master queue definition

```
Begin Queue
QUEUE_NAME = master
DESCRIPTION = master queue definition cross-queue
PRIORITY = 50
FAIRSHARE = USER_SHARES[[STG@,6] [ITS@,4]]
FAIRSHARE_QUEUES = normal-q short-q
HOSTS = hostGroupA # resource contention
End Queue

Begin Queue
QUEUE_NAME = short-q
DESCRIPTION = short jobs
PRIORITY = 70 # highest
HOSTS = hostGroupA
RUNLIMIT = 5 10
End Queue

Begin Queue
QUEUE_NAME = normal-q
DESCRIPTION = default queue
PRIORITY = 40 # lowest
HOSTS = hostGroupA
End Queue
```

Host partition

All previous fairshare policies examined users and queues by using host partitions, although you can handle resource contention on the hosts. You can use a host partition, for example, if some hosts had special hardware and you wanted to prevent users from monopolizing this hardware.

To enable host partition fairsharing, you update the `lsb.hosts` file as shown in Example 3-12.

Example 3-12 Updating the lsb.hosts file

```
Begin HostPartition
HPART_NAME = GPUSystems
HOSTS = i05n48 i05n49
FAIRSHARE = USER_SHARES[[ITS@,3] [STG@,7]]
End HostPartition
```

3.5.4 Exclusive scheduling

As the name implies, exclusive scheduling gives a job exclusive use of the server host. This policy can be useful when a job is sensitive to other jobs that are running on the same host. To use a host exclusively, you must submit the job by using the **bsub -x** command. By default, exclusive jobs cannot be preempted, although you can override this setting. The second queue in Example 3-1 on page 52 shows the definition of an exclusive queue. Typically, exclusive queues are used for parallel MPI jobs, in combination with backfill scheduling.

3.5.5 Backfill scheduling

Backfill scheduling works by looking at the run limit that is specified on a job to calculate when a job will finish and work backwards from there. Backfill scheduling is best used with large parallel jobs. By using traditional FCFS scheduling, small sequential jobs always win because these jobs can start as soon as a slot becomes available. Parallel jobs must wait for enough resources or slots to become available before they start. By using backfill scheduling, smaller jobs to slots in between larger jobs can use up smaller spare slots.

Figure 3-3 shows six hosts (hostA - host F). A job (A) is using three of these hosts, and another job (B) is waiting in the queue to use all six hosts. It must wait until job A finishes before it is dispatched. However, job C needs only two hosts and will run for 3 minutes. Therefore, the scheduler can slot job C onto the free hosts that job A is not using, without affecting the start time of job B. If the user did not specify this 3-minute run time, the job is not dispatched. If the job uses more than 3 minutes, it is stopped.

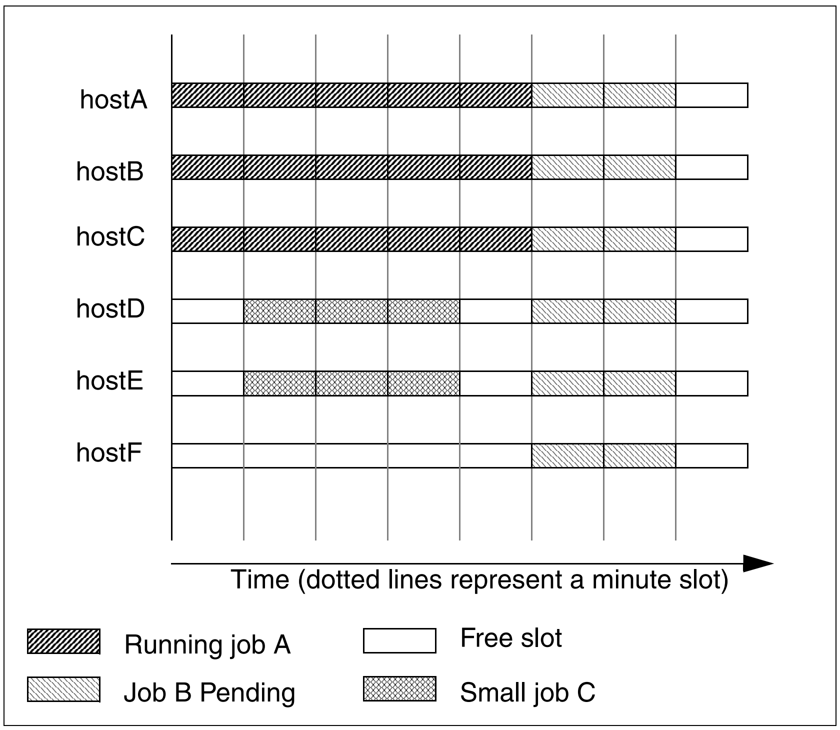


Figure 3-3 Showing job slots on hosts

By using backfill scheduling, the smaller jobs can be slotted in, but a processor limit *must* be specified to take advantage of back fill scheduling. This way, the dispatcher can calculate where to place the jobs. If the user does not specify a limit, no limit is assumed. In this example, job C would not be dispatched before job B.

Example 3-13 shows a queue that is defined with backfill scheduling.

Example 3-13 Defining a queue with backfill scheduling

```
Begin Queue
QUEUE_NAME = backFill
DESCRIPTION = used for parallel jobs
PRIORITY = 45
RES_REQ = select[ncpus==8 && ut<0.15]
BACKFILL = Y
```

```
RUNLIMIT = 20 300
SLOT_RESERVE = MAX_RESERVE_TIME[300]
End Queue
```

Tip: Jobs in a backfill queue can also be preempted. However, a high priority queue cannot preempt a backfill queue if a higher priority queue reserves resources.

Consider using backfill scheduling when you use exclusive queues because IBM Platform LSF can more efficiently use the available resources.

3.6 Goal-oriented scheduling

By using goal-based scheduling, you can ensure that users or jobs can gain access to key resources such as slots, hosts, or software licenses.

You might want to schedule a workload for many business reasons such as an impending deadline or a contractual obligation, such as a service-level agreement (SLA).

Goal-oriented scheduling is configured in the `lsb.serviceclasses` file. When you define a service class, you must decide the service-level goals:

Deadline Indicates when jobs within the service class should finish by
Throughput The number of jobs that should finish per hour
Velocity The number of concurrent jobs that should run

Each goal might also have an optional time window when the goal is active. For example, you can define periods for hour runs or weekend runs. You can have multiple time periods for a service class, but they must not overlap.

3.6.1 Defining a service class

Example 3-14 shows how to define two new service classes that are called `weekend` and `engineering`. In this example, we want to ensure that jobs submitted against the weekend service class finish by Monday morning. Only the engineering department is guaranteed to run 10 jobs per hour.

Example 3-14 Defining a service class

```
Begin ServiceClass
NAME = Weekend
PRIORITY = 20
GOALS = [DEADLINE timeWindow (5:18:00-1:8:30)]
DESCRIPTION = "weekend regression testing"
End ServiceClass

Begin ServiceClass
NAME = Engineering
PRIORITY = 20
USER_GROUP = eng
GOALS = [THROUGHPUT 10 timeWindow ()]
DESCRIPTION = "constant throughput for CEs"
End ServiceClass
```

After you add the lines to `lsb.serviceclasses` file and restart the Load Information Manager (LIM) and MBD, enter the **`bsla`** command to verify that they are active (Example 3-15).

Example 3-15 Viewing a service class

```
$ bsla
SERVICE CLASS NAME:  Weekend
 -- "weekend regression testing"
PRIORITY:  20

GOAL:  DEADLINE
ACTIVE WINDOW: (5:18:00-1:8:30)
STATUS:  Inactive
SLA THROUGHPUT:  0.00 JOBS/CLEAN_PERIOD

   NJOBS   PEND    RUN     SSUSP   USUSP   FINISH
      0      0       0        0       0       0
----------------------------------------------------------------------
SERVICE CLASS NAME:  Engineering
 -- "constant throughput for CEs"
PRIORITY:  20
USER GROUP: eng

GOAL:  THROUGHPUT 10
ACTIVE WINDOW: Always Open
STATUS:  Active:On time
SLA THROUGHPUT:  0.00 JOBS/CLEAN_PERIOD
OPTIMUM NUMBER OF RUNNING JOBS:  0

   NJOBS   PEND    RUN     SSUSP   USUSP   FINISH
      0      0       0        0       0       0
```

Notice that a time window was not provided for the engineering department, which means that the service level is always active. Only the *eng* group can use this window, unlike the weekend service class, which is available to everyone.

3.6.2 Using a service class

To use a service class, the user submits their job against the class as shown in Example 3-16.

Example 3-16 Submitting a job against a service class

```
$ bsub -sla Weekend myjob
```

The **`sla`** parameter names the service class to be used, which is *weekend* in this example. Specify a run limit with the job, which you can specify on the queue against the application profile or on the command line (using the **`-W`** parameter). When you specify a job limit, the scheduler can properly calculate the service class. If the scheduler cannot detect how long a job will run, how can it accurately calculate the deadline of the job or how many jobs to schedule per hour?

3.7 Job submission

After you define the queues and the scheduling policies of these queues, you can start using them. To submit a job to the job queue, you enter the **bsub** command. You can run the **bsub** command by using a script, a command line, with a spool file or run interactively. You can also use the web interface through IBM Platform Application Center if you have installed it.

The **bsub** command has many options. Table 3-2 lists the common flags that are used throughout this book.

Table 3-2 Common flags that are used with the bsub command

Flag	Function
-n #	Number of cores that are required for a job
-q *qname*	Submit a job to a specified queue
-R *string*	Resource selection criteria
-U *reservation*	Submit a job against a defined advance reservation
-P *project*	Submit a job against a named project
-app *name*	Use a defined application profile
-x	Have exclusive use of a server host
-W	Specify a wall clock limit for a job

You can learn about the **bsub** command options in *Running Jobs with IBM Platform LSF*, SC27-5307, at the following address or in the online manual by typing the **man bsub** command:

http://publibfp.dhe.ibm.com/epubs/pdf/c2753070.pdf

The IBM Platform LSF daemons that run on server hosts report resource usage periodically to the master. The master host gathers this information and calculates where to dispatch jobs based on the information that it receives from the hosts and the user requirements.

A resource can be classified as one of the following values:

- `Numeric`, for example, amount of memory, number of cores
- `String`, for example, host type or host status
- `Boolean`, for example, presence of a particular feature such as a GPU

These resources can be built in and discovered by IBM Platform LSF, or they can be defined externally by the site administrators (see 3.13, “Defining external resources (elims)” on page 85).

Table 3-3 shows static host information that is discovered by IBM Platform LSF when it starts.

Table 3-3 Static host information

Resource	Measures	Unit
type	Host type	String
model	Host model	String
hname	Host name	String
cpuf	CPU factor (relative)	Numeric

Resource	Measures	Unit
server	Host can run jobs	Boolean
rexpri	Remote execution priority (nice)	Numeric
ncpus	Number of cores	Numeric
ndisks	Number of local disks	Numeric
maxmem	Maximum RAM available	Numeric (MB)
maxswp	Maximum swap available	Numeric (MB)
maxtmp	Maximum available in /tmp	Numeric (MB)

When you select a resource, the requirement string is divided into the sections that are shown in Table 3-4.

Table 3-4 String usages

Requirement	Usage
Selection	select[selection_string]
Job usage	rusage[rusage_string]
Ordering	order[order_string]
Locality	span[span_string]
Same	same[same_string]

The span and same sections as shown in Table 3-4 are specifically for parallel jobs. For example, you can use the following command to select hosts with more that 1 GB of available memory. The following commands are shown with different string parameters:

```
$ bsub -R "select[type==any && mem>1024]" myJob
```

The `select` string was dropped if it is first in the resource requirement. The following command selects nodes with six cores and a utilization of less than 0.5 or nodes with 12 cores and a utilization of less than 0.75:

```
$ bsub -R "(ut<0.50 && ncpus==6) || (ut<0.75 && ncpus==12)" myJob2
```

The following command selects machines with more than six cores and 4 GB of memory, ordering them by utilization (lowest first) and reserving 300 MB of swap and 1 GB of memory for the duration of the job:

```
$ bsub -R "select[ncpus>6 && mem>4096] order[ut] rusage[swap=300,mem=1024]" myJob3
```

The next command runs a 32-way job on three hosts where their utilization is less than 0.20 and prioritizes them based on the lowest utilization:

```
$ bsub -n 32 -R "select[ut<0.20] order [ut] span[hosts=3]" myJob4
```

The next command runs a 64-way job on the parallel queue in exclusive host mode on the same type of hosts that use 12 slots per host. The job runs for 12 hours.

```
$ bsub -n 64 -x -W 12:0 -q parallel -R "select[type=any] span[ptile=12]" myJob5
```

You can also use multiple **-R** options as follows:

```
$ bsub -R "select[tmp > 100]" -R "select[ut<0.10]" -R "order[ut]" myjob5
```

These examples can be useful if you implement script job submission and want to build them by using the command line. IBM Platform LSF merges these individual selects into one and selects hosts that meet the criteria.

3.7.1 Job arrays

A *job array* is an IBM Platform LSF structure that allows a sequence of jobs to be submitted together. These jobs share executable file, but have different input and output files.

The advantage of using job arrays is that the job array can be controlled as a single unit. However, IBM Platform LSF schedules each individual element of the array separately. Job arrays can be used, for example, for bulk file conversion, image rendering, or regression testing.

Jobs arrays are submitted by using the **bsub** command as shown in Example 3-17.

Example 3-17 Submitting job arrays

```
$ bsub -J "aptest[1-8]" -i “"inputfile.%I" -o "outputfile.%J.%I" APsapp
```

In Example 3-17, a job with eight elements, called `aptest`, is submitted. The value `%I` is replaced by IBM Platform LSF to be the array element number (1 - 8 in this example). The value `%J` is the job number.

Example 3-18 shows a small script, called **APsapp**, that we created.

Example 3-18 Sample script for job arrays

```
#!/bin/bash
echo
echo Called on `hostname` at `date`
echo Input parameters are $*
echo STDIN is:
cat -
sleep 5
exit 0
```

In the same directory, we created eight files in the range `inputfile.1` - `inputfile.8`. These files are used as `stdin` for the script in Example 3-18. After you submit the job, you can check the output by running the **bjobs** command (Example 3-19).

Example 3-19 bjobs output for job arrays

```
$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
3462    lsfadmi RUN   normal     i05n45.pbm. i05n47.pbm. aptest[1]  Oct 29 15:06
3462    lsfadmi RUN   normal     i05n45.pbm. i05n47.pbm. aptest[2]  Oct 29 15:06
3462    lsfadmi RUN   normal     i05n45.pbm. i05n47.pbm. aptest[3]  Oct 29 15:06
3462    lsfadmi RUN   normal     i05n45.pbm. i05n47.pbm. aptest[4]  Oct 29 15:06
3462    lsfadmi RUN   normal     i05n45.pbm. i05n47.pbm. aptest[5]  Oct 29 15:06
3462    lsfadmi RUN   normal     i05n45.pbm. i05n47.pbm. aptest[6]  Oct 29 15:06
3462    lsfadmi RUN   normal     i05n45.pbm. i05n47.pbm. aptest[7]  Oct 29 15:06
3462    lsfadmi RUN   normal     i05n45.pbm. i05n47.pbm. aptest[8]  Oct 29 15:06
```

Example 3-19 shows each job that is running. They all have the same job number. If the user (or administrator) wants to cancel the entire job, it is easy to do. You can also reference individual tasks within the job by using `JOBID[array number]`, for example, **`bjobs "3462[2]"`** or **`bkill "3462[8]"`**.

After the job finishes, we had eight new files in the range `outputfile.3461.1` - `outputfile3461.8`. Example 3-20 shows the output for each file.

Example 3-20 stdout from job element

```
Sender: LSF System <lsfadmin@i05n47.pbm.ihost.com>
Subject: Job 3461[1]: <aptest[1-8]> Done

Job <aptest[1-8]> was submitted from host <i05n45.pbm.ihost.com> by user
<lsfadmin> in cluster <cluster1>.
Job was executed on host(s) <i05n47.pbm.ihost.com>, in queue <normal>, as user
<lsfadmin> in cluster <cluster1>.
</home/lsfadmin> was used as the home directory.
</gpfs/fs1/home/home/lsfadmin> was used as the working directory.
Started at Mon Oct 29 15:05:31 2012
Results reported at Mon Oct 29 15:05:37 2012

Your job looked like:

------------------------------------------------------------
# LSBATCH: User input
/gpfs/fs1/home/home/lsfadmin/APsapp
------------------------------------------------------------

Successfully completed.

Resource usage summary:

    CPU time   :      0.03 sec.
    Max Memory :         1 MB
    Max Swap   :        36 MB

    Max Processes  :         1
    Max Threads    :         1

The output (if any) follows:


Called on i05n47 at Mon Oct 29 15:05:31 EDT 2012
Input parameters are
STDIN is:
1
```

Example 3-20 shows the output from `stdout`. Obviously our script can create eight new files instead of sending its output to `stdout`. However, without passing any parameters, we would have to identify the element within the array from the environment variable `LSB_BATCH_JID`. This value is different for each element of the array, for example, `3462[1]` to `3462[8]`.

By default, IBM Platform LSF allows a maximum of 1000 elements in a job array. You can increase this value to a maximum of 65534 by changing the `MAX_JOB_ARRAY_SIZE` parameters in the `lsb.params` file.

3.7.2 Lifecycle of jobs

A job goes through several states from the time it is started to the time it ends. Table 3-5 lists the states of a job.

Table 3-5 Lifecycle for different jobs

Job state	Description
PEND	Waiting in a job queue to be scheduled and dispatched
RUN	Dispatched to hosts and running
DONE	Finished normally with a zero exit value
PSUSP	Suspended by its owner or the IBM Platform LSF administrator while in the PEND state
USUSP	Suspended by its owner or the IBM Platform LSF administrator after it is dispatched
SSUSP	Suspended by the LSF system after it is dispatched
EXIT	Abnormally terminated job

When you submit a job, you can see how it changes through each of these states. Example 3-21 shows a job that is submitted to IBM Platform LSF.

