Redbooks
ibm.com/redbooks

IBM Software Defined Infrastructure for Big Data Analytics Workloads

Dino Quintero

Daniel de Souza Casali

Marcelo Correia Lima

Istvan Gabor Szabo

Maciej Olejniczak

Tiago Rodrigues de Mello

Nilton Carlos dos Santos

Analytics

Power Systems

IBM

Redbooks

International Technical Support Organization

IBM Software Defined Infrastructure for Big Data Analytics Workloads

June 2015

SG24-8265-00

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

First Edition (June 2015)

This edition applies to IBM Platform Symphony v7.1 with Application Service Controller, IBM Platform LSF 9.1.3, IBM Spectrum Scale (formerly GPFS) 4.1, IBM Platform Application Center 9.1.3, IBM Platform HPC 4.2, IBM Platform Cluster Manager Advanced Edition 4.2, IBM Platform Data Manager for LSF.

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

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
Algo®
Algo Market®
Algo Risk®
Algorithmics®
BigInsights™
Bluemix™
Cognos®
DataStage®
DB2®
developerWorks®
GPFS™
IBM®
IBM Spectrum™
IBM Watson™
InfoSphere®
Insight™
Insights™
Jazz™
LSF®
OpenPages®
Power Systems™
POWER6®
POWER7®
POWER8™
PowerLinux™
PowerPC®
PureSystems®
QRadar®
Redbooks®
Redbooks (logo) ®
Smarter Planet®
SPSS®
Symphony®
Tealeaf®
Tivoli®
WebSphere®

The following terms are trademarks of other companies:

SoftLayer, and SoftLayer device are trademarks or registered trademarks of SoftLayer, Inc., an IBM Company.

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

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

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

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

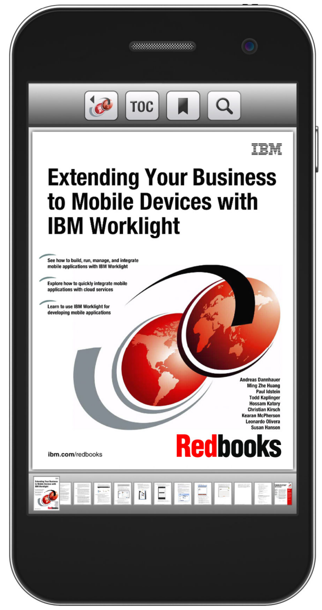
Other company, product, or service names may be trademarks or service marks of others.

Find and read thousands of IBM Redbooks publications

- Search, bookmark, save and organize favorites
- Get up-to-the-minute Redbooks news and announcements
- Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App

iOS **Download Now** Android



Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!

ibm.com/Redbooks
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redbooks® publication documents how IBM Platform Computing, with its IBM Platform Symphony® MapReduce framework, IBM Spectrum™ Scale (based Upon IBM GPFS™), IBM Platform LSF®, the Advanced Service Controller for Platform Symphony are work together as an infrastructure to manage not just Hadoop-related offerings, but many popular industry offeringsm such as Apach Spark, Storm, MongoDB, Cassandra, and so on.

It describes the different ways to run Hadoop in a big data environment, and demonstrates how IBM Platform Computing solutions, such as Platform Symphony and Platform LSF with its MapReduce Accelerator, can help performance and agility to run Hadoop on distributed workload managers offered by IBM. This information is for technical professionals (consultants, technical support staff, IT architects, and IT specialists) who are responsible for delivering cost-effective cloud services and big data solutions on IBM Power Systems™ to help uncover insights among client’s data so they can optimize product development and business results.

Authors

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

Dino Quintero is a Complex Solutions Project Leader and an IBM Senior Certified IT Specialist with the ITSO in Poughkeepsie, NY. His areas of knowledge include enterprise continuous availability, enterprise systems management, system virtualization, technical computing, and clustering solutions. He is 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.

Daniel de Souza Casali is an IBM Cross Systems Senior Certified and has been working at IBM for 11 years. Daniel works for Systems and Technology Group in Latin America as a Software Defined infrastructure IT Specialist. Daniel holds an Engineering degree in Physics from the Federal University of São Carlos (UFSCar). His areas of expertise include UNIX, SAN networks, IBM Disk Subsystems, clustering cloud, and analytics solutions.

Marcelo Correia Lima is a Business Intelligence Architect at IBM. He has 17 years of experience in development and integration of Enterprise Applications. His current area of expertise is Business Analytics Optimization (BAO) Solutions. He has been planning and managing Solutions lifecycle implementation, involving Multidimensional Modeling, IBM InfoSphere® Data Architect, IBM InfoSphere DataStage®, IBM Cognos® Business Intelligence, and IBM DB2®. In addition, Marcelo has added Hadoop, big data, IBM InfoSphere BigInsights™, cloud computing, and IBM Platform Computing to his background. Marcelo holds Level 1 IT Specialist Certification Experienced Level in the Services and Business Information Management specializations (by The Actualizing IT Solutions Capability Center - IBM Integrated Technology Delivery Headquarters). Before working as a Business Intelligence Architect, he was involved in the design and implementation of IBM WebSphere® and Java Enterprise Edition Applications for IBM Data Preparation/Data Services.

Istvan Gabor Szabo is an Infrastructure Architect and Linux Subject Matter Expert at IBM Hungary (IBM DCCE SFV). He joined IBM in 2010 after receiving his bachelor degree in Engineering Information Technology from the University of Obudai - John von Neumann

Faculty of Informatics. Most of the time, he works on projects as a Linux technical lead. His areas of expertise are configuring and troubleshooting complex environments, and building automation methodologies for server builds. In his role as an Infrastructure Architect, he works on the IBM Standard Software Installer (ISSI) environment, where he designs new environments based on customer requirements.

Maciej Olejniczak is a Cross-functional Software Support Team Leader in a collaborative environment. He works internationally with external and internal clients, IBM Business Partners, services, labs, and research teams. He is a dedicated account advocate for large customers in Poland. Maciej is an IBM Certified Expert in Actualizing IT Solutions: Software Enablement. He achieved a master level in implementing all activities that transform information technology from a vision to an actual working solution. Maciej is an Open Group Master Certified IT Specialist.

Tiago Rodrigues de Mello is a Staff Software Engineer in Brazil with more than 10 years of experience. Tiago's ares of expertise include Linux system administration, software development, and cloud computing. He is an OpenStack developer and a Continuous Integration engineer at the IBM Linux Technology Center. Tiago holds a Bachelor in Computer Science degree from the Federal University of Sao Carlos, Brazil.

Nilton Carlos dos Santos is an IT Architect and a Certified IT Specialist and has been with IBM since 2007 and has 18 years experience in the IT industry. Before joining IBM, he worked in several different areas of technology, including Linux and UNIX administration, database management, development in many different languages, and network administration. Nilton Carlos also has deep expertise in messaging, automation, monitoring, and reporting system tools. He enjoys working with open source software.

Thanks to the following people for their contributions to this project:

Richard Conway, David Bennin
International Technical Support Organization, Austin Center

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:

ibm.com/redbooks

- Send your comments by email:

  redbooks@us.ibm.com

- Mail your comments:

  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

Introduction to big data

As our planet becomes more integrated, the rate of data growth is increasing exponentially. This data explosion is rendering commonly accepted practices of data management inadequate. This has resulted in a new wave of business challenges related to data management and analytics. Everywhere, you hear and read many people mentioning the term *big data*.

This chapter provides a foundational understanding of big data: What it is, and why you need to care about it. It also describes IBM technology to handle these data management challenges.