Example 3-21 Sample job

```
$ cat job.sh
echo $SHELL
export sleep_time=1000
if [ $# -eq 1000 ] ; then
sleep_time=$1
fi
date
sleep ${sleep_time}
date
```

Example 3-21 shows the sample job that is submitted to IBM Platform LSF. Example 3-22 shows its initial state.

Example 3-22 Initial state of the job

```
$ bsub -o job.out -e job.err < job.sh
Job <2219> is submitted to default queue <normal>.

$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
2219    yjw     RUN   normal     i05n45.pbm. i05n47.pbm. *ime};date Oct 15 16:56
```

The job was submitted and is now running. After you suspend the job, you can check its state as shown in Example 3-23.

Example 3-23 Stopping the job and checking its state

```
$ bstop 2219
Job <2219> is being stopped

$ bjobs
```

```
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
2219    yjw     USUSP normal     i05n45.pbm. i05n47.pbm. *ime};date Oct 15 16:56
```

The job is now suspended. You can resume the job and check its state as shown in Example 3-24.

Example 3-24 Resuming a stopped job and checking its state

```
$ bresume 2219
Job <2219> is being resumed

$ bjobs 2219
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
2219    yjw     RUN   normal     i05n45.pbm. i05n47.pbm. *ime};date Oct 15 16:56
```

The job continues to run. You can then terminate the job as shown in Example 3-25.

Example 3-25 Terminating the job

```
$ bkill 2219
Job <2219> is being terminated

$ bjobs  2219
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
2219    yjw     EXIT  normal     i05n45.pbm. i05n47.pbm. *ime};date Oct 15 16:56
```

Figure 3-4 shows the different states that a job can go through, and the various commands that you can use to change the state.

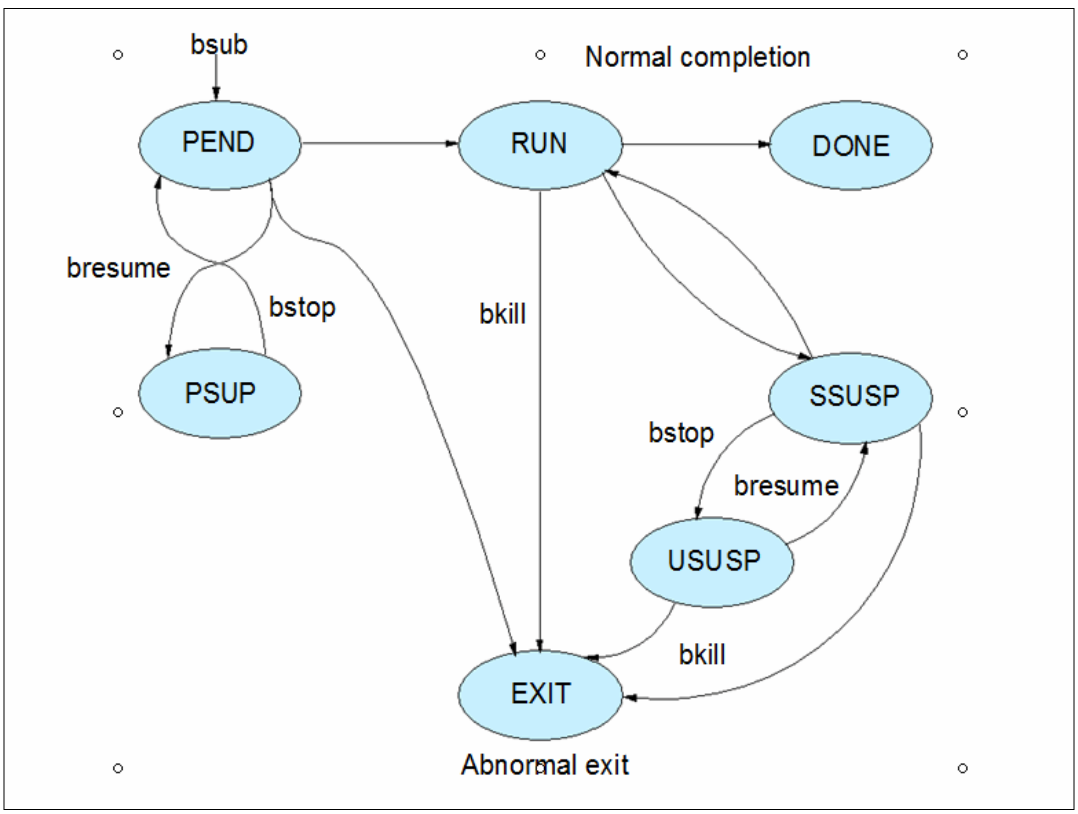


Figure 3-4 Different states of a job and the commands to manage them

3.8 Compute units

Most clusters are composed of enclosure-based systems that are divided into multiple node groups. These node groups typically communicate through a local, high-speed interconnect, which, in turn, can connect to a backbone or core switch so that all nodes can communicate with each other. Using compute units can influence where IBM Platform LSF dispatches work to get the best performance from this interconnect. For example, when you run large MPI jobs, you want to ensure the lowest latency between nodes by dispatching nodes that are connected to the same switch. Figure 3-5 shows the concept of compute units within a rack.

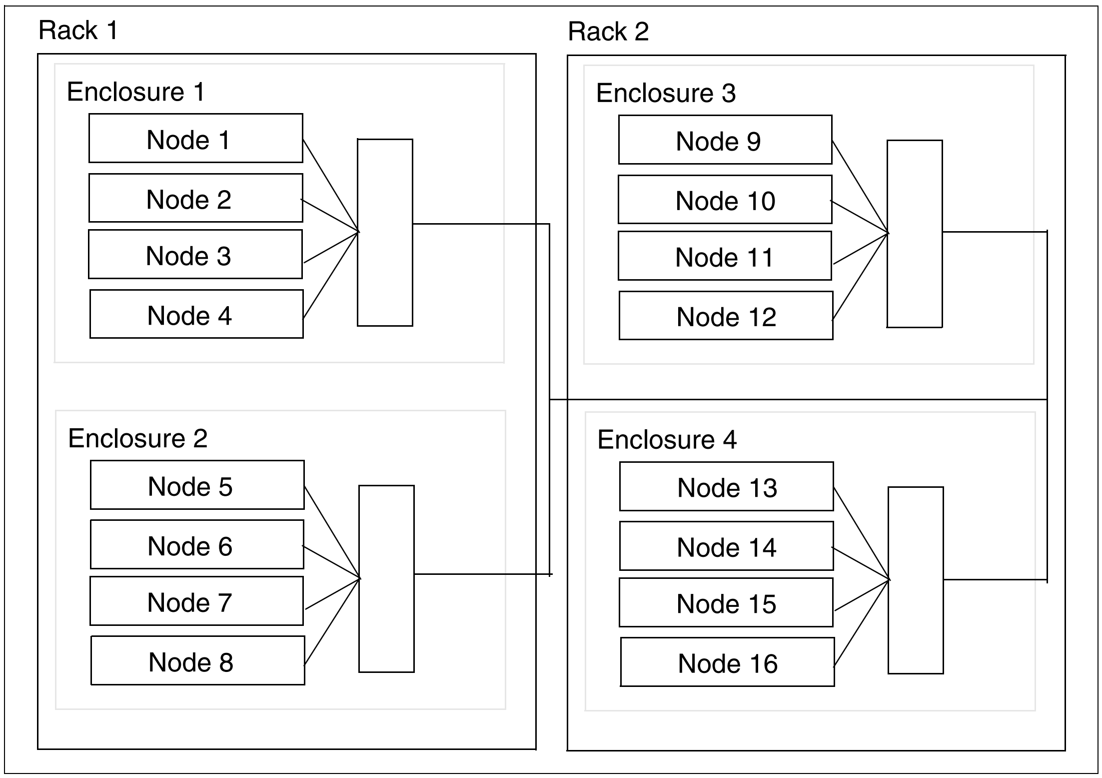


Figure 3-5 Showing compute units within racks

Compute units define a hierarchy within the cluster. Each enclosure shares a common local switch. These switches are then connected to other switches within the cluster. Running jobs on nodes within the same enclosure minimizes network hops between nodes and minimizes latency. Using compute units makes IBM Platform LSF topology-aware when scheduling jobs. When you use larger jobs, you can also ensure that jobs run on the same racks or balance work evenly between racks.

3.8.1 Configuring compute units

To create compute units, you must modify two files. First, you modify the `lsb.params` file to define the compute unit types. You add the following line to the `lsb.params` file before the `End` statement (Example 3-26).

Example 3-26 Defining the compute unit

```
COMPUTE_UNIT_TYPES = enclosure rack cabinet
```

We defined the hierarchy of our compute units as *enclosure*, *rack*, and *cabinet*. We then modified the `lsb.hosts` file to define the relationship between the nodes in the cluster as shown in Example 3-27.

Example 3-27 Defining the compute unit hierarchy

```
Begin ComputeUnit
NAME MEMBER TYPE
encl1 (i05n46 i05n47) enclosure
encl2 (i05n48 i05n49) enclosure
encl3 (i05n50 i05n51) enclosure
rack1 (encl1 encl2) rack
rack2 (encl3) rack
End ComputeUnit
```

Example 3-28 adds nodes i05n46 and i05n47 in enclosure *encl1*. Nodes i05n48 and i05n49 are in enclosure *encl2*. Both of these enclosures are in *rack1*. Nodes i05n50 and i05n51 are in a separate enclosure, which is also in a separate rack. To reread the configuration, you can use the **`badmin reconfig`** command. Then, you can verify the compute units by using the **`bmgroup`** command (Example 3-28).

Example 3-28 Adding the nodes into enclosures

```
$ bmgroup -cu
NAME          TYPE          HOSTS
encl1         enclosure    i05n46.pbm.ihost.com i05n47.pbm.ihost.com
encl2         enclosure    i05n48.pbm.ihost.com i05n49.pbm.ihost.com
encl3         enclosure    i05n50.pbm.ihost.com i05n51.pbm.ihost.com
rack1         rack         encl1/ encl2/
rack2         rack         encl3/
```

3.8.2 Using compute units

After you define compute units are, users can submit their jobs against the compute units by using the *CU* job resource requirement as shown in Example 3-29.

Example 3-29 Submitting jobs

```
$ bsub -n 16 -R "cu[type=enclosure;pref=minavail]" ...
```

In Example 3-29, *type* is the name of the compute unit that we defined in the `lsb.params` file in Example 3-26 on page 69. In this case, *type* is either *enclosure* or *rack*.

The `pref` value defines the order of the compute units. Compute units with a higher preference value are considered first during scheduling. In this case, compute units (or enclosures) with fewer free slots than compute units with more slots are preferable to enclosures with more free slots. This way, you can pack smaller jobs into enclosures and leave other enclosures free for larger jobs as shown in Figure 3-6 on page 71. This figure shows 16 hosts in each enclosure and each host with four slots.

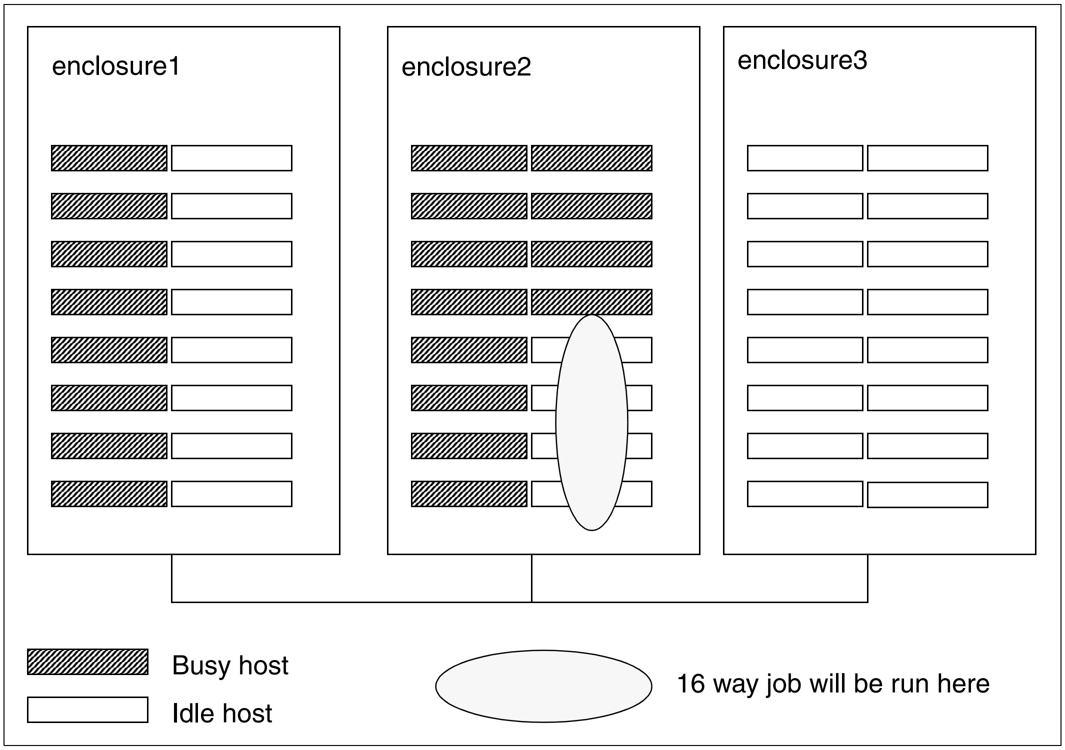


Figure 3-6 Enclosures status

In Figure 3-6, the scheduler selects *enclosure2* over the other enclosures, even though the other enclosures have more free slots available.

When you run a much larger job on a cluster with enclosures that are similar in size to the enclosures shown in Figure 3-6, use the **maxavail** parameter. This way, the jobs can be dispatched on enclosures with more free slots to maximize the jobs that are running in the same enclosure.

Figure 3-7 illustrates a situation of minimizing the number of enclosures used. In this case, we run a 96-slot job on enclosures with only 64 slots per enclosure so that the jobs must span enclosures.

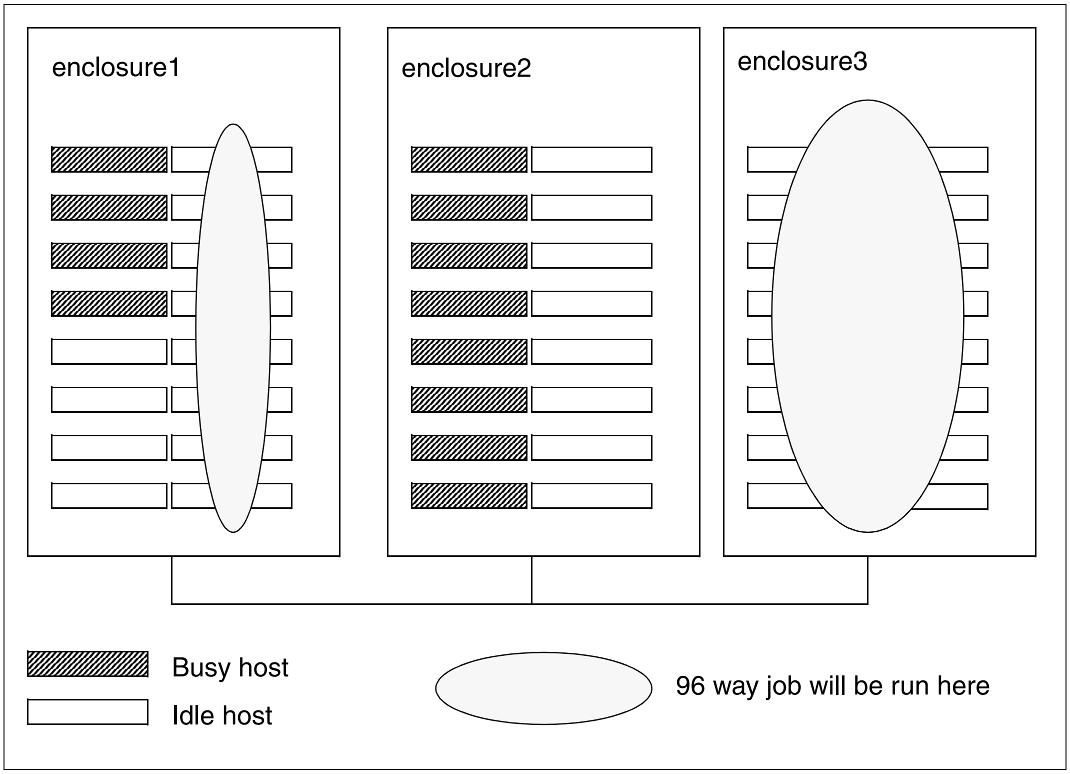


Figure 3-7 Jobs spanning enclosures

In Figure 3-7, the scheduler favors *enclosure1* and *enclosure2* because they have the most free slots, even though the job fits in *enclosure2*. You can also limit the number of compute units your job runs on. Suppose you have 64 slot enclosures and you want to ensure your jobs to run only in single enclosures. Example 3-30 shows the command to run within a single enclosure.

Example 3-30 Showing command to run within a single enclosure

```
$ bsub -n 32 -R ”cu[type=enclosure:maxcus=1]” ...
```

In Example 3-30, IBM Platform LSF ensures that the 32 slots are allocated from the same enclosure by setting the value `maxcus` to `1`. Jobs are rejected if too many slots are requested for the size of enclosures that are configured. By minimizing the number of enclosures, you can keep the job traffic localized within the enclosure.

By using compute units, you can also balance jobs equally across all the compute units. Alternatively, you can ensure to gain exclusive access to an entire compute unit, assuming the administrator allows it. This way, you can minimize traffic from other jobs that are running in the same enclosure from competing with your job, which might be necessary if you have jobs that are sensitive to other network traffic. Example 3-31 shows how to submit a job equally across all compute units.