The following topics are covered in this chapter:

1.1 Evolution and characteristics of big data

If your profession is heavily based in the realm of information management, there is a good chance that you heard the term *big data* at least once over the past week or so. It has become increasingly popular to incorporate "big data" in data management, cloud computing, and application development discussions.

In a similar way, it was previously popular to bring the advent of service-oriented architecture (SOA) and Web 2.0, just as examples. "Big data" is a trendy talking point at many companies, but few people understand exactly what it means.

Rather than volunteering an arbitrary definition of the term, we believe that a better approach is to explore the evolution of data, along with enterprise data management systems. This approach ultimately arrives at a clear understanding of what big data is and why we need to care about it.

The IBM Smarter Planet® initiative was launched during a speech to the Council of Foreign Relations in New York in 2008. Smarter Planet focuses on development of technologies that are advancing everyday experiences. A large part of developing such technology is dependent on the collection and analysis of data from as many sources as possible. This process is difficult, because the number and variety of sources continues to grow. The planet is exponentially more instrumented, intelligent, and integrated, and it will continue to expand, with better and faster capabilities. The World Wide Web is truly living up to its name. Through its continued expansion, cloud computing and the web are driving our ability to generate and have access to virtually unlimited amounts of data.

The statistics that are presented in Figure 1-1 confirm the validity of the world becoming exponentially more instrumented.

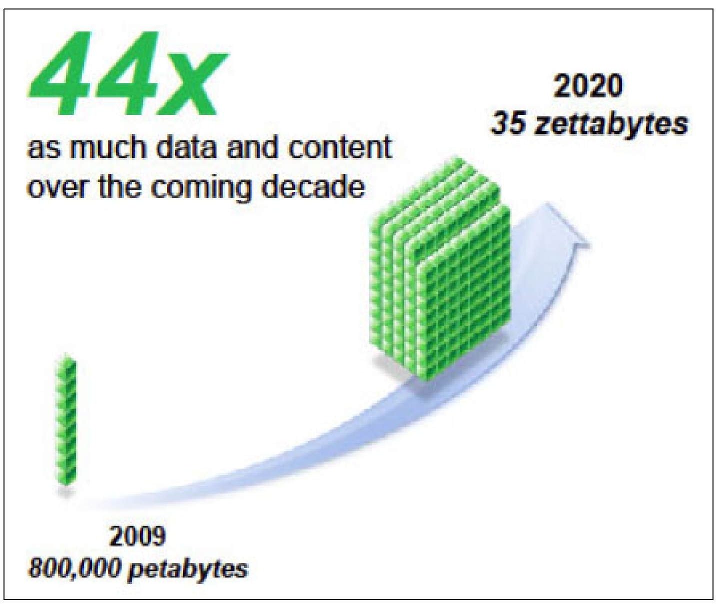


Figure 1-1 Predicted worldwide data growth

There was an earlier point in history when only home computers and web-hosting servers were connected to the web. If you had a connection to the web and ventured into the world of chat rooms, you were able to communicate by instant messaging with someone in another part of the world. Hard disk drives were 256 MB, CD players were top-shelf technology, and cell phones were as large as lunch boxes. We are far from those days.

Today, the chances are that you are now able to download this book from your notebook computer or tablet while you are sending email, sending instant messages back and forth with a friend overseas, or texting your significant other, all while enjoying your favorite clothing retailer's Facebook page. The point is, you now generate more data in 30 seconds than you would have in 24 hours just 10 years ago.

We are now at the crux of a data explosion with significantly more items continuously generating data. Where, exactly, is this data coming from? Figure 1-2 shows a few examples of the items and sources of this data explosion.

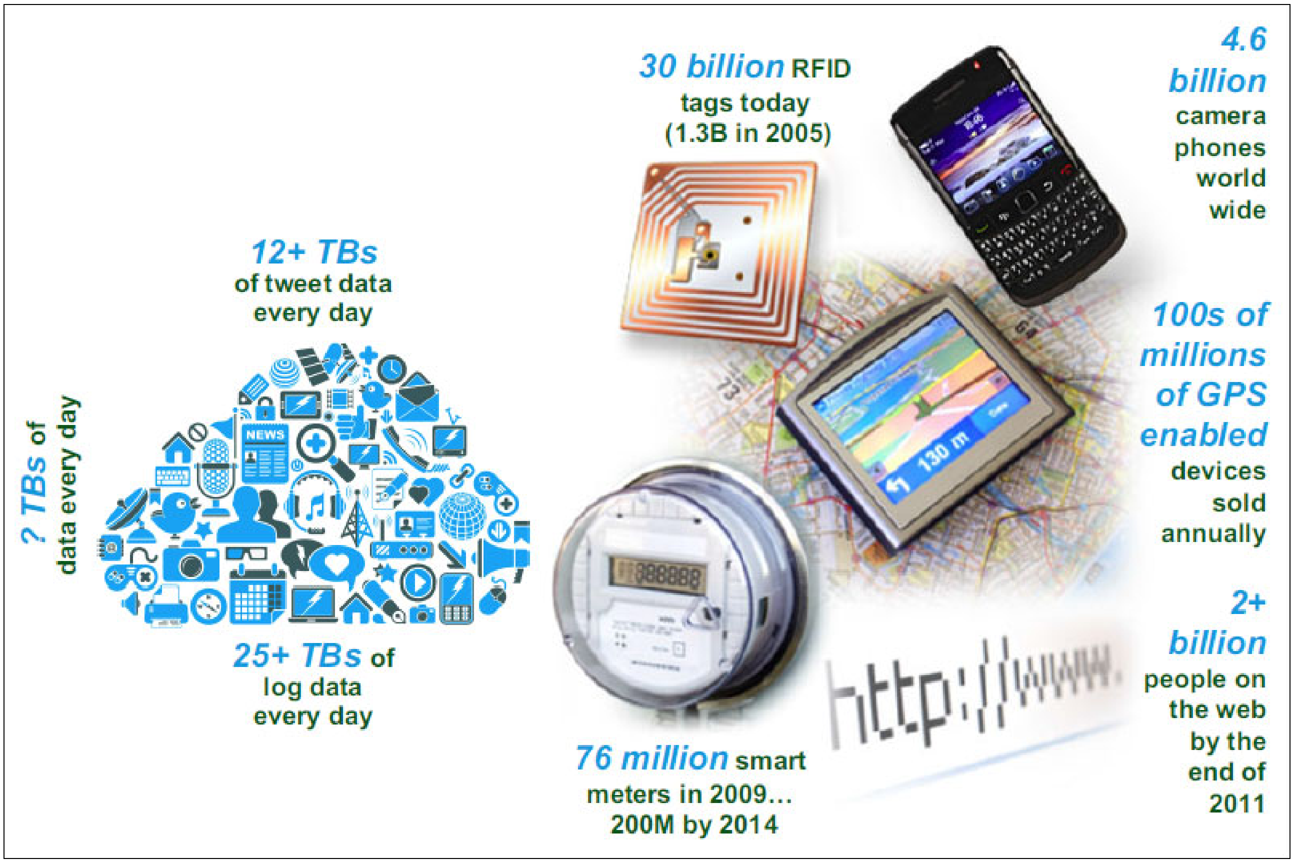


Figure 1-2 Big data explosion

Every day, we create 2.5 quintillion bytes of data — so much that 90% of the data in the world today has been created in the last two years. This data comes from everywhere: Sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, medical devices, and cell phone GPS signals, to name a few.