Example 3-31 Balanced compute unit access

```
$ bsub -n 64 -R "cu[balance]" ....
```

The command that is shown in Example 3-31 on page 72 submits the job evenly among the compute units, using as few compute units as possible. Therefore, the command tries one compute unit first. If the first one is not possible to use, the command tries the second compute unit. If the second one is not possible to use, the command tries the third compute unit. For example, the scheduler places all 64 slots on one compute unit or 32 slots on two compute units.

In exclusive mode (Example 3-32), a job can start only when no other jobs are running within the compute unit. After the exclusive jobs starts, no other jobs can use nodes within that compute unit, which ensures that no other job can use the bandwidth for this enclosure.

Example 3-32 Exclusive compute unit access

```
$ bsub -n 64 -R "cu[excl:type=enclosure]" ....
```

Example 3-32 sets exclusivity on the enclosure, but you can do it at the rack level.

Exclusive mode: CU exclusive is not enabled by default. The administrator must enable it.

To enable exclusive mode, the administrator must enable exclusive access on the queue (see 3.5.4, “Exclusive scheduling” on page 59), or by adding the parameter that is shown in Example 3-33 to the queue definition.

Example 3-33 Enabling exclusive compute unit resources

```
EXCLUSIVE = CU[CU_TYPE]
```

In Example 3-33, CU_TYPE is the compute unit type that the administrator wants to allow exclusive access to. In this example, we used *enclosure*.

3.9 Application profiles

By using application profiles, you can improve the management of application submissions and the queues within the cluster. With application profiles, system administrators can map common execution requirements such as resources that are required for a common set of specific job containers.

Applications that require pre-execution or post-execution commands or resource limits can be grouped, which should reduce the number of queues and configurations within the cluster. Users will find it easier because they can specify the job characteristics for the job, rather than specifying them on the queue or the command line.

3.9.1 Defining application profiles

Application profiles are defined in the `lsb.applications` file. This file contains many examples and comments to help you build your own profiles. Example 3-34 shows a simple profile setup in the test environment.

Example 3-34 Sample application profile

```
Begin Application
NAME          = dyna
DESCRIPTION   = ANSYS LS-DYNA
CPULIMIT      = 48:0     # Max 48 hours of host
```

```
RUNTIME      = 24:0     # Default estimated run time (not hard limit)
MEMLIMIT     = 45000000 # Machines have 48GB memory - allow for O/S
PROCLIMIT    = 128             # job processor limit
PRE_EXEC     = /gpfs/fs1/lsdyna/scripts/testq_pre >> /tmp/pre.$$.out
POST_EXEC    = /gpfs/fs1/lsdyna/scripts/testq_post >> /tmp/post.$$.out
REQUEUE_EXIT_VALUES = 55 255 78
MAX_JOB_REQUEUE = 1
BIND_JOB     = BALANCE
End Application
```

Example 3-34 on page 73 shows a sample application profile for LS-DYNA. This parallel application has exclusive access to the hosts and will potentially run for a long period. This scenario uses the default run time of 24 hours, but a user can change this setting. We specify a run time because we might submit this job to an exclusive queue with a backfill scheduler. The processes are bound equally between the cores on each slot to maximize performance.

MEMLIMIT was imposed to prevent the job from using swap because Linux can overcommit memory. Pre-execution and post-execution scripts are specified to set up the job and possibly validate that the environment is ready to run the job. Requeue values are set so that, if the job fails, the job is automatically resubmitted (at least once). This feature is useful if the job fails over a weekend and the user is expecting the results on Monday. The job can run for a maximum of 48 hours, which is a hard limit, unlike RUNTIME to aid scheduling.

3.9.2 Using application profiles

After you define the application profile in the `lsb.applications` file, reconfigure IBM Platform LSF to make it aware of the changes. You can then list the available profiles and use them as shown in Example 3-35.

Example 3-35 List of available application profiles

```
$ bapp
APP_NAME            NJOBS      PEND       RUN      SUSP
dyna                    0         0         0         0

$ bapp -l dyna

APPLICATION NAME: dyna
 -- ANSYS LS-DYNA

STATISTICS:
   NJOBS     PEND       RUN     SSUSP     USUSP       RSV
      64       64         0         0         0         0

PARAMETERS:

 CPULIMIT
 720.0 min

 PROCLIMIT
 128

 FILELIMIT DATALIMIT CORELIMIT
  20000 K      20 G      20 G
```

```
REQUEUE_EXIT_VALUES: 55 255 78
PRE_EXEC: /gpfs/fs1/lsdyna/scripts/testq_pre >> /tmp/pre.$$.out
POST_EXEC: /gpfs/fs1/lsdyna/scripts/testq_post >> /tmp/post.$$.out

$ bsub -app dyna -n 32 -q exclusive -We 12:0
```

Because this parallel job uses the exclusive queue, which is managed by a backfill scheduler, we must submit an estimated job execution time, which we did by specifying an estimated run time of 12 hours. Otherwise, a default of 24 hours is assumed, which is the RUNTIME in the profile. After 48 hours, the job terminates because the CPULIMIT is set in the application profile.

3.10 Job submission prechecks and setup

Suppose you want to verify jobs that the users are submitting and are not doing anything outside the limits of the system. How do you validate all jobs that are submitted to the system before they are dispatched? Any job that does not meet certain requirements is rejected at the submission stage, rather than languishing on the queue, waiting for resources it does not have, or using all the resources it is not entitled to use.

You can also run scripts or programs before and after a job runs. You can use these scripts to validate that certain resources are available before the job starts or to take corrective action to ensure that the job runs and then clean up after execution. For more information about pre-execution and post-execution scripts, see 3.10.3, “Pre-execution and post-execution job scripts” on page 79.

Jobs can be validated by IBM Platform LSF when they are submitted or modified or when a checkpointed job is restarted by using the **esub** script. Example 3-36 shows a script that runs when the user submits the job. This script ensures that only the right user can submit jobs against the right project. First, you must create the `$LSF_SERVERDIR/esub` file, which contains the script that is shown in Example 3-36.

Example 3-36 Script that runs when a user submits a job

```
#!/bin/bash
if [ "$LSB_SUB_PARM_FILE" != " " ] ; then
        . $LSB_SUB_PARM_FILE
        # source bsub options
        USER=`whoami`
        if [ "$LSB_SUB_MODIFY" = "Y" ] ; then # bmod
                echo "Job has been modified">&2
        elif [ "$LSB_SUB_RESTART" = "Y" ] ; then # brestart
                echo "Job has been restarted">&2
        elif [ "$LSB_SUB_PROJECT_NAME" = "Project1" -a "$USER" != "sdenham" ]
        then
                echo "Only sdenham can charge Project1">&2
                exit $LSB_SUB_ABORT_VALUE
                # abort job submission
        else
                echo "Job submission accepted">&2
                exit 0
        fi
fi
```

Make sure that the new file is executable. We create the file as the root user and then modify it so that the lsfadmin user can modify the file. We now attempt to submit a job against this project as two different users to see what happens (Example 3-37).

Example 3-37 Submitting the job with two different users

```
[sdenham ~]$ bsub -P Project1 -q normal -n 4 sleep 10
Job submission accepted
Job <2231> is submitted to default queue <normal>.
[sdenham ~]$ logout

[root ~]# su - parkera
[parkera ~]$ bsub -P Project1 -q normal -n 4 sleep 10
Only sdenham can charge Project1
Request aborted by esub. Job not submitted.
```

As you can see, the job can be submitted only against the project by user sdenham.

The **esub** script is run only on the submission host and uses the credentials of the user who submitted the job. When the **esub** script runs, LSF defines the environment variables when the job is submitted. We can verify the variables with the rules that the administrators choose to write. These parameters are stored in the `$LSB_SUB_PARM_FILE` file. As shown in Example 3-36 on page 75, we source these values on line 2. For a complete list of variables type **man esub** or see Chapter 50, “External Job Submission and Execution Controls,” in *Administering IBM Platform LSF*, SC22-5346, at:

http://publibfp.dhe.ibm.com/epubs/pdf/c2253460.pdf

Example 3-38 shows the settings of the values.

Example 3-38 Sample submission values

```
LSB_SUB_QUEUE="normal"
LSB_SUB_PROJECT_NAME="Project1"
LSB_SUB_NUM_PROCESSORS=4
LSB_SUB_MAX_NUM_PROCESSORS=4
```

By looking at the variables that are shown in Example 3-38, you can see how rules can easily be written to validate user jobs. This example is written in shell, but you can choose a language, such as Perl or Python, to write your scripts. IBM Platform LSF determines whether to allow a job into the queue based on the exit code of the script.

Example 3-39 verifies that the user did not request too many processor slots.

Example 3-39 Script to check whether the user exceeded the number of slots

```
#!/bin/bash

# Define maximum number of slots a user can request
MAX_SLOTS=5

if [ "$LSB_SUB_PARM_FILE" != " " ]
   then
   . $LSB_SUB_PARM_FILE
   # source bsub options
   USER=`whoami`
   if [ ! -z "$LSB_SUB_NUM_PROCESSORS" ] ; then
      if [ $LSB_SUB_NUM_PROCESSORS -ge $MAX_SLOTS ]
```

```
        then
           echo "Maximum number of slots exceeded. Job terminated" >&2
           echo "You requested $LSB_SUB_NUM_PROCESSORS Max is $MAX_SLOTS" >&2
           exit $LSB_SUB_ABORT_VALUE
        fi
        echo "Job submission accepted" >&2
    fi
    exit 0
fi
```

When we run the request with too many slots, we see the output that is shown in Example 3-40.

Example 3-40 Verification that the number of slots is exceeded

```
$ bsub -n 10 sleep 10
Maximum number of slots exceeded. Job terminated
You requested 10 Max is 5
Request aborted by esub. Job not submitted.
```

3.10.1 Modifying a submitted job

The **esub** script can validate user scripts and correct them if the administrator specifies this task in their script. The script in Example 3-41 modifies the submission queue based on the user ID. Again we create the `$LSF_SERVERDIR/esub` file.

Example 3-41 Script to modify the queue based on a user ID

```
#!/bin/bash
if [ "$LSB_SUB_PARM_FILE" != " " ]
        then
        . $LSB_SUB_PARM_FILE
        # source bsub options
        USER=`whoami`
        if [ "$LSB_SUB_QUEUE" = "priority" -a "$USER" != "sdenham" ]
        then
                echo "Only sdenham can submit to priority queue" >&2
                echo "Changing the queue to the normal queue" >&2
                echo 'LSB_SUB_QUEUE="normal"' > $LSB_SUB_MODIFY_FILE
        fi
        echo "Job submission accepted" >&2
        exit 0
fi
```

Example 3-42 shows the output when you attempt to submit jobs.

Example 3-42 Verification of job queue changing script

```
[sdenham ~]$ bsub -n 4 -q priority sleep 10
Job submission accepted
Job <2245> is submitted to queue <priority>.
[sdenham ~]$ logout
[root ~]# su - parkera
[parkera ~]$  bsub -n 4 -q priority sleep 10
```

```
Only sdenham can submit to priority queue
Changing the queue to the normal queue
Job submission accepted
Job <2246> is submitted to queue <normal>.
```

In Example 3-42, the user *sdenham* can use the priority queue, but when user *parkera* attempts to use it, the job is automatically submitted to the normal queue instead of being rejected. You can use the **esub** script to validate every job that is submitted, verifying that the job has a valid wall clock limit and modifying this limit based on historical information. This process makes job scheduling more efficient.

3.10.2 Using multiple esub scripts

The examples in the previous sections use one **esub** script to validate *all* jobs that are submitted. You can specify an **esub** script at submission by using the **-a** flag at submission (Example 3-43).

Example 3-43 Submitting a job by using the esub -a script

```
$ bsub -a forecast -n 32 -q priority a.out
```

The script in Example 3-43 creates the `$LSF_SERVERDIR/esub.forecast` file. This script contains the validation that is required to submit this job.

> **Mandatory esub script:** You can define a mandatory **esub** script by modifying the `lsf.conf` file and adding the following line:
>
> `LSB_ESUB_METHOD="sitea"`
>
> Where `$LSF_SEVERDIR/esub.sitea` is the script that is run at submission. If this file does not exist, no job enters the queue.

Ultimately, you can run three **esub** scripts against a submitted job as shown in Figure 3-8.

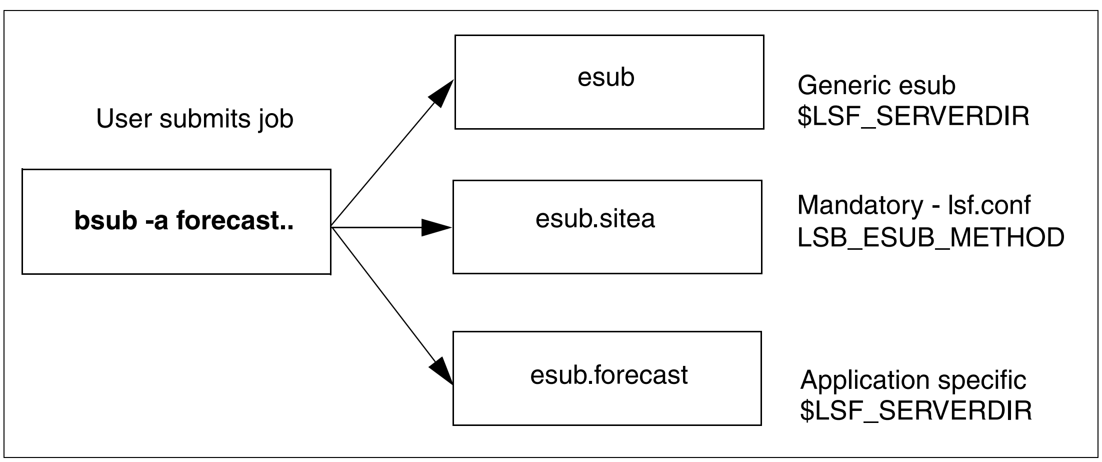


Figure 3-8 Using multiple esub scripts

Modifying one script to manage all jobs within a cluster is difficult. Therefore, it makes sense to use separate **esub** scripts for each type of job and then to use application profiles (see 3.9, “Application profiles” on page 73) to ensure that the right script runs against the right job.

3.10.3 Pre-execution and post-execution job scripts

In addition to **esub** scripts that check a job at the submission stage, you can run scripts or programs before the job runs and after the job has completed. You can use these scripts to verify the setup that is required for the environment for a successful job run. For example, the scripts can ensure that a certain file system is available, create working directories for the job, and, if necessary, take corrective actions. After the job completes, the script can copy the data back to a central file system or clean up after the job.

A user can submit a job with a pre-execution script by using the following syntax on the **bsub** script:

```
$ bsub -E /fullpath/pre_job job1
```

Where *fullpath* is the full path to the script on a shared file system. As the administrator, you can also set up a pre-execution script on a queue that runs *before* the user specified script. Example 3-44 shows the key information for specifying this option.

Example 3-44 Queue with a pre-execution script

```
Begin Queue
QUEUE_NAME = normal-q
DESCRIPTION = default queue
PRIORITY = 40 # lowest
PRE_EXEC = /gpfs/fs1/apps/bin/pre_job
HOSTS = i05n47
End Queue
```

The best way to run a pre-execution script is by using an application profile. By using this method, you can have different scripts for each application, rather than one script on the queue. Additionally users do not have to remember to specify the script when they submit a job. Example 3-45 shows an application profile with pre-execution and post-execution scripts.

> **Return code:** Any script that runs before the job, must exit with a return code of 0 (`rc=0`). If not, the job goes into a PEND state. If exit code 99 is set, the job exits without running.

Example 3-45 Application profile with pre-execution and post-execution scripts

```
Begin Application
NAME         = ZZTestApp
DESCRIPTION  = Test profile for ZZ jobs
RUNTIME      = 4:0
PROCLIMIT    = 32
PRE_EXEC     = /gpfs/fs1/apps/bin/ZZPre_exec
POST_EXEC    = /gpfs/fs1/apps/bin/ZZPost_exec
End Application
```

Post-execution scripts

Post-execution scripts are *always* run after the job exits, even if it exits abnormally. You can define these scripts on the queue, as shown in Example 3-46 on page 80, or at the job level by using the **bsub -Ep** command. You can use an application profile (as shown in Example 3-45) similar to what you use for the pre-execution scripts. If you use job scripts and queue scripts, the queue script runs *after* the job-level script.

Example 3-46 Queue with a post-execution script

```
Begin Queue
QUEUE_NAME = short-q
DESCRIPTION = short jobs
PRIORITY = 70 # highest
POST_EXEC = /gpfs/fs1/apps/bin/post_job
HOSTS = i05n47
RUNLIMIT = 5 10
End Queue
```

Tip: When you submit parallel jobs, the pre-execution and post-execution scripts run only on the first execution host.

3.11 Job resizing

By using job resizing, jobs can ramp up the number of resources they require or release resources early. Suppose you have a job that requires 12 cores, but initially the job can run on 4 cores. Toward the end, the job can run only on 1 core because the last parts of the job involve I/O (compiling the output file into a single file).

The job size changes during the lifecycle of the job, and its core or slot requirements change. By enabling *Resizable Jobs* on a queue, you can specify a minimum and maximum number of slots that are required. IBM Platform LSF dispatches the job when the minimum number of slots is available. Then, as more cores come available, IBM Platform LSF allocates these cores to the job and, optionally, runs a command to inform the job of the new resources. Toward the end of the job, the job can run the **bresize** command to release its slots. Other jobs can then use the released slots. If the last part of a job is I/O, releasing the resources allows other jobs to start faster. Figure 3-9 on page 81 illustrates this effect.