Web-based applications, including social media sites, now exceed standard e-commerce websites in terms of user traffic. Facebook, Twitter, YouTube, WhatsApp, produce unlimited amounts of data and logs on a daily basis. Almost everyone has an email address (often multiple), a smartphone (sometimes multiple also), usually a cache of photo images and videos (whether they choose to share with the social network or not), and can express opinions globally through a personal blog.

In this increasingly instrumented world, there are sensors everywhere constantly generating and transmitting data. In the IT realm, machine data is being generated by servers and switches, and they are always generating log data (commonly known as *data exhaust*). These software applications are all operational 24x7 and continuously generating more data.

Despite establishing that there is significantly more data generated today than there was in the past, big data is not just about the sheer volume of data that is being created. With a myriad of unstructured sources creating this data, a greater variety of data is now available.

Each source produces this data at different rates or what we call *velocity*. In addition, we still must decipher the *veracity* of this new information, as we do with structured data. Here is where the information management industry had its awakening moment: Whether our workload is largely transactional or online analytics processing (OLAP) and resource-intensive, both cases operate on structured data. Systems that are designed for the management and analysis of structured data provide valuable insight into the past, but what about all of the newer text-based data that is being created? This data is being generated everywhere we look. There is a larger volume of data, a greater variety of data, and it is being generated at a velocity that traditional methods of data management are no longer capable of efficiently harvesting or analyzing.

Big data applies to all types of data and is characterized by "the four Vs":

Volume Terabytes, petabytes, or more

Velocity Data in motion or *streaming* data

Variety Structured and unstructured data of all types: text, sensor data, audio, video, click streams, log files, and more

Veracity The degree to which data can be trusted

Note: Big data can be subcategorized as *data at rest* and *data in motion*.

Data at rest is a snapshot of the information that is collected and stored, ready to be analyzed for decision making. For example, a video camera can collect a vast amount of terabytes data per day or a popular Twitter page can generate millions of text comments, which can be used for analysis of social media users.

Data in motion is the process of analyzing real-time data dynamically without storing it. For example, a critical fraud detection system or a medical tracking system are good examples of data in motion (streaming).

In general, this book features working scenarios of data at rest, although there is an example of an application called Spark that can be used for streaming data and can be controlled with the Application Service Controller (see Chapter 3, "IBM Platform Symphony with Application Service Controller" on page 27).

Figure 1-3 shows a visual representation of the four Vs.

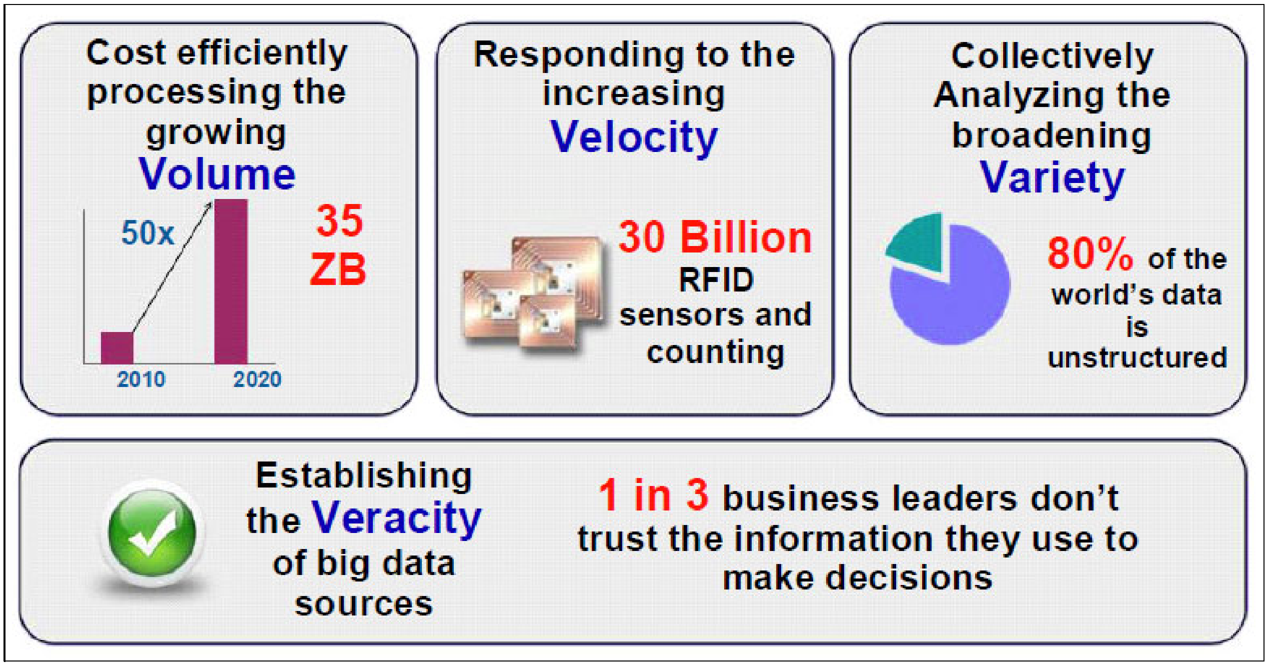


Figure 1-3 Four Vs of big data

1.2 What's in a bite

Terabyte A terabyte is approximately one trillion bytes or 1000 gigabytes. To put that in perspective, a terabyte could hold about 3.6 million 300-kilobyte images or as much as about 300 hours of good quality video. A terabyte could hold 1000 copies of the *Encyclopedia Britannica*. Ten terabytes could hold the printed collection of the Library of Congress.

Petabyte A petabyte is approximately 1000 terabytes or one million gigabytes. 1 petabyte could hold approximately 20 million four-door filing cabinets full of text. It could hold 500 billion pages of standard printed text. It would take about 500 million floppy disks to store the same amount of data.

Exabyte An exabyte is approximately 1000 petabytes. Another way to look at it is that an exabyte is approximately one quintillion bytes or one billion gigabytes. It has been said that 5 exabytes would be equal to all of the words ever spoken by humankind.

Zettabyte A zettabyte is approximately 1000 exabytes. The US is building a facility to house four. Who knows how many other facilities exist or are being built?

Yottabyte A yottabyte is approximately 1000 zettabytes. It would take approximately 11 trillion years to download a yottabyte file from the Internet using a high-speed broadband connection. You can compare it to the World Wide Web, because the entire Internet consists of about one yottabyte.

Brontobyte A brontobyte is approximately 1000 yottabytes. A brontobyte is a 1 followed by 27 zeros.

Geopbyte A geopbyte is about 1000 brontobytes.

There is no limit to the units creation of units that can represent big data.

1.3 Is the demand for a big data solution real?

Consider the following data, and you can answer that question:

- The healthcare industry spends $250 - 300 billion per year on healthcare fraud. In the US alone, this costs $650 million per day.
- One rogue trader at a leading global financial services firm created $2 billion worth of losses, almost bankrupting the company.
- $93 billion in sales are missed each year because retailers do not have the right products in stock to meet customer demand.
- Global subscribers in the telco industry are demanding unique and personalized offerings that match their individual lifestyles.

1.4 What is Hadoop?

Fundamentally, Hadoop consists of two components:

- The Hadoop Distributed File System (HDFS)
- The MapReduce framework

HDFS provides a way to store data, and the MapReduce framework is a way of processing data in a distributed manner. These components were developed by the open source community. They are based on documents that were published by Google in an attempt to overcome the problems in trying to deal with an overwhelming volume of data. Google published papers on their approach to resolve these issues, and then Yahoo started working on an open source equivalent that is named after a child's toy elephant, called Hadoop.

Hadoop consists of many connected computers called *data nodes* that store data on their local file systems and process the data as directed by a central management node. The management nodes consist of the following processes:

- NameNode

  The NameNode process maintains the metadata that relates to where the data is stored on the data nodes. When a job is submitted, this metadata is accessed to locate the data blocks that are needed by the job. The NameNode is also used and the metadata is updated if data is saved. No other processing during a MapReduce is carried out on the NameNode.

  Depending on the version of Hadoop that you are running, the NameNode can be a single point of failure within the Hadoop cluster. The cluster requires manual intervention if it fails.

- Secondary NameNode

  The Secondary NameNode holds a checkpoint of the metadata on the NameNode and "edits" file that logs all changes that are made to the locations of the data. This process is not a redundancy for the NameNode but significantly speeds up the process if the NameNode fails.

- JobTracker

  When a MapReduce job is submitted, the JobTracker decides on which nodes the work is to be carried out. The JobTracker coordinates the distributed processing to ensure that the nodes that are local to the data start to carry out the map and reduce functions. When and where possible, the JobTracker also ensures that work is carried out simultaneously over multiple nodes.

On each data node, you also find a TaskTracker. The role of the TaskTracker is to accept jobs from the JobTracker and create a Java virtual machine (JVM) process to do each task.

Figure 1-4 shows an access plan for a MapReduce framework that was submitted by a client.

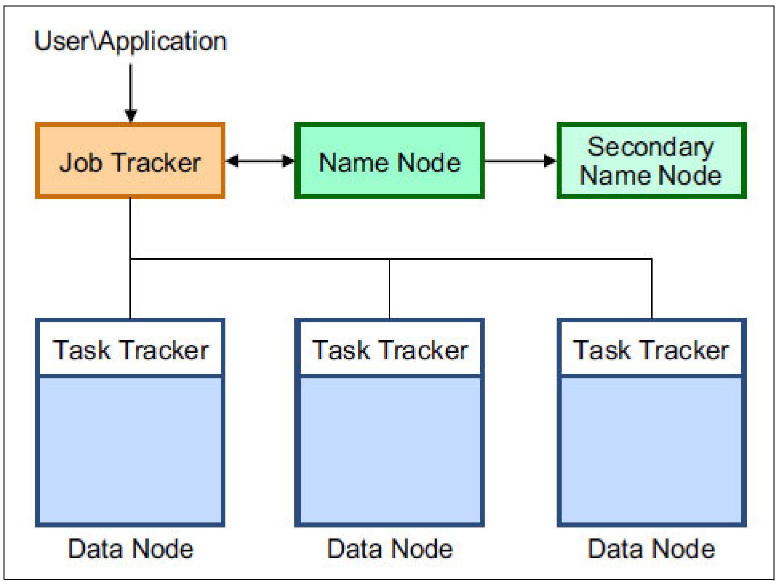


Figure 1-4 Access plan for a MapReduce framemark, submitted by a client

1.5 Hadoop Distributed File System in more detail

HDFS is the file system that is used to store the data in Hadoop. The way that it stores data is special.

When a file is saved in HDFS, it is first broken into blocks, with any remainder data occupying the final block. The size of the block depends on the way that HDFS is configured.

Then, each block is sent to a different data node and written to the hard disk drive. When the data node writes the file to disk, it sends the data to a second data node where the file is written. When this process completes, the second data node sends the data to a third data node. The third node confirms the completion of the write back to the second and then back to the first. The NameNode is then notified and the block write is complete.

After all blocks are written successfully, the result is a file that is broken into blocks, with a copy of each block on three data nodes. The location of all of this data is stored in memory by the NameNode.

Figure 1-5 shows an example of HDFS.

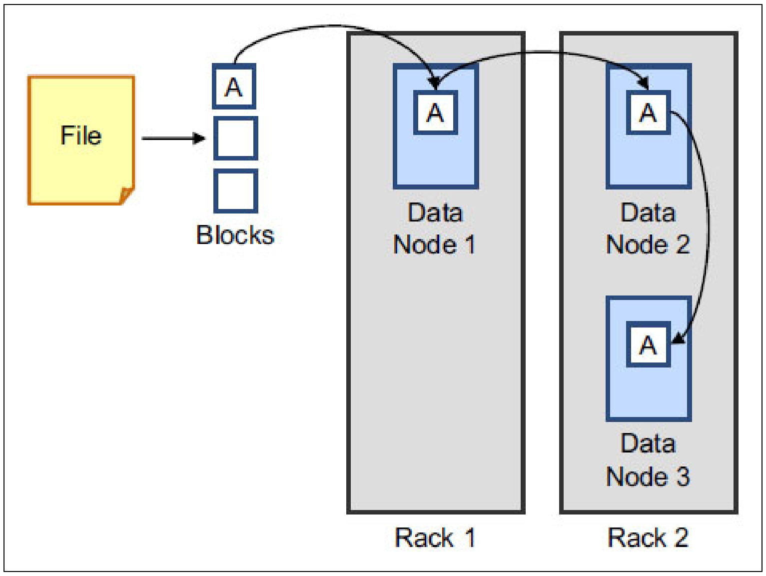


Figure 1-5 A visual example of HDFS

Note: IBM has developed a fast, simple, scalable and complete Software Defined Storage Solution for Data Intensive Enterprise named IBM Spectrum Scale (formerly GPFS). Spectrum Scale is a proven, scalable, high-performance data and file management solution that is being used extensively across multiple industries worldwide. Spectrum Scale provides simplified data management and integrated information lifecycle tools that are capable of managing petabytes of data and billions of files to help reduce the growing cost of managing ever-growing amounts of data.

Spectrum Scale has several advantages over HDFS:

- Active/Active metadata distribution, no single point of failure
- POSIX file system, which is easy to use and manage
- Policy-based data ingestion
- Versatile, multipurpose
- Enterprise class with support for advanced storage features, such as snapshots and replication
- Variable block sizes, suited to multiple types of data and the metadata access pattern

For more information about Spectrum Scale, see Chapter 5, “IBM Spectrum Scale for big data environments” on page 115.

1.6 MapReduce in more detail

As a batch processing architecture, the major value of Hadoop is that it enables ad hoc queries to run against an entire data set and return results within a reasonable time frame. Distributed computing across a multinode cluster is what allows this level of data processing to take place.

MapReduce applications can process vast amounts of data (multiple terabytes), in parallel,on large clusters in a reliable, fault-tolerant manner. MapReduce is a computational paradigm in which an application is divided into self-contained units of work. Each of these units of work can be issued on any node in the cluster.

A MapReduce job splits the input data set into independent chunks that are processed by map tasks in parallel. The framework sorts the map outputs, which are then input to reduce tasks. Job inputs and outputs are stored in the file system. The MapReduce framework and the HDFS are typically on the same set of nodes, which enables the framework to schedule tasks on nodes that contain data.

The MapReduce framework consists of a single primary JobTracker and one secondary TaskTracker per node. The primary node schedules job component tasks, monitors tasks, and reruns failed tasks. The secondary node runs tasks as directed by the primary node.

Minimally, applications specify input and output locations and supply map and reduce functions through implementation of appropriate interfaces or abstract classes.

MapReduce consists of two phases:

- Map
- Reduce

1.6.1 The map phase

The *map phase* is the first part of the data processing sequence within the MapReduce framework.