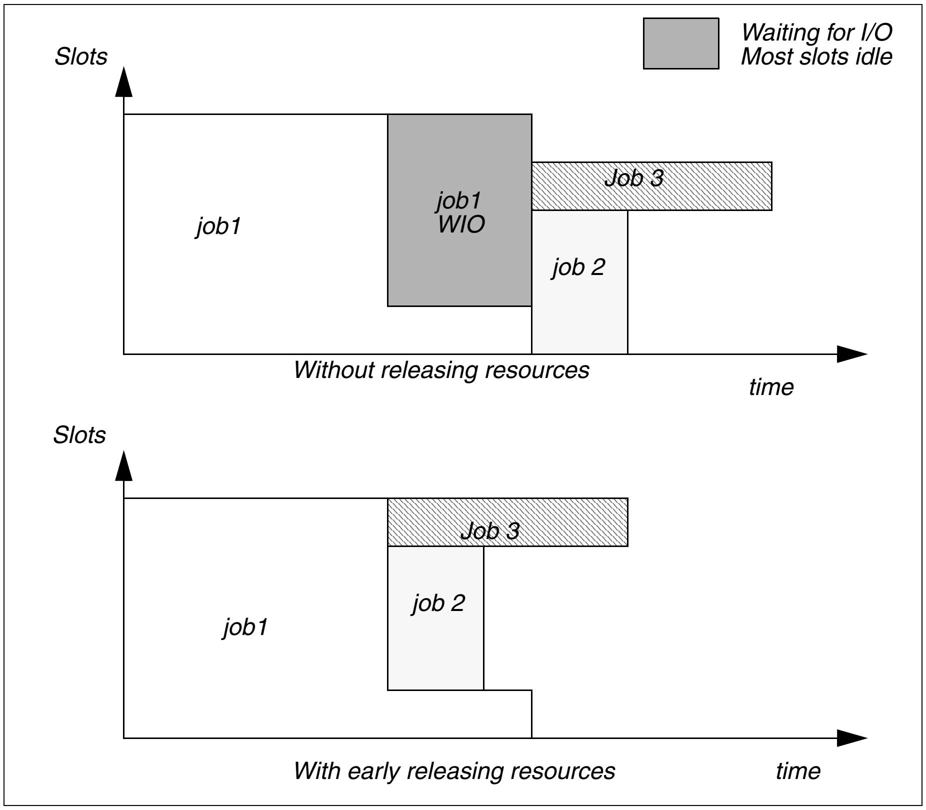


Figure 3-9 Job packing with early release

With job 1 releasing its resources early, jobs 2 and 3 can start, while job 1 finishes its I/O. The I/O needs only a few slots in comparison to the rest of the job, which needs many slots. Similarly, if job 3 is resizable, it can start with fewer slots and then use more slots as they become available, reducing its overall run time even more.

3.11.1 Enabling resizable jobs

Resizable jobs are enabled by default. However, the best way to use resizable jobs is with an application profile (see 3.9, "Application profiles" on page 73). Users can submit a job by using the **bsub** command as follows:

```
$ bsub -n 2,8 myresizablejob -rnc /gpfs/fs1/apps/resizecommand
```

The command requests a minimum of two slots and a maximum of eight slots. When resources are added or removed, the **resizecommand** command is run.

3.11.2 Using application profiles

By using application profiles, you have more control over jobs. After a job finishes with most of the resources, although it might have some post processing to complete, you can release its slots (resources) by using the **bresize release** command. By using this command, a job can release all of its slots (except one slot on the first execution host) or a selection of slots on selected nodes. This command runs on the first execution host of a job when slots are added or removed. This command is also used to inform the running job of a change to the number of slots.

Example 3-47 shows an application profile with the resizable options in bold.

Example 3-47 Application profile with resizable jobs

```
Begin Application
NAME         = adamsApp
DESCRIPTION  = Test profile for resizeable jobs
CPULIMIT     = 1:0     # Max 1 hours of host
RUNTIME      = 1:0     # Default estimated run time (not hard limit)
MEMLIMIT     = 45944508 # Machines have 48GB memory - allow for O/S
PROCLIMIT    = 12            # job processor limit
RESIZABLE_JOBS = Y # Allow jobs resizable
RESIZE_NOTIFY_CMD = /gpfs/fs1/apps/resize.sh
End Application
```

You then submit the jobs as shown in Example 3-48 and resize them during execution. As shown in Example 3-48, we request a minimum of two slots (up to a maximum of 4). IBM Platform LSF provides four slots. We then resize the job, releasing all the resources except for one slot. We anticipate that the job will normally resize itself, rather than letting an administrator (or the job owner) resize it.

Example 3-48 Submitting and resizing a job

```
$ bsub -n 2,4 -app adamsApp sleep 500
Job submission accepted
Job <3052> is submitted to default queue <normal>.
$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
3052    lsfadmi RUN   normal     i05n45.pbm. i05n47.pbm. sleep 500  Oct 26 10:50
                                             i05n47.pbm.ihost.com
                                             i05n47.pbm.ihost.com
                                             i05n47.pbm.ihost.com
$ bresize release all 3052
Job resize request is submitted.
$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
3052    lsfadmi RUN   normal     i05n45.pbm. i05n47.pbm. sleep 500  Oct 26 10:50
```

In the application profile, we specified a program to run when a **resize** command is received. This program gathers the environment variables as shown in Example 3-49. By using these environment variables, a job can calculate where it is running and releases the appropriate resources. Example 3-49 highlights some of the more interesting environment variables that help with this task. After all, we might not want to release all slots on all of the hosts.

Example 3-49 Resizable job environment variables

```
LSB_APPLICATION_NAME=adamsApp
LSB_BATCH_JID=3052
LSB_CHKFILENAME=/home/lsfadmin/.lsbatch/1351263015.3052
LSB_DJOB_HB_INTERVAL=10
LSB_DJOB_HOSTFILE=/home/lsfadmin/.lsbatch/1351263015.3052.hostfile
LSB_DJOB_HOSTFILE_MODTIME=1351263041
LSB_DJOB_HOSTFILE_SIZE=21
LSB_DJOB_NUMPROC=4
LSB_DJOB_RU_INTERVAL=10
LSB_ECHKPNT_RSH_CMD=ssh
```

```
LSB_EEXEC_REAL_GID=
LSB_EEXEC_REAL_UID=
LSB_EXEC_CLUSTER=cluster1
LSB_EXIT_PRE_ABORT=99
LSB_HOSTS='i05n47.pbm.ihost.com '
LSB_JOBEXIT_STAT=0
LSB_JOBFILENAME=/home/lsfadmin/.lsbatch/1351263015.3052
LSB_JOBID=3052
LSB_JOBINDEX=0
LSB_JOBNAME='sleep 500'
LSB_JOBPGIDS='25286 '
LSB_JOBPIDS='25286 '
LSB_JOBRES_PID=25286
LSB_JOB_EXECUSER=lsfadmin
LSB_MCPU_HOSTS='i05n47.pbm.ihost.com 1 '
LSB_QUEUE=normal
LSB_RESIZABLE=Y
LSB_RESIZE_EVENT=shrink
LSB_RESIZE_HOSTS='i05n47.pbm.ihost.com 3 '
LSB_RESIZE_NOTIFY_FAIL=1
LSB_RESIZE_NOTIFY_FAIL_CANCEL_REQ=2
LSB_RESIZE_NOTIFY_OK=0
LSB_SUB_HOST=i05n45.pbm.ihost.com
LSB_TRAPSIGS='trap # 15 10 12 2 1'
LSB_UNIXGROUP_INT=itso
LSFUSER=lsfadmin
LSF_BINDIR=/gpfs/fs1/lsf/8.3/linux2.6-glibc2.3-x86_64/bin
LSF_EAUTH_AUX_PASS=yes
LSF_EAUTH_CLIENT=user
LSF_EAUTH_SERVER=mbatchd@cluster1
LSF_EGO_ENVDIR=/gpfs/fs1/lsf/conf/ego/cluster1/kernel
LSF_ENVDIR=/gpfs/fs1/lsf/conf
LSF_INVOKE_CMD=bsub
LSF_LIBDIR=/gpfs/fs1/lsf/8.3/linux2.6-glibc2.3-x86_64/lib
LSF_LOGDIR=/gpfs/fs1/lsf/log
LSF_SERVERDIR=/gpfs/fs1/lsf/8.3/linux2.6-glibc2.3-x86_64/etc
LSF_VERSION=24
```

If you want to release two slots on host i05n47, you issue the following command:

```
$ bresize release "2*i05n47" 3052
```

Example 3-50 shows this command and the results of running it.

Example 3-50 Job submission with variable slot release

```
$ bsub -n 2,4 -app adamsApp sleep 500
Job submission accepted
Job <3053> is submitted to default queue <normal>.
$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
3053    lsfadmi RUN   normal     i05n45.pbm. i05n49.pbm. sleep 500  Oct 26 11:13
                                             i05n49.pbm.ihost.com
                                             i05n49.pbm.ihost.com
                                             i05n49.pbm.ihost.com
$ bresize release "2*i05n49" 3053
```

```
Job resize request is submitted.
$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
3053    lsfadmi RUN   normal     i05n45.pbm. i05n49.pbm. sleep 500  Oct 26 11:13
                                             i05n49.pbm.ihost.com
```

3.12 Idle job detection

Make sure that your resources are being used efficiently. As cluster owners, you want to get as close to 100% utilization as possible. Unfortunately users can submit jobs that hang or submit interactive jobs as batch jobs so that they hang while waiting for input. By using idle job detection, IBM Platform LSF identifies these idle jobs so that the administrator can intervene and take corrective actions.

By default, IBM Platform LSF checks for idle jobs every minute. You can tune this value by changing `EADMIN_TRIGGER_DURATION` in the `lsb.params` file. Setting the value *too low* can cause *false positives*, and setting the value *too high* might *not trigger exceptions* quickly enough.

To use idle job detection, you also need to set the `JOB_IDEL` parameter for the job queue. We modified our queue as shown in Example 3-51.

Example 3-51 Queue definition with idle detection

```
Begin Queue
QUEUE_NAME   = normal
EXCLUSIVE    = CU[enclosure]
PRIORITY     = 30
NICE         = 20
INTERACTIVE  = NO
JOB_IDLE     = 0.10
DESCRIPTION  = For normal low priority jobs, running only if hosts are lightly
loaded.
End Queue
```

The `JOB_IDLE` parameter value must be in the range 0 - 1.0. It represents the processor time or run time.

Example 3-52 shows the submission and start time of an idle job.

Example 3-52 Submitting an idle job

```
$ bsub -n 1 -app adamsApp sleep 800
Job submission accepted
Job <3565> is submitted to default queue <normal>.

$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
3565    lsfadmi RUN   normal     i05n45.pbm. i05n45.pbm. sleep 800  Oct 30 16:32
```

Example 3-53 shows the mail that is received from IBM Platform LSF within a few minutes of the idle job starting.

Example 3-53 Mail to the administrator showing the idle job

```
From lsfadmin@i05n45.pbm.ihost.com  Tue Oct 30 16:34:04 2012
Return-Path: <lsfadmin@i05n45.pbm.ihost.com>
Date: Tue, 30 Oct 2012 16:34:04 -0400
From: LSF <lsfadmin@i05n45.pbm.ihost.com>
Subject: Job Exceptions
To: lsfadmin@i05n45.pbm.ihost.com
Status: R

--------------------------------
 Idle jobs: (1)
--------------------------------

  JOB_ID         IDLE_FACTOR
  3565           0.00
```

3.13 Defining external resources (elims)

IBM Platform LSF gathers many indices, such as free memory, system load, and free slots, on each host. This process aids the scheduler in determining where to dispatch jobs. But what if you want to add your own values? For example, maybe you want to favor hosts that have more free space in their local scratch file system, nodes that have mounted a certain file system, or nodes that have a higher uptime.

You can write your own programs to gather external resource information and then give this information to the scheduler.

3.13.1 External resource gathering

First, you must decide what external resources you want to gather. Consider whether these resources are static or dynamic. These resources must be defined in the *Resource* section of the `lsf.shared` file.

You must also define the location of the resource in the *ResourceMap* section of the `lsf.cluster.clustername` file. In this example, we gather the amount of time that the server was started and the free space in the `/tmp` file system. These values are classified as dynamic resources because they change.

We modified the `lsf.shared` file as shown in Example 3-54 by inserting the lines that are highlighted in bold in the *Resource* section of the file.

Example 3-54 Defining external resources

```
Begin Resource
RESOURCENAME  TYPE    INTERVAL INCREASING  DESCRIPTION        # Keywords
uptime         Numeric        60      N       (Time in days server has been up)
scratch        Numeric        60      N       (Free space in MB in /scratch)
EndResource
```

You must follow several rules when you define a new resource. For example, the name cannot begin with a number or contain certain symbols. For information about these rules, type **`man lsf.cluster`** to access the online manual, or see the section about configuring the `lsf.shared` resources in *Administering IBM Platform LSF*, SC22-5346, at:

http://publibfp.dhe.ibm.com/epubs/pdf/c2253460.pdf

You then add the lines that are shown in Example 3-55 to the `lsf.cluster.clustername` file.

Example 3-55 Defining the scope of external resources

```
Begin ResourceMap
RESOURCENAME  LOCATION
uptime          ([default])
tmpfree         ([default])
End ResourceMap
```

Example 3-55 shows the external resource gathering program that runs on all hosts within the cluster (default). You can limit where the program is run based on individual hosts, the master host, or host groups. Running this script on *all* hosts in the cluster might not be beneficial if you have jobs that need exclusive access to all cores for long running jobs. Depending on how many interrupts your resource gathering script uses, it can slow down a long running job.

Alternatively you can run the script on the master host, which checks an external system, such as an external license server, for values.

Finally, you must write the **`elim`** script to gather the indices that you need. In this example, we installed the script that is shown in Example 3-56 in the `$LSF_SERVERDIR/elim` file. When you create the script, you must follow these rules for the script:

- Use the name **`elim`** or **`elim.name`**.
- Run the script in an endless loop.
- Place the script in the `$LSF_SERVERDIR` directory.
- Ensure that the script is executable and is owned by the root user.
- Verify that the script reports or generates output in the following format:

  `K` `name1 $value1 name2 $value2 nameN $valueN`

  Where:

  – *`K`* is the number of value pairs that are returned.
  – *`name`* is the name of the resource as defined in the `lsf.shared` file.
  – *`value`* is the value of the resource.

Test the script outside of IBM Platform LSF to ensure that it returns the right values before you integrate it into IBM Platform LSF. The script must be able to run on its own, and you must be able to verify the values that are returned (Example 3-56).

Example 3-56 External resource gathering script

```
#!/bin/sh
uptime=0
while true ; do
   # all hosts including master, will gather the following:
   upt=`uptime | grep days` # Catch machines that have been up minutes!
   if [ ! -z "$upt" ] ; then
      uptime=`echo $upt | awk '{print $3}'`
   fi
   # We don't have /scratch, so use /tmp instead!
```

```
   scrafree=`df -k /tmp | grep /tmp | awk '{print $3 / 1024}'`>&2
   echo 2 uptime $uptime scratch $scrafree
   # the same INTERVAL values defined in lsf.shared
   sleep 60
done
```

If you defined the script to run on certain nodes, verify whether the script is available on these hosts and verify the output there. After you refresh LIM and MBD on all hosts, you can verify the values by using the **lsload -l** command as shown in Example 3-57.

Example 3-57 Verifying values of external resources

```
$ lsload -l
HOST_NAME               status  r15s   r1m  r15m   ut    pg    io  ls    it   tmp   swp   mem uptime scratch
i05n45.pbm.ihost.com        ok   0.0   0.1   0.1  10%   0.0     3   1     0 3689M    4G   45G    4.0  3689.4
i05n46.pbm.ihost.com        ok   0.0   0.1   0.0  14%   0.0     4   0    19 3688M    4G   45G    3.0  3688.8
i05n47.pbm.ihost.com        ok   0.0   0.0   0.0   0%   0.0     2   1     0 3690M    4G   46G    0.0  3690.0
i05n48.pbm.ihost.com        ok   0.0   0.0   0.0   0%   0.0     1   0    27 3686M    4G   46G    4.0  3686.0
i05n49.pbm.ihost.com        ok   0.0   0.0   0.0   0%   0.0     1   0  5040 3686M    4G   46G    4.0  3686.0
```

In Example 3-57, you can see the external resource indices at the end of the line (highlighted in bold). For example, node i05n47 was up for less than one day.

3.13.2 Using an external resource

After the administrator installs and tests the external resources, users can use these resources as part of their job submission. For example, a user might dispatch a job only to hosts where the free space in /scratch is greater than 100 MB (Example 3-58).

Example 3-58 Job submission with external resources

```
$ bsub -R "scratch > 100" myApp
```

Alternatively, a user might dispatch only jobs to hosts that have been running for more than two days, favoring nodes that are running the longest (Example 3-59).

Example 3-59 Job submission that favors systems with a higher uptime

```
$ bsub -R "select[uptime > 2] order [uptime]" myApps
```

3.14 Using advance reservations

Consider a situation where you have to run a certain job, possibly everyday, to get output by a certain period. For example, you need to run a daily weather forecast for broadcast on the six o’clock news. How can you guarantee that the right resources will be available, at the right time, for your job to run? You can reserve a whole series of hosts, just for your job to run at certain key times, but this approach might be a waste of resources when your job is not running. The answer is to use an advance reservation.

With an advance reservation, you can reserve a certain set of resources for your jobs to run on and at fixed periods. Only authorized users or groups can use this resource during this period. New jobs are unlikely to be dispatched to these systems during the period that the advance reservation is active.

When a reservation becomes active, IBM Platform LSF attempts to run all the jobs that are associated with that reservation. Any job that was running before, but that is not associated with the reservation, is *suspended* on the resources that are defined by the reservation. The exception is if the reservation is not using all the resources that are allocated, in which case these jobs would continue as normal.

If you want to prevent jobs from being suspended, all jobs must be submitted with a run limit by using the **bsub -W** command. The **-W** parameter signals the scheduler to calculate how long a job needs, and if it might overlap with an advance reservation on a host, to avoid dispatching this job to that host.

3.14.1 Defining an advance reservation

By default, only administrators or the root user of IBM Platform LSF can define reservations. IBM Platform LSF users can view only the reservations and then, if authorized, they can use them. You can modify this default behavior by editing the `lsb.resources` file.