Map functions serve as worker nodes that can process several smaller snippets of the entire data set. The *MapReduce framework* is responsible for dividing the data set input into smaller chunks and feeding them to a corresponding map function. When you write a map function, there is no need to incorporate logic to enable the function to create multiple maps that can use the distributed computing architecture of Hadoop. These functions are oblivious to both data volume and the cluster in which they are operating. As such, they can be used unchanged for both small and large data sets (which is most common for those using Hadoop).

Based on the data set that you are working with, a programmer must construct a map function to use a series of key/value pairs. After processing the chunk of data that is assigned to it, each map function also generates zero or more output key/value pairs to be passed forward to the next phase of the data processing sequence in Hadoop. The input and output types of the map phase can be (and often are) different from each other.

1.6.2 The reduce phase

As with the map function, developers must also create a *reduce* function. The key/value pairs from map outputs must correspond to the appropriate reducer partition so that the final results are aggregates of the appropriately corresponding data. This process of moving map outputs to the reducers is known as *shuffling*.

When the shuffle process is completed and the reducer copies all of the map task outputs, the reducers can go into what is known as a *merge* process. During this part of the reduce phase, all map outputs can be merged to maintain their sort ordering, which is established during the map phase. This process is done in rounds for performance optimization purposes. When the final merge is complete, the final reduce task of consolidating results for every key within the merged output (and the final result set) is written to the disk on the HDFS.

1.7 The changing nature of distributed computing

Workload management has evolved to support a variety of different types of distributed workloads. The following list includes examples of frequently encountered distributed workload patterns:

- Goal-oriented, SLA-driven scheduling
- Automation of multistep, complex workflow
- Parametric sweeps and session-oriented workloads
- Parallel job scheduling with backfill scheduling optimizations
- Various types of preemptive scheduling policies that ensure service levels are respected
- Service-oriented workloads for various low-latency, scatter-gather scheduling challenges

Although it is possible to use general-purpose grid managers to support many of these workload patterns, those who specialize in workload management appreciate that, to maximize efficiency and resource use, workload manager software needs to be optimized to the specific workload pattern that is being supported. From a scheduling perspective, the series of steps that comprise a MapReduce workflow are simply another type of distributed computing workload. It turns out that there are ample opportunities to optimize performance and efficiency in a fashion that is apparent to higher-level application frameworks that rely on distributed computing services.

1.8 IBM Platform Symphony grid manager

IBM Platform Symphony is an enterprise-class grid manager for running distributed application services on a scalable, shared, heterogeneous grid. It accelerates a wide variety of compute and data-intensive applications, quickly computing results while making optimal use of available infrastructure.

Platform Symphony was originally built to support service-oriented application workloads that are common in the highly competitive financial services industry where traditional batch-oriented schedulers were simply too slow to keep up with real-time demands. For a variety of compute-intensive simulations in areas such as pricing, risk management, fraud-detection, and credit risk modeling, raw performance is essential, but “time-to-result” is critical for market competitiveness.

Firms that can get risk models involving millions of discrete calculations on and off the grid instantly enjoy a distinct competitive advantage over rivals with less-capable grid middleware. Platform Symphony was purposely built to deliver extremely high levels of performance and agility at scale to serve the unique challenges of this market, where seconds count.

Clients typically realize the following benefits when they deploy IBM Platform Symphony as a distributed computing platform:

- Ability to get results faster
- Increased capacity to run more rigorous simulations
- Reduced infrastructure and management costs
- More effective sharing of resources
- Reduced development costs
- Ability to respond quickly to real-time demands

2

Big data, analytics, and risk calculation software portfolio

This chapter introduces and describes the IBM big data analytics and risk calculation offerings which can help customers complement and improve their analytics solutions.

This chapter includes the following sections:

2.1 The impact of big data

Surveys show that 90% of the data generated around the world is unstructured, primarily due to the large amount of data that social networks are creating. For this reason, companies have been looking for technologies to filter useful information for business.

A single phone call can result in hundreds of records that contain the details about it, which must be stored by the telephone service provider to create plans according to the profiles of each if their customers. The world has millions of wireless lines in service, so we can imagine the vast amount of information that this sector generates. If that can be mined and analyzed accurately and in real time, that data can become a valuable asset. Executing this workload is not an easy task, and big data appears to be the solution to meet the requirement of fast data mining.

Telecommunications, manufacturing, retail, utility companies, media, and so on generate much data every minute. Without control, this ocean of data is challenging areas of IT, and businesses need to find ways to use the data to gain competitive advantages and achieve better business results.

The term *big data* was created by market analysts to describe the exponential growth of data that companies need to extract and classify as useful information. The concept began to be disseminated to alert organizations about the need to adopt a strategy to assess mainly unstructured data, which is beyond the control of IT. Actually, the big data challenge is not the high volume of data but to capture the complexity of information in different media formats and access them in real time. Companies are struggling to capture content from social networks when they go live and analyze them along with databases of call centers. This new approach is somewhat different from what has been proposed by business intelligence, which looks over the past, based on historical events that happened, to aid in making decisions.

Tracking big data has the advantage of analyzing events when they are happening and to anticipate measures. For example, you can monitor the consumption pattern of a supermarket's sales in a given region in real time and correlate those with climate to offer ice cream according to the preference of buyers. If the temperature changes, the company can quickly change the campaign to another type of promotion.

In health care areas, it might be possible to determine when a patient has more chances to have a heart attack, cancer, or any other disease. Medical exams, such as x-rays and magnetic resonance imaging (MRIs), or medical and monitoring devices might be a rich source of data to predict health problems before they happen. In this area, IBM offers IBM Watson™ technology, which can help doctors determine and predict patients' health situations before they happen. For more information, see IBM Watson Solutions:

http://www.ibm.com/smarterplanet/us/en/ibmwatson/health/

Another example of this type of analysis can be done by telecommunication operators to create plans tailored to mobile users, looking at the consumption pattern with real-time ratings. The information is there. The challenge is to manage, extract, and analyze the data to provide better customer service and gain business advantages.

Today, companies make decisions based on leaders' predictions, based on their market research. But the best data is stored within the companies, and they have exclusive information that can improve business processes and monetary outcomes. Big data comes with the prospect of helping them find their own wealth.

Big data is a new name for an old problem. Institutes of meteorology were pioneers in adopting this approach. They looked toward the satellite that could watch the clouds, and used it to gather numerous historical information to see if it would rain. With this information, farmers were able to determine trends for planting the next crop of soybeans, for example.

This analytic workload involved a high investment for scientific supercomputers, which were expensive. With the evolution of technology, this computational power has become available for commercial applications, allowing companies to adopt strategies for big data. For example, a large supermarket might notice that people who buy milk also buy diapers. However, in this example, the information is normally structured in a relational database.

Now the challenge is to correlate this data with unstructured data and do it quickly. Big data is considered a new kind of service oriented architecture (SOA), and all IT suppliers want to ride this new wave. Many suppliers have announced platforms to help companies handle big amounts of data more efficiently and extract important data from social networks and other unstructured sources.

Hadoop open source software is closely associated with the movement of big data. The market is promoting devices based on Hadoop, and data warehousing is transforming into a technology that is increasingly necessary. Another solution is Spark, which is 100x times faster than Hadoop when running programs in memory and 10x faster on disk.

To read more about Spark, see the Apache website:

https://spark.apache.org/

Big data is not just about the sheer volume of data that is being created. With a numberless of unstructured sources creating this data, a greater variety of data is now available. Each source produces this data at different rates, or what we call *velocity*. In addition, you still must solve the *veracity* of this new information as you do with structured data.

This is where the information management industry had its awakening moment: Whether your workload is largely transactional or online analytics processing (OLAP) and resource-intensive, both cases operate on *structured* data. Systems that are designed for the management and analysis of structured data provide valuable insight in the past, but what about all of the newer text-based data that is being created? This data is being generated everywhere you look. There is a larger volume of data, a greater variety of data, and it is being generated at a velocity that traditional methods of data management are no longer capable of efficiently harvesting or analyzing it.

To provide more insight into what is going on within your particular business arena, you must address the 4 Vs that define big data. Figure 2-1 on page 16 is a visual representation of those 4Vs.

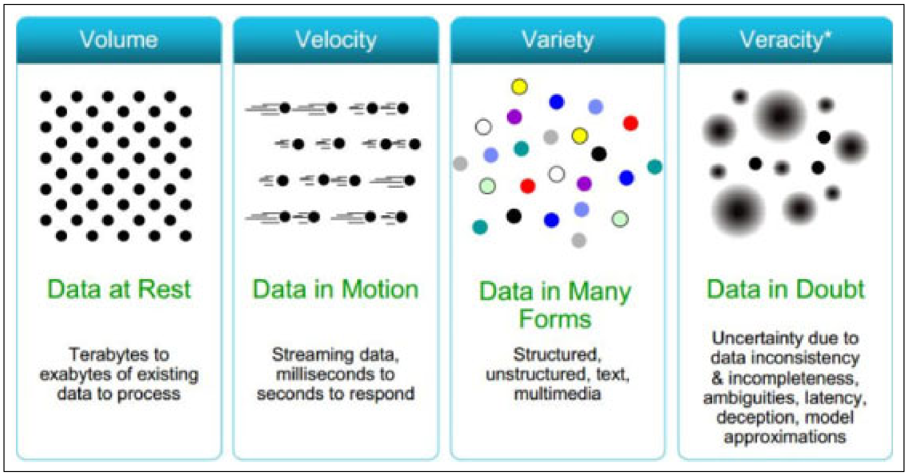


Figure 2-1 The 4 Vs of big data

2.2 Big data analytics

With information growing at fast rates and users who demand quick and effective research of this information, your analytics workloads need a powerful software base. IBM Platform Computing software improves the performance of your computing infrastructure for your most demanding analytics programs.

2.2.1 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, so this reduces efficiency for users and delays decision making.

2.2.2 Big data analytics solutions

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 7.1 offers several editions. Figure 2-2 on page 17 summarizes the features and differences between each edition.

Features	Developer	Express	Standard	Advanced
Low-latency HPC SOA	Yes	Yes	Yes	Yes
Agile service and task scheduling	Yes	Yes	Yes	Yes
Dynamic resource orchestration		Yes	Yes	Yes
Standard and custom reporting			Yes	Yes
Desktop, server, and VM harvesting capability			Yes	Yes
Data affinity				Yes
MapReduce framework	Yes			Yes
MultiCluster management				Yes
Maximum hosts per cores	2 hosts	240 cores	5K hosts, 128K cores	5K hosts, 128K cores
Maximum application managers		5	300	300

Product add-ons are optional and serve to enhance the functionality of the Standard and Advanced editions.

Add-ons	Standard	Advanced
Application Service Controller for Platform Symphony		Yes
Co-processor harvesting	Yes	Yes
Desktop harvesting	Yes	Yes
Server and VM harvesting	Yes	Yes
GPU harvesting	Yes	Yes

Figure 2-2 IBM Platform Symphony 7.1 editions

To have a better grid experience, use Platform Symphony Developer Edition to integrate with your applications. The Platform Symphony Developer Edition can be downloaded at no cost at the following website:

`http://www.ibm.com/developerworks/downloads/ibm/platformsymphony/`

The Platform Symphony Developer Edition software provides great resource availability and predictability. The software also supports several programs and file systems, operational maturity, SLA policy control, and high resource use for MapReduce and applications other than 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. It offers unmatched distributed workload runtime services for distributed computing and big data analytics programs.

2.2.3 IBM big data and analytics areas with solutions

There are six different areas where IBM big data and analytics can help discover fresh insights, capture the time-value of data, and act with confidence:

- Marketing
- Operations
- Finance and human resources
- New business models
- IT solutions

Marketing

Platform Symphony software uses marketing methods that allow sub-nanosecond response and quick provisioning for a wide range of workloads. Short-running jobs spend smaller percentage of time in provisioning and deprovisioning actions, so they provide a better ratio of useful work to overhead.

The solution also has a high job throughput rate. The system allows more than 17,000 tasks per second to be submitted.

Marketing analytics solutions from IBM can help you understand which strategies and offers appeal most to your high-value customers and prospects.

Marketing solutions

The following are the key analytics areas and the options available for them:

- Customer analytics

  Customer analytics solutions from IBM helps marketers understand and anticipate what customers want. These solutions provide the ability to target the best customers marketing programs, predict which customers are at-risk of leaving so you can retain them, and maximize customer lifetime value through personalized up-sell and cross-sell:

  – IBM Analytical Decision Management
  – IBM Cognos Business Intelligence
  – IBM Digital Analytics
  – IBM Predictive Customer Intelligence
  – IBM Social Media Analytics
  – IBM SPSS® Data Collection
  – IBM SPSS Modeler
  – IBM SPSS Statistics
  – IBM Tealeaf® CX
  – IBM Tealeaf cxImpact
  – IBM Tealeaf cxOverstat
  – IBM Tealeaf cxReveal
  – IBM Tealeaf CX Mobile

- Marketing performance analytics

  IBM Marketing Performance Analytics solutions give marketers the ability to measure ROI and eliminate the guesswork from marketing programs. Marketers rely on these solutions to access and analyze critical marketing metrics through customized reporting options such as dashboards, KPIs, and easy to understand visualizations:

  - IBM Business Intelligence
  - IBM Cognos Insight™
  - IBM Cognos Express
  - IBM SPSS Statistics
  - IBM SPSS Modeler
  - IBM Social Media Analytics

- Social media analytics

  The IBM Social Media Analytics solutions reveal the value of customer sentiment in social media. Marketers use these solutions to measure the social media impact of products, services, markets, and campaigns, and then use these insights to improve marketing programs and address customer satisfaction issues:

  - IBM Social Media Analytics
  - IBM Social Media Analytics, SaaS

Operations

IBM Predictive Maintenance and Quality can reduce asset downtime by analyzing asset data in real time, and detecting failure patterns and poor quality parts before problems occur.

Features

IBM Predictive Maintenance and Quality is preconfigured software, available either on the cloud or on-premises, which helps you monitor, maintain, and optimize assets for better availability, use, and performance. The software analyzes various types of data, including usage, wear, and conditional characteristics from disparate sources and detects failure patterns and poor-quality parts earlier than traditional quality control methods.

The goal is to reduce unscheduled asset down time and ensure that your quality metrics are achieved or exceeded. The product sends those insights, combined with your institutional knowledge, to provide optimized recommended decisions to people and systems. With Predictive Maintenance and Quality, organizations can optimize operations and supply chain processes. This results in better quality products, higher profit margins, and competitive advantages.

Predictive Maintenance and Quality includes these capabilities:

- Real-time capabilities

  Integrate, manage, and analyze sensor and real-time information in combination with existing static data.

- Big data, predictive analytics and business intelligence

  Combine predictive modeling, decision management, workflows and dashboards, and early warning algorithms in coordination with all types and volumes of data.

- Open architecture and data integration

  Link to many systems and data sources with included connectors and APIs.

- Process integration

  Deliver insights and recommendations and process work orders in existing enterprise asset management (EAM) systems.