When you define a reservation, you must know the following information:

- The number of processors to reserve
- The list of hosts that are required
- The start time and end time of the reservation
- The users or groups that can use the reservation

Use careful planning with these parameters because, when a reservation ends, jobs that are running against the reservation are terminated, unless their termination time was modified by using the **bmod -t** command.

To create an advance reservation, you use the **brsvadd** command. In Example 3-60, we create a reservation against 32 processors, on three hosts, between 17:00 and 18:00 for the user sdenham.

Example 3-60 Adding an advance reservation

```
$ brsvadd -n 32 -m "i05n45 i05n46 i05n47" -u sandy -b 17:00 -e 18:00
Reservation sandy#0 is created
```

To check for advance reservations, enter the **brsvs** command as shown in Example 3-61.

Example 3-61 Checking for advance reservations

```
$ brsvs
RSVID       TYPE      USER       NCPUS          RSV_HOSTS      TIME_WINDOW
sandy#0     user      sandy      0/32     i05n45.pbm.:0/12     10/15/17/0-10/15/18/0
                                 i05n46.pbm.ihost.com:0/12
                                 i05n47.pbm.ihost.com:0/8
```

Example 3-61 creates a single reservation. For example, if you want user sandy to have exclusive access to this resource everyday, you set up a recurring reservation as shown in Example 3-62.

Example 3-62 Creating a recurring advance reservation

```
$ brsvadd -n 32 -m "i05n45 i05n46 i05n47" -u sandy -t "18:00-19:00"
Reservation sandy#2 is created
$ brsvs
RSVID        TYPE      USER       NCPUS          RSV_HOSTS      TIME_WINDOW
sandy#2      user      sandy      0/32     i05n45.pbm.:0/12     18:00-19:00 *
```

```
                                   i05n46.pbm.ihost.com:0/12
                                   i05n47.pbm.ihost.com:0/8
```

Example 3-62 shows the time window with an asterisk next to it, indicating a recurring reservation.

The system creates the reservation name based on the user ID or group ID that is specified in the command. You can also give the advance reservation a name by using the **-N** flag. The user should provide the name of the advance reservation, rather than using a system generated name (Example 3-63).

Example 3-63 Naming a recurring advance reservation

```
$ brsvadd -n 16 -m "i05n45 i05n46" -u sandy -t "15:00-16:00" -N "Forecast_PM1" -d
"Afternoon Forecast"
Reservation Forecast_PM1 is created

$ brsvs
RSVID        TYPE      USER       NCPUS          RSV_HOSTS     TIME_WINDOW
Forecast_PM1 user      sandy      0/16     i05n45.pbm.:0/12   15:00-16:00*
                                  i05n46.pbm.ihost.com:0/4
```

The name can help to avoid confusion when you decide which jobs to run against which advance reservation.

3.14.2 Using a reservation

After the administrator creates the reservation, you can use it. Example 3-64 shows the user sandy submitting a job against the advance reservation by using the **bsub** command.

Example 3-64 Using an advance reservation

```
denham@i05n45 $ brsvs
RSVID        TYPE      USER       NCPUS          RSV_HOSTS     TIME_WINDOW
sandy#2      user     sandy       0/32   i05n45.pbm.:0/12  18:00-19:00*
                                  i05n46.pbm.ihost.com:0/12
                                  i05n47.pbm.ihost.com:0/8

sandy@i05n45 $ bsub -U sandy#2 -n 2 sleep 500
Job <2220> is submitted to default queue <normal>.

sandy@i05n45 $ bjobs
JOBID  USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
2220   sandy   PEND  normal     i05n45.pbm.             sleep 500  Oct 15 17:47
sandy@i05n45 $ bjobs -l

Job <2220>, User <sandy>, Reservation <sandy#2>, Project <default>, Status
                     <PEND>, Queue <normal>, Command <sleep 500>
Mon Oct 15 17:47:51: Submitted from host <i05n45.pbm.ihost.com>, CWD <$HOME>, 2
                      Processors Requested;
 PENDING REASONS:
 Job is waiting for advance reservation to become active;
```

```
SCHEDULING PARAMETERS:
          r15s   r1m  r15m   ut      pg    io   ls    it    tmp    swp    mem
loadSched   -     -     -     -       -     -    -     -     -      -      -
loadStop    -     -     -     -       -     -    -     -     -      -      -
```

By using the **-U** flag with the **bjobs** command, the user can specify the name of the advance reservation to use. You can then see the job waiting for the advance reservation to become open. After the reservation is open, the job starts.

3.14.3 Modifying an advance reservation

By default, only IBM Platform LSF administrators or a root user can modify advance reservations. You can modify the reservation by adding or removing resources, by modifying the time window, or by disabling occurrences of the advance reservation. Example 3-65 shows how to delete an existing advance reservation by using the **brsvdel** command.

Example 3-65 Deleting an advance reservation

```
$ brsvdel sandy#0
Reservation <sandy#0> is being deleted
$ brsvs
No reservation found
```

You can also disable a single instance of a recurring advance reservation as shown in Example 3-66.

Example 3-66 Disabling a single advance reservation

```
$ brsvmod disable -tn  Forecast_PM1
Reservation will be disabled on Tue Oct 16.
The reservation cannot be reenabled for this period.
Are you sure? [y/n] y
Reservation Forecast_PM1 is modified

$ brsvs -l
RSVID         TYPE      USER       NCPUS         RSV_HOSTS     TIME_WINDOW
Forecast_PM1 user       sandy      0/16   i05n45.pbm.:0/12  15:00-16:00 *
                                    i05n46.pbm.ihost.com:0/4
Reservation Status: Inactive
Description: Afternoon Forecast
Reservation disabled for these dates:
   Tue Oct 16 2012
Next Active Period:
   Wed Oct 17 15:00:00 2012 - Wed Oct 17 16:00:00 2012
Creator: lsfadmin
Reservation Type: CLOSED
```

Tip: After you disable an occurrence of a recurring advance reservation, you cannot re-enable that occurrence.

It is likely that the users will request more resources for the reservation. By using the **brsvmod** command, you can add more hosts. Example 3-67 demonstrates the addition of another host (i05n47) with 12 processors to an existing reservation.

Example 3-67 Adding another resource to an advance reservation

```
$ brsvs
RSVID         TYPE      USER       NCPUS          RSV_HOSTS     TIME_WINDOW
Forecast_PM1  user      sandy      0/16    i05n45.pbm.:0/12   15:00-16:00 *
                                   i05n46.pbm.ihost.com:0/4

$ brsvmod addhost -n 12 -m "i05n47" Forecast_PM1
Reservation Forecast_PM1 is modified

$ brsvs
RSVID         TYPE      USER       NCPUS          RSV_HOSTS     TIME_WINDOW
Forecast_PM1 user      sandy      0/28    i05n47.pbm.:0/12   15:00-16:00 *
                                   i05n45.pbm.ihost.com:0/12
                                   i05n46.pbm.ihost.com:0/4
```

3.14.4 Creating a system reservation

You can also create a reservation so that no user jobs can run on a host or series of hosts. For example, the system administrators need to perform scheduled system maintenance, such as a firmware upgrade. In this case, you can create a system reservation so that only system tasks can be run. Example 3-68 shows the creation of a system reservation.

Example 3-68 Creating a system reservation

```
$ brsvadd  -m "i05n45 i05n46" -s -b 17:01:00 -e 17:02:00
Reservation system#1 is created
$ brsvs
RSVID        TYPE      USER       NCPUS          RSV_HOSTS     TIME_WINDOW
system#1     sys        -         0/24    i05n45.pbm.:0/12   10/17/1/0-10/17/2/0
                                  i05n46.pbm.ihost.com:0/12
```

Notice that we do not specify slots as the whole host is reserved for system use. This reservation is scheduled for the 17th of the current month between 01:00 and 02:00 in the morning on hosts i05n45 and i05n46.

3.14.5 Advance reservation accounting

How do you know whether users are making efficient usage of reservations? How do you know whether they reserve a whole chunk of machines everyday and then do not use them, wasting the resource? How do you know whether their reservation is the wrong size? To check the usage of a reservation, you can use the **bacct** command as shown in Example 3-69. This example shows that user sandy is not efficiently using the reservation, which consumes only 0.2 seconds of the hour-long reservation.

Example 3-69 Verifying reservation usage

```
$ bacct -U sandy#2
Accounting about advance reservations that are:
  - accounted on advance reservation IDs sandy#2,
  - accounted on advance reservations created by lsfadmin,
----------------------------------------------------------------------
```

```
RSVID        TYPE      CREATOR   USER    NCPUS       RSV_HOSTS     TIME_WINDOW
sandy#2     user     lsfadmin    sandy   32  i05n45.pbm.:12    18:00-19:00 *
                                       i05n46.pbm.ihost.com:12
                                       i05n47.pbm.ihost.com:8
SUMMARY:
Total number of jobs:           1
Total CPU time consumed:      0.2 second
Maximum memory of a job:      2.0 MB
Maximum swap of a job:      231.4 MB
Total active time:            1 hour  0 minute  0 second
```

3.15 Hyper-Threading technology

Hyper-Threading was introduced to improve the multithreading capabilities of some processors. Some applications work well with Hyper-Threading enabled. Applications with a mix of processing and I/O can take advantage of Hyper-Threading. However, some codes cannot, and enabling Hyper-Threading can have an even more detrimental impact on performance. Before you enable Hyper-Threading, check and test the characteristics of your application.

By default, IBM Platform LSF determines the number of cores on a host and schedules the jobs against the number of cores. Example 3-70 shows the output when checking cores.

Example 3-70 Core verification

```
$ lim -t
Host Type             : X86_64
Host Architecture     : x6_5866_IntelRXeonRCPUX5670293GHz
Physical Processors   : 2
Cores per Processor   : 6
Threads per Core      : 2
License Needed        : 12 core(s)
Matched Type          : X86_64
Matched Architecture  : x15_6789_IntelRXeon
Matched Model         : Intel_EM64T
CPU Factor            : 60.0

$ bhosts
HOST_NAME          STATUS       JL/U    MAX  NJOBS    RUN  SSUSP  USUSP    RSV
i05n45.pbm.ihost.c ok              -     12      0      0      0      0      0
i05n46.pbm.ihost.c ok              -     12      0      0      0      0      0
i05n47.pbm.ihost.c ok              -     12      0      0      0      0      0
i05n48.pbm.ihost.c ok              -     12      0      0      0      0      0
i05n49.pbm.ihost.c ok              -     12      0      0      0      0      0

$ lshosts i05n46
HOST_NAME      type    model  cpuf ncpus maxmem maxswp server RESOURCES
i05n46.pbm.  X86_64 Intel_EM  60.0    12    47G     4G    Yes (mg)
```

In Example 3-70 on page 92, each of system has 12 cores and, therefore, 12 slots per node. However, if you want to schedule by threads rather than by cores, you need to add the line that is shown in Example 3-71 to the `lsf.conf` file.

Example 3-71 Schedule by threads

```
EGO_DEFINE_NCPUS=threads
```

You then restart LIM by using the **lsreconfig** command, restart MBD by using the **brconfig** command, and run the same commands as shown in Example 3-72.

Example 3-72 Thread verification

```
$ bhosts
HOST_NAME          STATUS       JL/U    MAX  NJOBS    RUN  SSUSP  USUSP    RSV
i05n45.pbm.ihost.c ok              -     24      0      0      0      0      0
i05n46.pbm.ihost.c ok              -     24      0      0      0      0      0
i05n47.pbm.ihost.c ok              -     24      0      0      0      0      0
i05n48.pbm.ihost.c ok              -     24      0      0      0      0      0
i05n49.pbm.ihost.c ok              -     24      0      0      0      0      0

$ lshosts i05n46
HOST_NAME      type    model  cpuf ncpus maxmem maxswp server RESOURCES
i05n46.pbm.  X86_64 Intel_EM  60.0    24    47G     4G    Yes (mg)
```

As Example 3-72 shows, each host now has 24 slots. Remember to test your code to ensure that it works well with Hyper-Threading before you enable this feature.

> **Tip:** Having twice as many job slots does not affect your license. IBM Platform LSF is licensed by core, not by thread.

If you want a mixed environment, for example, where some machines have Hyper-Threading and others without Hyper-Threading must disable it in the BIOS. Example 3-73 shows how to disable Hyper-Threading on one of the nodes by using the IBM Advanced Settings Utility (ASU).

Example 3-73 ASU command to disable HT

```
$ asu64 set uEFI.ProcessorHyperThreading Disable --host 129.40.127.49
IBM Advanced Settings Utility version 9.20.77G
Licensed Materials - Property of IBM
(C) Copyright IBM Corp. 2007-2012 All Rights Reserved
Connected to IMM at IP address 129.40.127.49
uEFI.ProcessorHyperThreading=Disable
Waiting for command completion status.
Command completed successfully.
```

On other systems, you can usually enter the BIOS setup utility and disable HT. You then restart the host. Run the same **lshosts** command, and you see the results that are shown in Example 3-74.

Example 3-74 lshosts command results after restarting the host

```
$ lshosts
HOST_NAME      type    model  cpuf ncpus maxmem maxswp server RESOURCES
i05n45.pbm.  X86_64 Intel_EM  60.0    24    47G     4G    Yes (mg)
```

```
i05n46.pbm.  X86_64 Intel_EM  60.0    24    47G     4G    Yes (mg)
i05n47.pbm.  X86_64 Intel_EM  60.0    24    47G     4G    Yes ()
i05n48.pbm.  X86_64 Intel_EM  60.0    24    47G     4G    Yes ()
i05n49.pbm.  X86_64 Intel_EM  60.0    12    47G     4G    Yes ()

$ bhosts
HOST_NAME          STATUS       JL/U    MAX  NJOBS    RUN  SSUSP  USUSP    RSV
i05n45.pbm.ihost.c ok              -     24      0      0      0      0      0
i05n46.pbm.ihost.c ok              -     24      0      0      0      0      0
i05n47.pbm.ihost.c ok              -     24      0      0      0      0      0
i05n48.pbm.ihost.c ok              -     24      0      0      0      0      0
i05n49.pbm.ihost.c ok              -     12      0      0      0      0      0
```

As shown in Example 3-74, host i05n49 is now back to 12 cores per slot.

3.16 Changing the paradigm with guaranteed resources

You must configure the resources so that everyone gets their fair share of the cluster. It is acceptable for everyone to use your compute resource as a single entity. However, when different departments or projects fund the resource, things can become more complex. These departments or projects will pay for, and might expect, a certain amount of the cluster to be used exclusively by them.

By using traditional methods, you might set up different queues for each cost center, with some shared nodes as overflow between departments. Alternatively you can set up fewer queues and configure fair sharing between departments as explained in “Queue user hierarchical” on page 57. The only problem is that fairshare queuing assumes that everyone is busy all the time. If they have offset workloads, which are combined with long running jobs, one department might not be able to get onto the cluster.

Vertical thinking can result in the concept of towers and boundaries, which is inflexible. If one department has little or no work, some of the compute resource are wasted. Instead, you need to think across the cluster. You might establish the following policy for the departments:

> “You can have these resources that you paid for, but if you are not using them, we will *lend* them to someone else. However, when you need them back, they will be quickly returned”.

This policy is similar to the concept described in 3.6, “Goal-oriented scheduling” on page 61, where you set deadlines for jobs or a certain velocity of jobs. You must set up guaranteed resources.

3.16.1 Configuring guaranteed resources

Configuring guaranteed resources is similar to SLA-based scheduling. You set up a service class for the group of users that you want to use it. However, this time the goal is *guarantee*. Example 3-75 shows a service class definition with a goal of guaranteed resources.

Example 3-75 Service classes for guaranteed resources

```
Begin ServiceClass
NAME = blade_engine
GOALS = [ GUARANTEE ]
ACCESS_CONTROL = USERS[tmgr1 tmgr2] QUEUES[normal priority]
AUTO_ATTACH = Y
```

```
DESCRIPTION = Service class for the fan blade engineering team
End ServiceClass

Begin ServiceClass
NAME = design
GOALS = [ GUARANTEE ]
ACCESS_CONTROL = USERS[akumar alinegds] QUEUES[normal priority]
AUTO_ATTACH = Y
DESCRIPTION = Service class for design team
End ServiceClass

Begin ServiceClass
NAME = it
GOALS = [ GUARANTEE ]
ACCESS_CONTROL = USERS[denhams] QUEUES[normal priority]
AUTO_ATTACH = Y
DESCRIPTION = Service class for the IT group
End ServiceClass
```

Example 3-75 shows the lines that you add to the `lsb.serviceclasses` file. These lines define three groups of users. As the example shows, we still have queues. Therefore, users can prioritize or schedule different types of jobs. However, these queues might be few and global, spanning across the cluster.

Now define a guaranteed resource pool in the `lsb.resources` file. The guaranteed resource pool defines which slots or hosts are defined against the service class. They also define a policy to loan unused resources.

Example 3-76 shows how to define a resource pool that is called GROUP1, which sets five machines in the pool and guarantees three machines to engineering and one to design and IT. You do not have to enter specific values, and you can specify percentages.

Example 3-76 Defining guaranteed resource pools

```
Begin GuaranteedResourcePool
NAME = GROUP1
TYPE = hosts
HOSTS = i05n46 i05n47 i05n48 i05n49 i05n50
DISTRIBUTION = [blade_engine,3] [design,1] [it,1]
DESCRIPTION = A resource pool used by Blade engineering, Design and IT
LOAN_POLICIES = QUEUES[normal] CLOSE_ON_DEMAND
End GuaranteedResourcePool
```

Blade engineering does not necessarily get the same three machines each time, but rather they get three machines from the pool.

The defined loan policy defines that jobs that are submitted to the normal queue can borrow resources from other departments, if their own resources are full. However, if the owner wants their resources back, the running job is stopped. You can restrict the type of job that can be dispatched to the loaned hosts.

For example, if the IT departments jobs are typically short running, and engineering jobs are long running, but infrequent, you can define a duration on the load as shown in Example 3-77. Here, only jobs with a run time of less than 15 minutes are put on a loaned machine.