Finance and human resources

IBM financial analysis solutions enable analysts to create and maintain complex models of business structures, dimensions, and data sets to provide more insights into opportunities and risks. They help your finance team identify the drivers of profitability and performance. They deliver the insights that you need to make smarter decisions about revenue, profit, cash flow, and the full range of variables that affect financial performance. Essential tasks such as variance analysis, scenario modeling, and what-if analysis are easier and faster with financial analysis solutions from IBM.

Financial analysis software from IBM helps your finance team perform the following tasks:

- Create and maintain complex, multidimensional models of business structures, dimensions and data sets.
- Examine historical performance and compare it to current and forecasted performance results, and then modify assumptions to test plans, budgets and forecasts.
- Analyze profitability by product, customer, channel, region, and more to gain new insights into opportunities and risks.
- Identify the actions needed to better align financial and operational resources so that resources can be shifted to the most profitable areas of the business.

New business models

IBM SPSS Analytic Catalyst makes analysis and discovery on big data more accessible to business users by presenting analyses visually and using plain-language summaries.

SPSS Analytic Catalyst uses the power of SPSS Analytics Server to help accelerate analytics by identifying key drivers from big data. It automates portions of data preparation, automatically interprets results, and presents analyses in interactive visuals with plain-language summaries. As a result, statistical analysis and discovery on big data are all more accessible to business users.

SPSS Analytic Catalyst offers the following features and benefits:

- Automated key driver identification with sophisticated algorithms, automatic testing, and regression-based techniques
- Interactive visuals and plain-language summaries of predictive analytics findings that provide insights at a glance, supporting explanations and statistical details
- Accelerated predictive analytics in big data environments with field associations, decision trees, drill down capability, and functions for saving insights for later retrieval
- Distribution in an environment designed for big data and massive scale

IT solutions

These solutions maximize insights, ensure trust, improve IT economics, and harness and analyze all data, even real-time data streaming from the sensors and devices that make up the Internet of Things. They help ensure the privacy and security of that data and put the infrastructure in place to support advanced analytics, so your organization can take advantage of cloud-based services to accelerate innovation.

For cloud services

As you expand big data and analytics capabilities throughout your organization, you need to empower all of your business users to access and analyze data for faster insights. Taking delivery of software, solutions, infrastructure, platforms and services on the cloud can accelerate the value of big data and analytics capabilities and offer scalability with limited up-front investment.

The following options are available at the time of publication:

- Big data and analytics software as a service (SaaS)
- Business process as a service (BPaaS)
- Infrastructure as a service (IaaS)
- IBM Bluemix™ cloud platform

For business decisions

These solutions provide the big data and analytics capabilities that enable you to capture, analyze, and act on all relevant data, including the streams of data that are generated by a myriad of electronic devices. Marketers, sales managers, financial analysts, and other business users gain the insights they need to act in real time, based on timely, trusted data.

These capabilities introduce several ways to improve decisions:

- Analyze streaming data in real time as it flows through the organization.
- Make sense of unstructured data and put it into context with historical, structured data.
- Use predictive analytics and advanced algorithms to recommend actions in real time.
- Empower decision makers to act on insights in the moment, with confidence.

These are among the IBM business solutions to consider for more informed decisions:

- IBM SPSS Modeler Gold
- IBM InfoSphere Streams

For governance and security with trusted data

To enable decision makers to act with confidence, you need to ensure that the data they use is clean, timely, and accurate. There are two effective ways to do this:

- Information integration and governance:
  - This can help your organization understand information and analyze the data and its relationships.
  - Improve information by delivering accurate and current data.
  - Accelerate projects providing consistent information in time.
- IBM Security Intelligence with Big Data:

  IBM Security Intelligence with Big Data combines the real-time security visibility of IBM QRadar® Security Intelligence with custom analytics for big data. It includes the following key capabilities:

  - Real-time correlation and anomaly detection of diverse security data
  - High-speed querying of security intelligence data
  - Flexible big data analytics across structured and unstructured data, including security data, email, document and social media content, full packet capture data, business process data, and other information
  - Graphical front-end tool for visualizing and exploring big data
  - Forensics for deep visibility

  See Figure 2-3 on page 22 for more information about IBM Security Intelligence with Big Data.

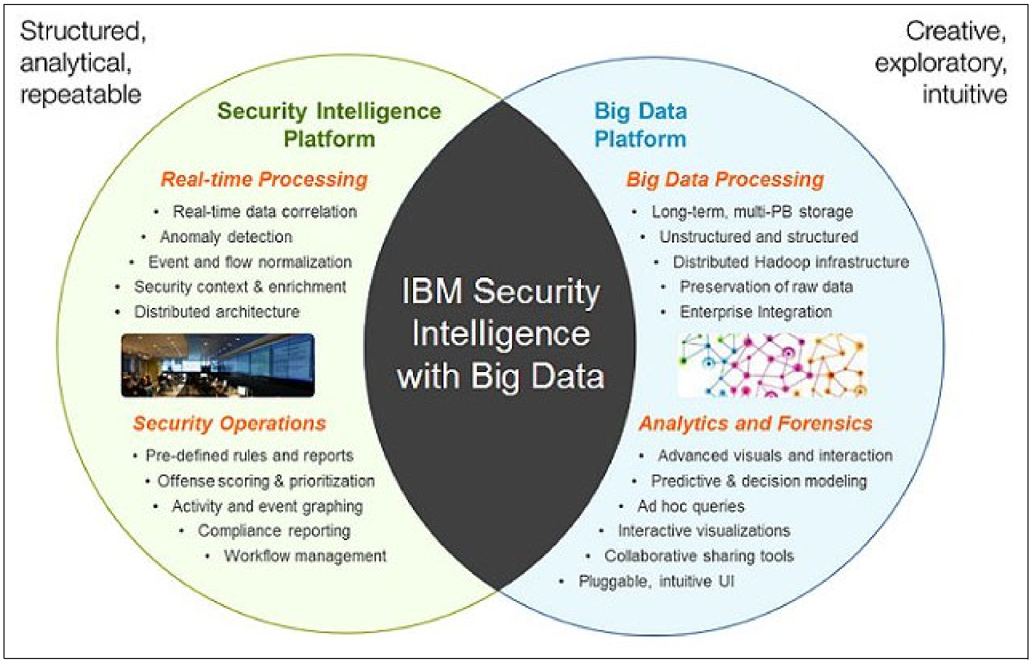


Figure 2-3 IBM Security Intelligence with Big Data

The infrastructure to maximize insights

A comprehensive big data and analytics platform such as IBM Watson Foundations needs the support of an infrastructure that takes advantage of technologies such as Hadoop to gain insights from streaming data, as well as data at rest. Integrated, high-performance systems, whether deployed on-premises or on the cloud, can reduce IT complexity and enable your organization to infuse analytics everywhere. This infrastructure includes the following components:

- IBM Solution for Hadoop, Power Systems Edition

  http://ibm.co/1KdtwLM

- IBM BLU Acceleration Solution, Power Systems Edition

  http://ibm.co/1SLn7su

- IBM Solution for Analytics, Power Systems Edition

  http://ibm.co/1FBSexc

2.2.4 Big data analytics advantage

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

2.3 IBM risk analytics solution advantages

IBM risk analytics solutions can help you balance risk and opportunity and make more informed decisions, based on risk analysis.

IBM risk analytics solutions enable the world's most successful companies to make risk-aware decisions through smarter enterprise risk management programs and methodologies. These improve business performance and outcomes. The combined risk management capabilities that are described in 2.3.1, "Algorithmics software" on page 23 and 2.3.2, "IBM OpenPages Operational Risk Management software" on page 23 can help your company achieve profitable growth and address increasing demands for regulatory compliance in today's volatile and complex market conditions.