Example 3-77 Restricting jobs on loaned hosts

```
LOAN_POLICIES = QUEUES[normal] DURATION [15]
```

3.16.2 Using guaranteed resources

The users submit their jobs against the service class the same way as explained in 3.6.2, “Using a service class” on page 62, and as shown in Example 3-78.

Example 3-78 Submitting a job against an SLA

```
$ bsub -sla blade_engine -q normal myapp
```

3.16.3 Viewing guaranteed resources

To examine the guaranteed resources, use the **bresources** command as shown in Example 3-79. The result shows how many machines are in the pool and who they are guaranteed for.

Example 3-79 Viewing guaranteed resources

```
$ bresources -g -l -m
GUARANTEED RESOURCE POOL: GROUP1
A resource pool used by Blade engineering, Design and IT

TYPE: hosts
DISTRIBUTION: [blade_engine, 3] [design, 1] [it, 1]
LOAN_POLICIES: QUEUES[normal] CLOSE_ON_DEMAND
HOSTS: i05n46 i05n47 i05n48 i05n49 i05n50

STATUS: ok

RESOURCE SUMMARY:
  TOTAL                        5
  FREE                         5

  GUARANTEE CONFIGURED         5
  GUARANTEE USED               0

                                              GUAR        GUAR       TOTAL
CONSUMERS                                   CONFIG        USED        USED
blade_engine                                     3           0           0
design                                           1           0           0
it                                               1           0           0

Hosts currently in the resource pool:
i05n46 i05n47 i05n48 i05n49 i05n50
```

4

Industry solutions from IBM Platform Computing

High-performance computing (HPC), workload management, big data analytics, and cluster management apply to several industries, with different solutions for each industry.

This chapter includes the following sections:

- Available solutions
- HPC Cloud
- Workload Management
- Big Data Analytics
- Cluster management
- Solutions by industry

4.1 Available solutions

IBM Platform Computing is a set of systems software that helps with HPC, including computationally and data-intensive design, manufacturing, financial analytics, business, and research applications. IBM Platform Computing solutions optimize complex application implementation and workloads in many of the world's largest environments (with more than 100,000 cores). The core value of the product portfolio is to simplify and accelerate high-performance simulations and analysis to help discovery of insights into business, products, and science.

Many organizations have the constant challenge of increasing compute capacity to support massive amounts of data that drive business value and competitive advantage. IBM Platform Computing solutions simplify the setup, integration, and management of your heterogeneous technical computing infrastructure and drive up server utilization to increase application throughput and to help greatly improve time to results. IBM Platform Computing software also helps integrate servers, storage, parallel execution environments, and applications. Together, they enable the delivery of complete solutions that simplify and accelerate implementation and management of high-performance clusters, grids, and HPC clouds. Business value is delivered in days versus weeks or months.

IBM Platform Computing solutions have the following benefits:

- Achieve high-quality results quickly.
- Reduce the costs of management and infrastructure.
- Adapt easily to changing requirements requested by users.

Table 4-1 shows a list of IBM Platform Computing solutions for HPC Cloud, workload management, big data analytics, and cluster management versus products available.

Table 4-1 IBM Platform Computing Solution versus available products

IBM Platform	HPC Cloud	Workload Management	Big Data Analytics	Cluster Management
LSF	✓	✓		
HPC	✓	✓		✓
Symphony	✓	✓	✓	
Cluster Manager - Advanced Edition	✓			✓
Application Center	✓			
Process Manager	✓			

IBM Platform Computing solutions complement IBM systems and technology portfolio, helping to provide simplified management software to eliminate the complexity of optimizing cluster, grid, and HPC cloud environments. The following products are used with IBM Platform Computing solutions:

- IBM Platform LSF

  This product is a powerful workload management platform for demanding, distributed HPC environments. It provides a comprehensive set of intelligent, policy-driven scheduling features so that you can use all of your compute infrastructure resources and ensure optimal application performance.

- IBM Platform HPC

  This product offers a complete HPC, easy-to-use management software solution. It includes a range of features that are ready for immediate use, such as cluster management, workload management, reporting, Message Passing Interface (MPI).

- IBM Platform Symphony

  This product offers a high-performance grid middleware and management solution that runs on your choice of hardware and operating environments.

- IBM Platform Cluster Manager (Advanced Edition)

  This product provides a single pane to monitor and manage the entire infrastructure. By using the management portal, with workload and resource monitoring, alerting and troubleshooting, administrators can manage more machines through a single point of administration.

For more information about IBM Platform Computing portfolio, current versions, and the software roadmap of products, see Chapter 1, “Products and portfolio for IBM Platform Computing solutions” on page 1. See also *IBM Platform Computing Solutions*, SG24-8073.

4.2 HPC Cloud

Growth in specialized computing programs often leads the number of infrastructure deployments. This approach has restricted cooperation and source optimization across divisions and workflows. IBM HPC Cloud solutions provide easy-to-use resource management techniques to effectively and separately (systems, groups, or grids) run the resources in an infrastructure that is enhanced for highly effective and specialized processing.

4.2.1 HPC Cloud challenge

Technological innovation, medical, financial, and research workloads have placed outstanding performance requirements on specialized and powerful processing (HPC) infrastructures. Traditionally such workload requirements were not feasible for the available infrastructure deployments, even though those workloads could use virtualization technology. However, the performance gap between physical and exclusive deployments is now ending. HPC and specialized processing users can now benefit from the versatility, cost effectiveness, and enhanced resource that reasoning provides.

4.2.2 HPC Cloud solution

Cloud technology, when carefully and flexibly used, increases the advantages and reduces the disadvantages for HPC. The emerging cloud technology and, in particular, IBM Platform Computing solutions for HPC Cloud provide clients with an infrastructure value proposition for HPC workloads.

IBM Platform Computing solutions for HPC Cloud convert static computing resources into a versatile high performance cloud. This cloud can be shared, remotely managed, and easily provisioned to support the requirements of compute-and data-intensive workloads and changing individual requirements. This way, your organization can set up an efficient, consolidated infrastructure that meets time-variant business, supports research demands, and offers the performance that your users demand.

These cloud offerings are based on proven cluster, grid, and HPC cloud technology from IBM Platform Computing. They have the following features to help ensure retail, enterprise, or community resources are optimally allocated and easy to manage:

- Versatile individual connections
- Automated workload-based virtual and physical machine provisioning
- Solid “resource-aware” job scheduling and migration
- Powerful self-service implementation capabilities

By using these tools, you can position resources with main concerns of a project. You can also realize enhanced functional effectiveness and cost effectiveness, while accomplishing business goals that are difficult to achieve with other management systems.

4.2.3 HPC Cloud advantage

By applying cloud technology and requirements to established technical computing and HPC infrastructures, you can remove costly, rigid silos. You can use distributed computing resources to help you achieve the following objectives:

- Improve efficiency and assist in cooperation with an easy-to-use, web-based visualization tools
- Improve data access, resource availability, and security, and enhance performance
- Improve throughput of heterogeneous workloads across different resources
- Improve automated and reduce manual effort for enhanced administrator productivity
- Reduce operating costs and total cost of ownership with a consolidated high-performance infrastructure

4.3 Workload Management

IBM Platform Computing offers a range of workload management abilities for demanding, allocated HPC environments. With an extensive set of intelligent, policy-driven scheduling features, you can gain the most from your estimate resources and data control infrastructure to improve application performance.

4.3.1 Workload Management challenge

Companies in nearly every industry are working with growing data volumes and development of application demands. These challenges drive the need for more compute capacity. Even in HPC environments, having several compute silos, irregular processing, design cycle issues, and late results are common. Today, you need ways to improve IT performance, reduce infrastructure costs, and bring your products to market quicker.

4.3.2 Workload Management solution

The IBM Platform Computing family offers a range of workload management capabilities. The IBM Platform LSF product family is a powerful workload management platform for demanding, distributed, and mission-critical HPC environments. It provides a comprehensive set of intelligent, policy-driven scheduling features so that you can take full advantage of your compute infrastructure resources and ensure optimal program performance. By having a highly scalable and available structure, you can schedule complex workloads and manage up to petaflop-scale resources.

IBM Platform Symphony software can deliver faster, better quality results, even when using less infrastructure. With the versatility to adapt when main concerns change, IBM Platform Symphony can reallocate over 1,000 compute engines per second to different workloads, based on policies and main concerns that you determine. The result is better program application, better usage, and an ability to respond quickly to business-critical demands. IBM Platform Symphony Advanced Edition contains a MapReduce implementation that is compatible with Apache Hadoop and optimized for low latency, reliability, and resource sharing.

IBM Platform HPC provides easy-to-use cluster provisioning and management and has a solid workload scheduling capability, which is based on IBM Platform LSF. By scheduling workloads intelligently according to plan, IBM Platform HPC improves user efficiency with minimal system management effort. In addition, HPC user groups can interact with application-centric interfaces, easily share computing resources and data, and reduce time between simulator iterations.

4.3.3 Workload Management advantage

Workload management solutions for IBM Platform Computing can help you achieve the most from your HPC investment. By using these solutions, you can take full advantage of all specialized computing resources, from application software licenses to available network bandwidth. IBM Platform Computing solutions can help in the following ways:

- Reduce functional and infrastructure costs by providing optimal service-level agreement (SLA) management and by providing greater versatility, visibility, and control of job scheduling.
- Improve efficiency and resource sharing by fully using hardware and application resources, whether they are just down the hall or halfway around the globe.
- Use investment strategies in current resources by combining pooling resources and managing application workloads across highly allocated environments, from single and local departmental clusters to a globally dispersed, multicluster infrastructure.

4.4 Big Data Analytics

With information that is growing at outstanding rates and users who demand quick, effective research of this information, your programs need a powerful base. IBM Platform Computing software improves the performance of your computing infrastructure for your most demanding programs.

4.4.1 Big Data Analytics challenge

With more intelligent and connected devices and systems, the amount of information that you are collecting is increasing at alarming rates. In some sectors, as much as 90% of that information is unstructured and increasing at rates as high as 50% per year. To keep your business competitive, to innovate, and to get products and solutions to market quickly, you must be able to evaluate that information and extract insight from it easily and economically. For big data analytics, current alternatives do not offer the response time for statistical tasks, reducing user efficiency and delaying decision making.

4.4.2 Big Data Analytics solution

IBM Platform Computing software improves the performance of your most demanding applications with a low latency solution for heterogeneous application integration on a shared multitenant architecture. IBM Platform Symphony Advanced Edition provides a high-performance distributed runtime engine for Hadoop MapReduce programs. The software provides great resource availability and predictability. It also supports several programs and file systems, operational maturity, SLA policy control, and high resource utilization for MapReduce and applications that are not MapReduce.

With years of experience in distributed workload scheduling and management, IBM Platform Computing offers proven technology that powers objective, critical, and the most demanding workloads of many large companies. IBM Platform Symphony software offers unmatched distributed workload runtime services for distributed computing and big data statistics programs.

IBM Platform Symphony software uses marketing methods that allow sub-nanosecond response and quick provisioning for a wide range of workloads. Short running jobs have a smaller percentage of time that is spent in the provisioning and deprovisioning actions, providing a higher ratio of useful work to overhead. It also has a high job throughput rate, where the system allows more than 17,000 tasks/second to be submitted.

4.4.3 Big Data Analytics advantage

IBM Platform Symphony provides the following advantages for your big data analytics applications:

- Policy-driven workload scheduler for better granularity and control
- Several instances of Hadoop, other programs, or both on a single shared cluster
- Distributed runtime engine support for high resource availability
- Flexibility from open architecture for application development and choice of file system
- Higher application performance for IBM InfoSphere BigInsights workloads
- Rolling software upgrades to keep applications running

4.5 Cluster management

Separated clusters can create major issues in an HPC environment and prevent companies that require significant compute and data processing capabilities from competing. IBM can help relieve this complexity with tools for the self-service creation and management of flexible clusters.

4.5.1 Cluster management challenge

Competitive organizations that require increasing amounts of compute and data processing capabilities face difficulties in maintaining complex, diverse HPC components with different software and hardware. Data processing capabilities increases compute-intensive HPC data processing capabilities infrastructures and make administration and control easier through one central interface.

4.5.2 Cluster management solution

IBM Platform Cluster Manager Advanced Edition enables self-assistance creation and management of flexible clusters to deliver the performance that is required by the compute-intensive workloads in HPC environments. Automated, self-service setup of multiple HPC environments means more timely provisioning.

IBM Platform Cluster Manager consolidates different cluster infrastructures of technical computing and statistics workloads. It also provides a single management pane to simplify management and maintenance of running the infrastructure. The dynamic capabilities of Platform Cluster Manager Advanced Edition are the foundation for building an HPC cloud.

4.5.3 Cluster management advantage

IBM Platform Cluster Manager Advanced Edition consolidates clusters into a more efficient infrastructure. The result is the creation of an agile environment for computing and analytics workloads to help in the following ways:

- Speed implementation of HPC clusters
- Improve user efficiency
- Increase the ability to meet and exceed SLAs
- Lower operating expenses

4.6 Solutions by industry

With over 2,000 customers globally, experience with single site to large international grid solution and programs that span a range of capabilities for high performance and technical computing, IBM Platform Computing can support key solutions in your industry.

4.6.1 Aerospace and defense industry

IBM Platform Computing offers a full range of solutions to address challenges that are specific to the aerospace and defense industry to improve the implementation and management of HPC environments. By using various software and services, the aerospace and defense industry can develop powerful, versatile HPC clusters and HPC cloud environments for fast product development to remain competitive.

Aerospace and defense challenge

The aerospace and defense sectors are increasing technical computing demands. As a result, they are driving high expectations in technological innovations. This industry must do more with less to continue to create cutting-edge ideas and items while adapting to new economic realities. Product design and development require a complex and highly scientific process. Simulation of real-world conditions that products face, including atmospheric pressure, temperature, and structural load, is critical for success. The aerospace and defense industry needs precise, hyper-efficient data processing and research.

Aerospace and defense solution

IBM Platform Computing solutions include items for workload and process management, HPC application integration, job submission and management, license scheduling, analytics and visualization, and MPI implementation. Several IBM Platform Computing solutions are ideal for application in every aspect of the aerospace and defense industry. For example, these products provide structural mechanics, finite element analysis, computational fluid dynamics, explicit dynamics, multiphysics, and computer-aided design. They also facilitate various in-house developed applications.

IBM Platform Computing solutions are quick and easy to set up, with simple HPC cluster installation and control and offer preintegration of ISV applications. They accelerate engineering simulations and offer users transparent access to distributed computing resources.

IBM Platform Computing solutions have been developed in cooperation with leading aerospace and defense manufacturers. They are supported by thousands of IBM professionals with worldwide aerospace and defense industry experience. In addition, they offer a complete set of product capabilities for long-term solutions with massive scalability.

4.6.2 Automotive industry

In the current fast moving *design anywhere, build anywhere* environment, automobile organizations need speed, agility, control, and visibility across the design ecosystem and lifecycle to meet time-to-market requirements and maximize productivity. IBM Platform Computing solutions can help automotive companies transform the design chain to develop designs better, faster, and cheaper.

Automotive challenge

Many of the improvements in fuel efficiency in today's automobiles are tied directly to innovative ways to integrate, for example, electronically controlled engines, braking systems, and battery power for increased efficiency and waste elimination. These connected systems need to operate effectively and in balance with safety systems, diagnostic tests, and drivers.

Companies in the automotive industry need a structured procedure and the IT infrastructure to support such a process to convert research and ideas into reality, in a fast, efficient, and cohesive manner.

Automotive solution

IBM Platform Computing solutions include products for workload and process management, HPC application integration, job submission and management, license scheduling, analytics and visualization, and MPI implementation. Several IBM Platform Computing solutions are ideal for application in every aspect of the automotive industry. For example, these products provide architectural mechanics, limited element analysis, computational fluid dynamics, explicit dynamics, and multiphysics. They also facilitate various in-house developed applications.

4.6.3 Aerospace, defense, and automotive advantage

IBM Platform Computing solutions help create dynamic, versatile clusters and HPC cloud environments to help the aerospace, defense, and automotive industries achieve a range of advantages:

- Improved infrastructure usage with pools of shared resource
- Higher application service levels and throughput
- Decreased costs by using a heterogeneous, shared infrastructure
- Enhanced collaboration
- Simple installation and implementation to reduce time and cost of IT administration

4.6.4 Financial markets and insurance industry

Financial services companies are facing increasingly limited economic demands and growing regulating requirements. As a result, they are looking for better ways to meet market needs. They are looking for ways to improve IT performance to meet these requirements and to reduce managing costs. IBM Platform Computing solutions can help.

Financial markets and insurance challenge

The biggest challenge for organizations in the financial and insurance industries is risk management. Rules, such as Basel III and Solvency II, are forcing these organizations to evaluate risk in near real time. In particular, they must examine the areas of credit and liquidity by using statistics to understand large amounts of information almost immediately. The applications that meet these difficulties place a large strain on computing resources.

Financial markets and insurance solution

IBM Platform Computing solutions can help you improve the performance of financial and insurance programs to support faster, more precise, and more reliable decision-making in markets. By using private HPC clusters, grids, and clouds, multiple applications and lines of business can effectively use a single heterogeneous, shared infrastructure to support the following areas:

- Costs of market and credit risk
- Compliance reporting
- Pretrade analysis
- Back testing
- New product development

IBM Platform Symphony is a powerful, enterprise-class grid management solution for companies that are running compute-intensive and data-intensive applications. IBM Platform Symphony accelerates many parallel applications that quickly compute results and create optimal use of the available infrastructure. This product provides the speed that is required to predictably meet and surpass throughput goals for the most demanding risk management applications.

The application software solutions from IBM Platform Computing help manage and accelerate workload processing and realize workloads across a distributed, shared IT environment, while fully using all HPC resources, regardless of operating system or architecture. The result is improved application performance for a reduced total cost of ownership.

Financial markets and insurance advantage

Financial and insurance companies need to get the most from their IT investment, with a particular focus on infrastructure capacity, utilization, and performance. IBM Platform Computing solutions can help perform more rigorous research with more information to properly simulate market and credit risk and to meet regulating requirements. IBM Platform Computing solutions offer the following support:

- Focused technical and distributed computing management software helps to create, integrate, and manage shared computing environments that are used in compute-intensive and data-intensive applications.
- A specific systems control environment instantly optimizes IT resources and speeds application system implementation, providing IT services effectively while decreasing management and administration costs.

4.6.5 Life sciences

Life sciences companies face huge difficulties in direction and efficiency. To increase development efficiency, innovate in research and growth, and contend more effectively, companies must identify an enhanced, versatile, and resilient infrastructure foundation to improve clinical development processes. IBM Platform Computing solutions can help address these challenges.

Life sciences challenge

With shifting regulatory burdens and the need to compress the timeline from discovery to approval, research teams need comprehensive, high performance technical computing infrastructure solutions. These solutions must have the versatility to process massive amounts of data and support progressively innovative studies.

Life sciences solution

IBM Platform Computing solutions provide cluster, grid, and HPC cloud management software to support HPC components and specialized computing application. These offerings speed up time to results for compute-intensive and data-intensive applications that run on distributed technical and statistics computing environments. They help to manage workloads as different as genome sequencing, drug design, simulations, product design analysis, and risk management.

The IBM Platform Computing product family helps to improve the confidence of IT professionals in life sciences, ensuring them that all available resources are used, from application software licenses to available network bandwidth. Application software solutions help manage and accelerate workload processing and guarantee their achievement across a distributed, shared IT environment, while completely using all HPC resources, regardless of operating system or architecture. Scheduling capabilities help spend resources on users, groups, and jobs in a way that is consistent with SLAs. With extended SLA-based scheduling policies, these application solutions simplify administration and ensure optimal alignment of business SLAs with available infrastructure resources.

Life sciences advantage

By enhancing usage, resources are more available so that scientists can complete more work in a smaller period. This advantage can enhance cooperation across the clinical development value chain for better insights and excellent outcomes. IBM Platform Computing solutions help in the following ways:

- Optimize resource usage and reduce costs with robust workload management.
- Increase throughput for quicker time-to-results with intelligent scheduling.
- Improve efficiency with simple, intuitive management.

5

Security considerations for IBM Platform Computing solutions

Security breaches have become a topic for discussion everywhere, from boardrooms to blogs to major media. Executives who are held responsible for critical corporate, customer, employee, investor, or partner data want to reconcile and understand how well they are prepared in this era of cyber attacks.

This chapter focuses on security considerations for the IBM Platform Computing solutions that are highlighted in this book. This chapter includes the following sections:

- IBM Platform HPC
- IBM Platform LSF
- IBM Platform Symphony

5.1 IBM Platform HPC

IBM Platform HPC is a comprehensive integrated solution for HPC users. It is a one-stop solution for cluster management, cluster monitoring, workload management, and workload monitoring. Platform HPC can help with many tasks because it is the integration of several IBM Platform Computing products such as IBM Platform LSF, IBM Platform Application Center, and IBM Platform Cluster Manager.

By default, user authentication for IBM Platform HPC is based on operating system (OS) users. However, you can also configure Network Information Service (NIS) or Lightweight Directory Access Protocol (LDAP) for IBM Platform HPC for authentication. To configure NIS or LDAP, see the instructions in *Administering IBM Platform HPC*, SC22-5379, at the following address:

http://publibfp.dhe.ibm.com/epubs/pdf/c2253790.pdf

This same manual is in the `platform_hpc_admin.pdf` file that ships with IBM Platform HPC.

In IBM Platform HPC, the user names and passwords that are defined on the master host are used across all hosts in the cluster. Therefore, use external LDAP or NIS. Otherwise, you need to synchronize the user names and passwords across the cluster by using the **`kusu-cfmsync`** command. To synchronize the user names and passwords, see *Administering IBM Platform HPC*, SC22-5379, at the following address:

http://publibfp.dhe.ibm.com/epubs/pdf/c2253790.pdf

This same manual is in the `platform_hpc_admin.pdf` file that ships with IBM Platform HPC.

IBM Platform HPC has two categories of users: the administrator and the users who submit jobs. By default when you install IBM Platform HPC, the `hpcadmin` user is created in the OS, who is the administrative user. Figure 5-1 shows how to assign the administrator role to a user. It also shows the tabs that an administrator can view.

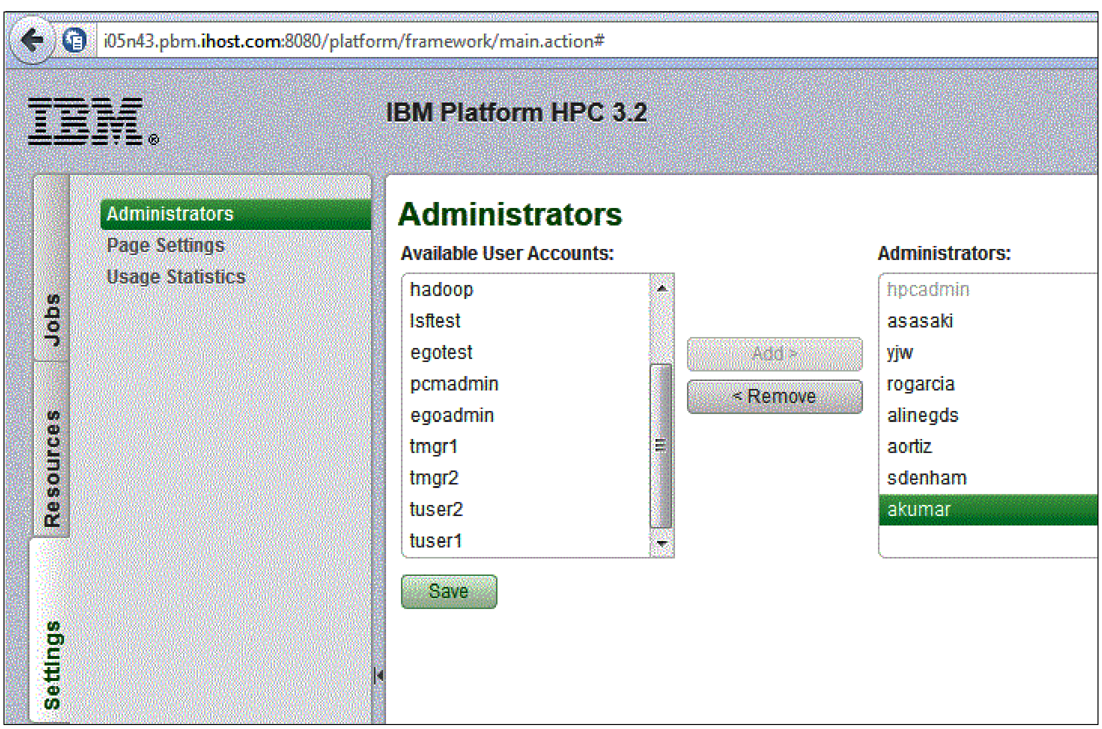


Figure 5-1 Making a user an administrator

Figure 5-2 shows the IBM Platform HPC window for a user who is not an administrator. The only tab a that this user can see is the **Jobs** tab.

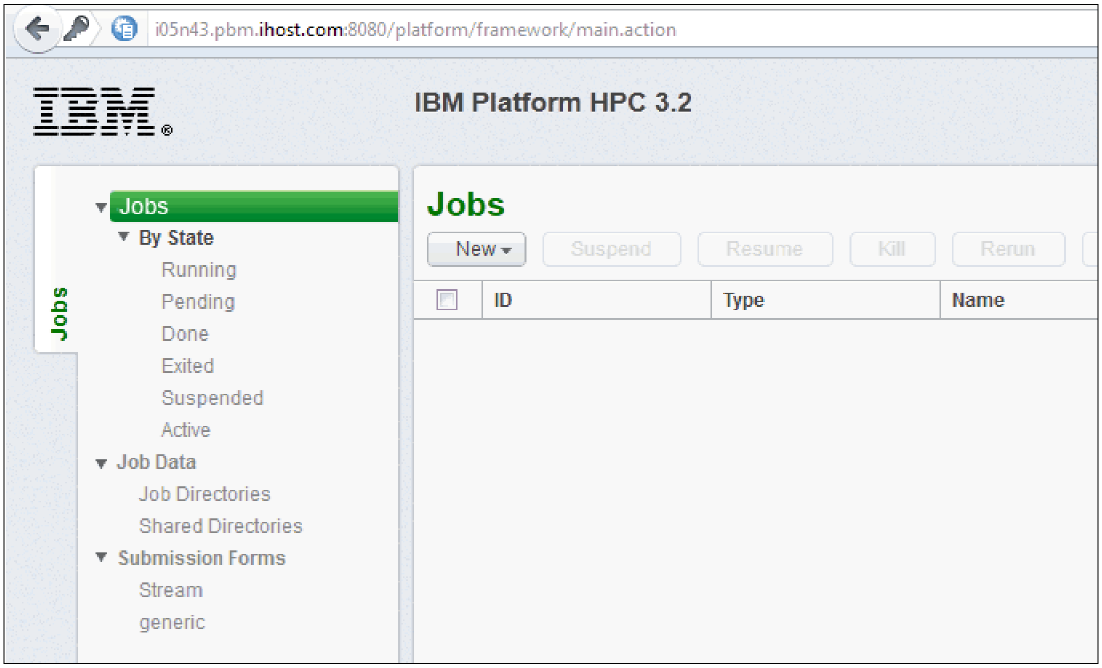


Figure 5-2 Nonadministrator user view of the web portal

To plan your firewall settings, see the instructions and default security settings in *Administering IBM Platform HPC*, SC22-5379, at the following address:

http://publibfp.dhe.ibm.com/epubs/pdf/c2253790.pdf

This same manual is in the `platform_hpc_admin.pdf` file that ships with IBM Platform HPC.

5.2 IBM Platform LSF

The IBM Platform LSF software is leading enterprise-class software that distributes work across existing heterogeneous IT resources. It creates a shared, scalable, and fault-tolerant infrastructure that delivers faster, more reliable workload performance while reducing cost. IBM Platform LSF balances workloads and allocates resources, while providing access to those resources.

IBM Platform LSF provides a resource management framework. By using your job requirements, it finds the best resources in the cluster to run the job and monitors its progress. Jobs always run according to host load and site policies.

Figure 5-3 describes the security model of IBM Platform LSF. For more information about this model, see the *IBM Platform LSF security considerations* manual, which is in the `lsf_security.pdf` file that ships with the product.

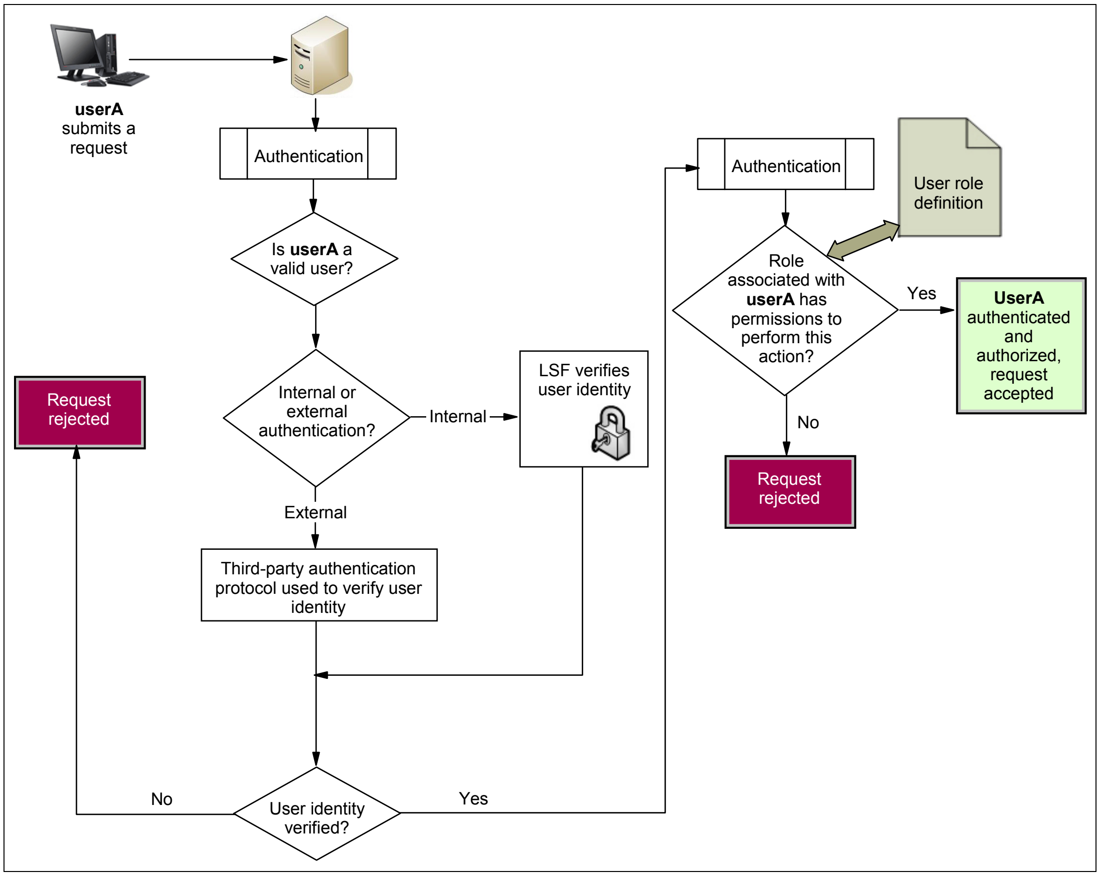


Figure 5-3 Security model for IBM Platform LSF

By default, the security model for IBM Platform LSF tracks user accounts internally. A user account that is defined in IBM Platform LSF includes a password to provide authentication and an assigned role, such as administrator, to provide authorization. You can configure IBM Platform LSF to use external or third-party security mechanisms, such as Kerberos, LDAP, and Active Directory.

By using the **busers** command, you can see detailed information for all users as shown in Example 5-1.

Example 5-1 Detailed information for all users

```
[root@i05n45 ~]# busers  all
USER/GROUP          JL/P    MAX  NJOBS    PEND     RUN  SSUSP  USUSP     RSV
ITS                    -      -      0       0       0      0      0       0
STG                    -      -      0       0       0      0      0       0
alinegds               -      -      0       0       0      0      0       0
default                -      -      -       -       -      -      -       -
lsfadmin               -      -      2       0       2      0      0       0
lsftest                -      -      0       0       0      0      0       0
mayesp                 -      -      0       0       0      0      0       0
osuser                 -      -      0       0       0      0      0       0
parkera                -      -      0       0       0      0      0       0
```

```
rogarcia          -   -   0   0   0   0   0   0
sdenham           -   -   0   0   0   0   0   0
userGroupB8       -   -   0   0   0   0   0   0
userGroupB8@      -   -   0   0   0   0   0   0
yjw               -   -   0   0   0   0   0   0
```

All IBM Platform LSF jobs run under the user ID of the user who submitted the job, unless you are using account mapping. IBM Platform LSF enforces restrictions on job access based on the user ID of the user who is running a command and the user ID associated with the submitted job.

All IBM Platform LSF users can view basic information about all jobs, including jobs that are submitted by other users. However, users can view only detailed information about or modify jobs that are submitted by their own user IDs. Only administrators can modify jobs that are submitted by other users.

Table 5-1 summarizes the job commands and their behavior with different users.

Table 5-1 Job command summary

Command	Run by	Comment
bjobs	All	
bhist	All	
bhosts	All	
bqueues	All	
bpeek	All	User can run this command for their own jobs, but administrator can run it for all jobs.
bbot	Administrator	
btop	Administrator	
bchkpnt	All	User can run this command for their own jobs, but administrator can run it for all jobs.
bkill	All	User can run this command for their own jobs, but administrator can run it for all jobs.
bmod	All	User can run this command for their own jobs, but administrator can run it for all jobs.
brestart	All	User can run this command for their own jobs, but administrator can run it for all jobs.
bresume	All	User can run this command for their own jobs, but administrator can run it for all jobs.
bstop	All	User can run this command for their own jobs, but administrator can run it for all jobs.

5.3 IBM Platform Symphony

IBM Platform Symphony provides an application framework that you can use to run distributed or parallel applications in a scaled grid environment. IBM Platform Symphony offers several security features. By using the authentication and user administration options, system administrators can specify user privileges for specific tasks and resources. A basic security model has the following key elements:

- User registry integration
- Integrity
- Confidentiality
- Secure sockets layer (SSL)-supported secure data transmission

IBM Platform Symphony is controlled by two interdependent processes: authentication and authorization. *Authentication* is used to determine the identity of the user and verify and validate that identity. Authorization checks the permissions of the authenticated user and controls access to functions based on the roles that are assigned to the user.

Figure 5-4 shows the security model. For more information about this model, see the *IBM Platform Symphony: An overview* manual, which is in the `foundations_sym.pdf` file that ships with the product.

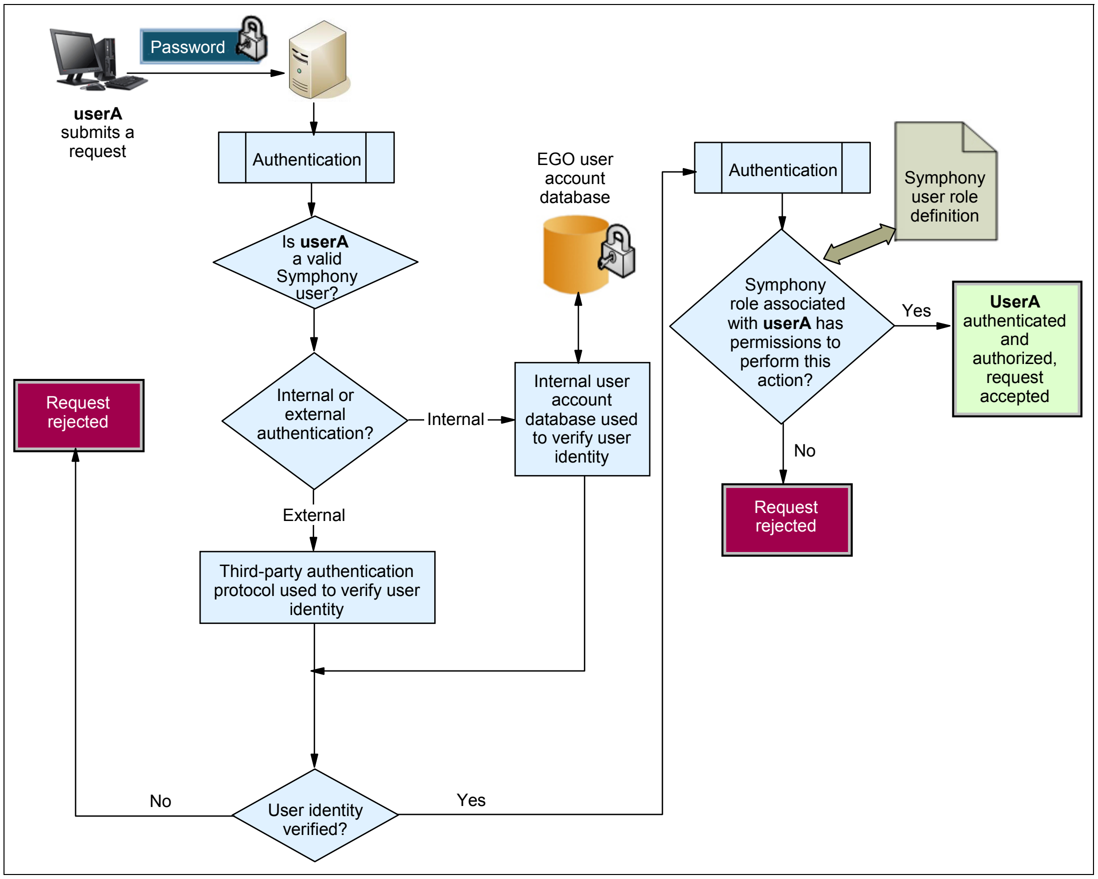


Figure 5-4 Security model for IBM Platform Symphony

5.3.1 Understanding users in symphony

IBM Platform Symphony has two sets of users: one set that performs all administrative operations and one set of regular users who are consumers. Table 5-2 summarizes these users.

Table 5-2 Overview of IBM Platform Symphony users

User	Description	Actions	Registry to exist in or to be configured to
Installation user		Installation	Operating system
Cluster administrator	This user can be same as the installation user or can be another user during installation process (default is egoadmin).	Cluster administration commands	Operating system

User	Description	Actions	Registry to exist in or to be configured to
Internal user	In the default configuration, Admin and Guest are defined.	Workload management or configuration	Internal Symphony database, operating system, or LDAP

Table 5-3 lists the cluster and workload configuration commands. For a description of each command, see the *IBM Platform Symphony Reference* manual that ships with the product.

Table 5-3 Configuration commands

Commands category		
Cluster management, configuration control commands	**Workload management commands**	**Development environment commands**
egosh **egostartup** **egoshutdown** **pversions** **rfa** **rsdeploy** **soamlog** **egoconfig** **egosetrc** **egoremoverc** **egosetsudoers**	**soamcontrol** **soamlogon** **soamdeploy** **soamlogoff** **soammod** **soamreg** **soamunreg** **soamswitch** **soamview** **symexec** **symping**	**soamcontrol** **soamdeploy** **soamlog** **soammod** **soamunreg** **soamreg** **soamshutdown** **soamstartup** **soamswitch** **soamview** **symexec** **symping**

5.3.2 Configuring the external user registry

You can configure internal users of IBM Platform Symphony to come from LDAP or from the operating system user registry. The following scenario has the following assumptions:

- You have a working IBM Platform Symphony V5.2 cluster setup.
- You are logged in as cluster administrator `egoadmin`.
- **pdsh** is installed on a master node and is configured for all compute nodes.
- ego is stopped by using the **egoshutdown** command.

The steps in this section apply to the following platforms:

- Linux-2.6-glibc-2.3-x86 (32-bit or 64-bit)
- Linux-2.4-glibc-2.3-x86 (32-bit)
- Linux-2.4-glibc-2.2-x86 (32-bit or 64-bit)
- Linux-2.6-glibc-2.3-ia64

To configure a user with the operating system:

1. Copy the `sec_ego_ext_pam.so` file to the `EGO_LIBDIR` directory on all management hosts (Example 5-2).

   Example 5-2 Copying the sec_ego_ext_pam.so file

   ```
   cp $EGO_TOP/gui/ego/1.2.6/lib/linux2.6-glibc2.3-x86_64/sec_ego_ext_pam.so
   $EGO_LIBDIR
   ```

2. Copy the `sec_ego_ext_co.so` file (client-only plug-in) to the `EGO_LIBDIR` directory on all compute hosts (Example 5-3).

Example 5-3 Copying the client-only plug-in

```
pdcp $EGO_TOP/1.2.6/linux2.6-glibc2.3-x86_64/lib/sec_ego_ext_co.so $EGO_LIBDIR
```

3. Create a plug-in configuration and save it as `pamauth.conf` in the `$EGO_CONFDIR` directory on all management nodes (Example 5-4).

Example 5-4 Copying the pamauth.conf file to all management nodes

```
      hosts'
# EGO PAM plugin configuration file
# Mandatory parameters
# PAM_ADMIN=<pam-account-name>
#          PAM account that should be mapped to EGO 'Admin' user.
#          For Active Directory use the format <domain-name>\<username>
PAM_ADMIN=egoadmin
# PAM_SERVICE=<pam-service-name>
#          PAM Service file (under /etc/pam.d) defining the
#          PAM policy to be used for EGO.
#          Default is "sshd"
PAM_SERVICE=sshd
```

4. Copy the **egostashpass-pam** utility to `$EGO_BINDIR` on the master host (Example 5-5).

Example 5-5 Copying the egostashpass-pam utility

```
copy 'egotashpass-pam' /$EGO_BINDIR
```

5. Change the password of the administrator. The password must be the same password of the PAM_ADMIN account that is in the `pamauth.conf` file.
6. Edit the `ego.conf` file on the management nodes, and modify the value of the **EGO_SEC_PLUGIN** and **EGO_SEC_CONF** parameters (Example 5-6). The **EGO_SEC_CONF** parameter can have the following values:
   - plugin-configuration-directory
   - created-ttl
   - plugin-log-level
   - plugin-log-directory

Example 5-6 Editing the ego.conf file on the management nodes

```
EGO_SEC_PLUGIN=sec_ego_ext_pam
EGO_SEC_CONF=/opt/ibm/sym52/kernel/conf,0,INFO,/opt/ibm/sym52//kernel/log
```

7. Edit the `ego.conf` file on the compute nodes, and modify the value of the **EGO_SEC_PLUGIN** parameter (Example 5-7).

Example 5-7 Editing the ego.conf file on the compute nodes

```
EGO_SEC_PLUGIN=sec_ego_ext_co
```

The **EGO_SEC_PLUGIN** parameter is optional on compute hosts and client machines. Specify this parameter if log messages are required from the client side plug-in. The format is same as specified in step 6. The client-side messages are logged to `ego_ext_plugin_client.log` in the `plugin-log-directory`.

8. Start the ego cluster (Example 5-8).

Example 5-8 Starting the ego cluster

```
egosh ego start all
```

9. Map the Pluggable Authentication Module (PAM) account to the EGO account (Example 5-9).

Example 5-9 Mapping the PAM account to the EGO account

```
egosh user logon -u Admin -x Admin
egosh user add
```

Users can log in to the EGO account by using the same user name and password that they use to log in to a Linux machine in your enterprise. The EGO cluster administrator can log in to EGO by using the user name `admin` and the password of the mapping account that is specified in the `pamauth.conf` file.

Considerations

When you configure the external user registry, keep in mind the following considerations:

- The operating systems of all management hosts must use the same authentication mechanism.
- This version of the plug-in does not support encryption and data integrity between authenticated EGO clients and VEMKD. IBM intends to provide this support in future releases of the plug-in.
- The current version of the PAM plug-in does not support the ability to write to or modify users and groups. Limited support is available in the glibc NSS interface.
- An issue with the LikeWise active directory integration software can make the `getgrent()` and `getpwent()` `libc` calls set `errno` to `ENOENT` (no such file nor directory exists) on successful exit. This problem is ignored by the PAM plug-in. For more information about LikeWise, go to:

  http://likewisesoftware.com
- If the NIS or the UNIX password and group files are used, the PAM plug-in ignores the default group concept. All user group mappings must be explicitly listed in the groups file. Implicit membership through the default group in the `passwd` file is not supported.

A

XML installation file for IBM InfoSphere BigInsights

During the testing and development of the content for this IBM Redbooks publication, the XML installation file shown in Example A-1 was used. This file was generated for the configuration of the 10 data nodes and the 1 management node from the hardware described in 2.2, “Environment” on page 21.

Example A-1 Exported Installation XML file “as is” through the web-based GUI


```
<?xml version="1.0" encoding="UTF-8"?>
<cluster-configuration>
    <operation>install</operation>
    <vendor>ibm</vendor>
    <general>
        <biginsights-install-directory>/opt/ibm/bi</biginsights-install-directory>

<biginsights-data-log-directory>/var/log/bi</biginsights-data-log-directory>
        <directory-prefix>/opt/ibm/bi</directory-prefix>
        <overwrite>false</overwrite>
        <file-system>hdfs</file-system>
        <shared-directory/>
    </general>
    <ssh>
        <configure>root_configured</configure>
        <auth-method/>
        <password>{xor}</password>
        <biadmin-userid>egoadmin</biadmin-userid>
        <biadmin-groupid>egoadmin</biadmin-groupid>
        <biadmin-password>{xor}OjgwPjsyNjE=</biadmin-password>
        <current-user-password>{xor}</current-user-password>
    </ssh>
    <Console>
        <node>i05i04</node>
        <https>false</https>
        <management-console-port>8080</management-console-port>
```



```
</Console>
<Jaql-server>
    <configure>true</configure>
    <node>i05i04</node>
    <jaql-server-port>8200</jaql-server-port>
</Jaql-server>
<Jaql>
    <configure>true</configure>
    <log-directory>/var/log/bi/jaql/logs</log-directory>
</Jaql>
<Derby>
    <configure>true</configure>
    <node>i05i04</node>
    <backup-node/>
    <port>1528</port>
</Derby>
<hadoop>
    <general>
        <cache-directory>/data/mapred/local</cache-directory>
        <log-directory>/var/log/bi/hadoop</log-directory>
        <mapred-system-directory>/data/mapred/system</mapred-system-directory>
    </general>
    <hdfs>
        <configure>true</configure>
    </hdfs>
    <namenode>
        <node>i05i04</node>
        <namenode-port>9000</namenode-port>
        <namenode-http-port>50070</namenode-http-port>
        <name-directory>/data/hdfs/name</name-directory>
    </namenode>
    <jobtracker>
        <node>i05i04</node>
        <jobtracker-port>9001</jobtracker-port>
        <jobtracker-http-port>50030</jobtracker-http-port>
    </jobtracker>
    <secondarynamenode>
        <node>i05i04</node>
        <secondarynamenode-http-port>50090</secondarynamenode-http-port>
        <data-directory-2nn>/data/hdfs/namesecondary</data-directory-2nn>
    </secondarynamenode>
    <datanode>
        <selection-type>All</selection-type>
        <nodes/>
        <datanode-port>50010</datanode-port>
        <datanode-ipc-port>50020</datanode-ipc-port>
        <datanode-http-port>50075</datanode-http-port>
        <tasktracker-http-port>50060</tasktracker-http-port>
        <data-directory>/data/hdfs</data-directory>
    </datanode>
</hadoop>
<Avro>
    <configure>false</configure>
</Avro>
<Hive>
```

```
        <configure>true</configure>
        <hwi-node>i05i04</hwi-node>
        <query-directory>/var/log/bi/hive/query</query-directory>
        <log-directory>/var/log/bi/hive/logs</log-directory>
        <hwi-port>9999</hwi-port>
        <server-port>10000</server-port>
    </Hive>
    <Lucene>
        <configure>true</configure>
    </Lucene>
    <Pig>
        <configure>true</configure>
        <log-directory>/var/log/bi/pig/logs</log-directory>
    </Pig>
    <Oozie>
        <configure>true</configure>
        <node>i05i04</node>
        <oozie-port>8280</oozie-port>
    </Oozie>
    <Zookeeper>
        <configure>true</configure>
        <nodes>i05i04</nodes>
        <data-directory>/var/log/bi/zookeeper/data</data-directory>
        <log-directory>/var/log/bi/zookeeper/logs</log-directory>
        <client-port>2181</client-port>
        <time-interval>2000</time-interval>
        <init-limit>5</init-limit>
        <sync-limit>2</sync-limit>
    </Zookeeper>
    <HBase>
        <configure>true</configure>
        <zookeeper-mode>shared</zookeeper-mode>
        <master-nodes>i05i04</master-nodes>
        <install-mode>fully</install-mode>
        <region-nodes>i05i04</region-nodes>
        <root-directory>/opt/ibm/bi/hbase</root-directory>
        <log-directory>/var/log/bi/hbase/logs</log-directory>
        <master-port>60000</master-port>
        <master-ui-port>60010</master-ui-port>
        <regionserver-port>60020</regionserver-port>
        <regionserver-ui-port>60030</regionserver-ui-port>
    </HBase>
    <Flume>
        <configure>true</configure>
        <zookeeper-mode>shared</zookeeper-mode>
        <master-nodes>i05i04</master-nodes>
        <nodes>i05i04</nodes>
        <pid-directory>/var/log/bi/flume/pids</pid-directory>
        <log-directory>/var/log/bi/flume/logs</log-directory>
    </Flume>
    <node-list>
        <node>
            <name-or-ip>i05i04</name-or-ip>
            <password>{xor}</password>
            <rack/>
```

```
        <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
        <gpfs-node-designation/>
        <gpfs-admin-node/>
        <gpfs-rawdisk-list/>
    </node>
    <node-range>
        <ip-prefix>129.40.128</ip-prefix>
        <start>6</start>
        <end>10</end>
        <password>{xor}</password>
        <rack/>
        <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
        <gpfs-rawdisk-list/>
    </node-range>
    <node-range>
        <ip-prefix>129.40.128</ip-prefix>
        <start>14</start>
        <end>18</end>
        <password>{xor}</password>
        <rack/>
        <hdfs-data-directory>/data/hdfs</hdfs-data-directory>
        <gpfs-rawdisk-list/>
    </node-range>
</node-list>
<GPFS>
    <install>false</install>
    <cluster>
        <cluster-name>bigpfs</cluster-name>
        <primary-configuration-server/>
        <secondary-configuration-server/>
    </cluster>
    <file-system>
        <default-metadata-replication>1</default-metadata-replication>
        <max-metadata-replication>3</max-metadata-replication>
        <default-data-replication>1</default-data-replication>
        <max-data-replication>3</max-data-replication>
        <block-allocation>hcluster</block-allocation>
        <block-group-factor>128</block-group-factor>
        <write-affinity-depth>1</write-affinity-depth>
        <striping-method>failureGroupRoundRobin</striping-method>
        <failuregroup-maskbits>8</failuregroup-maskbits>
        <estimated-cluster-size>32</estimated-cluster-size>
        <mount-point/>
        <tmp-fileset>tmp</tmp-fileset>
        <log-fileset>log</log-fileset>
    </file-system>
</GPFS>
<enterprise>
    <Orchestrator>
        <configure>true</configure>
        <node>i05i04</node>
        <port>8888</port>
    </Orchestrator>
</enterprise>
<TaskController>
```

```
        <directory>/var/bi-task-controller-conf</directory>
        <groups>*</groups>
        <hosts>*</hosts>
    </TaskController>
</cluster-configuration>
```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *IBM Platform Computing Solutions*, SG24-8073
- *Implementing IBM InfoSphere BigInsights on System x*, SG24-8077

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- *Administering IBM Platform LSF*, SC22-5346

  http://publibfp.dhe.ibm.com/epubs/pdf/c2253460.pdf
- *Administering IBM Platform HPC*, SC22-5379

  http://publibfp.dhe.ibm.com/epubs/pdf/c2253790.pdf
- *Running Jobs with IBM Platform LSF*, SC27-5307

  http://publibfp.dhe.ibm.com/epubs/pdf/c2753070.pdf

Online resources

These websites are also relevant as further information sources:

- IBM Platform Computing

  http://www.ibm.com/systems/technicalcomputing/platformcomputing/index.html
- IBM Technical Computing

  http://www.ibm.com/systems/technicalcomputing/

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM Redbooks

IBM Platform Computing Integration Solutions

(0.2”spine)
0.17”<->0.473”
90<->249 pages

IBM Platform Computing Integration Solutions

Learn about solutions that integrate IBM Platform Symphony and IBM InfoSphere BigInsights for analytics

See how the solutions use IBM System x and IBM GPFS

Understand details about IBM Platform LSF implementation

This IBM Redbooks publication describes the integration of IBM Platform Symphony with IBM BigInsights. It includes IBM Platform LSF implementation scenarios that use IBM System x technologies. This IBM Redbooks publication is written for consultants, technical support staff, IT architects, and IT specialists who are responsible for providing solutions and support for IBM Platform Computing solutions.

This book explains how the IBM Platform Computing solutions and the IBM System x platform can help to solve customer challenges and to maximize systems throughput, capacity, and management. It examines the tools, utilities, documentation, and other resources that are available to help technical teams provide solutions and support for IBM Platform Computing solutions in a System x environment.

In addition, this book includes a well-defined and documented deployment model within a System x environment. It provides a planned foundation for provisioning and building large scale parallel high-performance computing (HPC) applications, cluster management, analytics workloads, and grid applications.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-8081-00 ISBN 0738437883