With IBM risk analytics solutions, you can improve decision making by providing risk analysis and reduce the cost of regulatory compliance.

2.3.1 Algorithmics software

IBM Algorithmics® software enables financial institutions and corporate treasuries to make risk-aware business decisions. Supported by a global team of risk experts who are based in all major financial centers, the following IBM Algorithmics products and solutions address market, credit, and liquidity risk, as well as collateral and capital management:

- IBM Algo® Risk
- IBM Algo Risk® Service on Cloud
- IBM Algo Asset Liability Management
- IBM Algo Collateral Management

For more information, see the following web page:

http://www.ibm.com/software/analytics/algorithmics/

2.3.2 IBM OpenPages Operational Risk Management software

IBM OpenPages® Operational Risk Management automates the process of identifying, analyzing, and managing operational risk and enables businesses to integrate risk data into a single environment. This integrated approach helps improve visibility into risk exposure, reduce loss, and improve business performance. OpenPages enables you to embed operational risk management practices in the corporate culture, which makes procedures more effective and efficient. OpenPages GRC software enables your organization to manage enterprise operational risk and compliance initiatives using a single, integrated solution.

There are several versions to meet specific needs:

- OpenPages GRC on Cloud
- OpenPages GRC Platform
- OpenPages Operational Risk Management
- OpenPages Policy and Compliance Management
- OpenPages Financial Controls Management
- OpenPages IT Governance
- OpenPages Internal Audit Management

See the following web page for more information:

http://www.ibm.com/software/products/en/openpages-operational-risk-management

2.4 Scenario: How to minimize risk and build a better model

Sections 2.4.1, “Algo Market Risk analysis” on page 24 and 2.4.2, “IBM SPSS Statistics, Monte Carlo simulation” on page 24 provide brief descriptions that will help you better understand the example that follows. Then, 2.4.3, “Scenario” on page 25, shows how you can minimize the risk of money loss by using IBM Algo Market® Risk if you want to open a new retail store.

For more information, see the “Market risk management” web page:

http://www.ibm.com/software/analytics/risk-management/market-risk-management/

2.4.1 Algo Market Risk analysis

Algo Market Risk is a scenario-based solution that helps to measure and manage market risk. Its Monte Carlo simulations of mark-to-market valuations allow banks and financial institutions to significantly reduce regulatory capital requirements and help increase their returns on capital.

Algo Market Risk includes the following features:

- Proven, advanced analytics and risk reporting that delivers the highly accurate risk insights needed to help banks reduce their regulatory capital expenses
- Comprehensive instrument coverage that spans 20 geographic markets and 400 financial products
- Scenario-based portfolio optimization for proactive, risk-informed decision making
- Advanced computational speed that integrates the front and middle offices for active management of risk
- Customizable and scalable analytics that support the evolving needs of the enterprise

2.4.2 IBM SPSS Statistics, Monte Carlo simulation

SPSS Statistics combines the power of predictive analytics with the what-if capabilities of Monte Carlo simulation to help you in several essential ways:

- Go beyond conventional what-if analysis.

  Explore hundreds or thousands of combinations of factors and analyze all possible outcomes for significantly more accurate results.
- Identify the factors with the most impact.

  Quickly identify the factors in your model with the most impact on business outcomes.
- Gain a competitive advantage.

  Knowing what is likely to happen next enables you to offer the right products, target the right customers, or gain other advantages over competitors that lack this insight.
- Achieve better outcomes.

  Because you can predict results accurately, you can adjust your business strategies and processes to help you make the right decisions quickly and further reduce risk.

2.4.3 Scenario

In this scenario, assume that you have retail data based on other stores, and you want to use it as a starting point.

You want to know what is the likelihood of reaching your target number in the first months, for example, $7.5 million dollars in sales. You will use the available data to build a model that includes the following information:

- Advertising budget
- Customer confidence index
- Number of sales agents
- Monthly store visit by individual customers
- Previous month's income

You need to analyze this data and run a simulation. Assume that you have an existing simulation plan and you use that.

Currently, you are interested in the effect of direct and indirect advertising, and how it can decrease the risk of loss. First, you set fixed values, for example USD $50,000.

The simulation runs thousands of times. The results show that, based on the given parameters and the $50,000 advertising budget, you have a 52% chance of reaching the goal that you specified. With this result, you cannot convince management, so you need to run the simulation again to see whether you get a better result.

After you set the new budget for advertising to $70,000 and rerun the simulation, you get a result of a 68% chance, which is much more acceptable. If you are increasing the budget for advertising, you can decrease the risk factor. In this way, you can maximize the chance of reaching your target.

For more information about this scenario, watch the IBM Insights™ video titled "Monte Carlo Simulation and SPSS Statistics: Minimize Risk and Build Better Models" on YouTube:

https://www.youtube.com/watch?v=L_8VV1yXEjc

3

IBM Platform Symphony with Application Service Controller

This chapter describes the IBM Platform Symphony version 7.1 with the Application Service Controller add on. It also describes advantages of new technology, such as Platform Symphony working with the Apache Spark engine for large-scale data processing.

Platform Symphony 7.1 provides increased scaling and performance. The Application Service Controller helps you better manage cloud-native distributed computing environments, eliminate silos, and use available resources efficiently. It also offers several other benefits:

- Faster throughput and performance
- Higher levels of resource use
- Reduced infrastructure and management costs
- Reduced application development and maintenance costs
- The agility to respond instantly to real-time demands
- Improved management of heterogeneous distributed applications

The following topics are covered in this chapter:

3.1 Introduction to IBM Platform Symphony v7.1

IBM Platform Symphony provides a powerful application framework that enables you to run distributed or parallel applications in a scaled-out grid environment. It virtualizes compute-intensive application services and processes across existing heterogeneous IT resources. You can use Platform Symphony to run preintegrated applications that are available from a variety of independent software vendors (ISVs). You can either take advantage of new technologies, such as running the MapReduce framework using Platform Symphony Advanced Edition for data at rest and Apache Spark streams using Application Service Controller for data at motion together in the same cluster, setting priorities and scheduling according to your business needs in a multitenant way.

The IBM Platform Symphony v7.1 includes the following enhancements:

- Improved scale and performance

  3X increased scalability and improved performance across core Platform Symphony and MapReduce workloads

- Innovative data management technologies

  Data bottlenecks removed and data movement reduced

- Enhanced multitenancy and resource management

  Runtime elasticity with new cloud-native applications addressed

- Expanded workload management

  Emerging application workload patterns managed

The efficient low-latency middleware and scheduling architecture deliver the performance and agility that are required to predictably meet and exceed throughput goals for the most demanding analytic workloads. Platform Symphony helps organizations improve application performance at a significantly reduced total cost of ownership.

Platform Symphony can also help you attain higher-quality business results faster, reduce infrastructure and management costs, accelerate many types of Hadoop MapReduce workloads, and combine compute- and data-intensive applications in a single shared platform.

This software includes the following features:

- Ultrafast, low-latency grid scheduler
- Multicluster support for scalability to 128,000 service instances per cluster (typically mapped to cores)
- Unique resource-sharing model that enables multitenancy with resource lending and borrowing for maximum efficiency
- Optimized, low-latency MapReduce implementation compatible with IBM InfoSphere BigInsights and other big data solutions

Figure 3-1 on page 29 shows the target audience for IBM Platform Symphony.

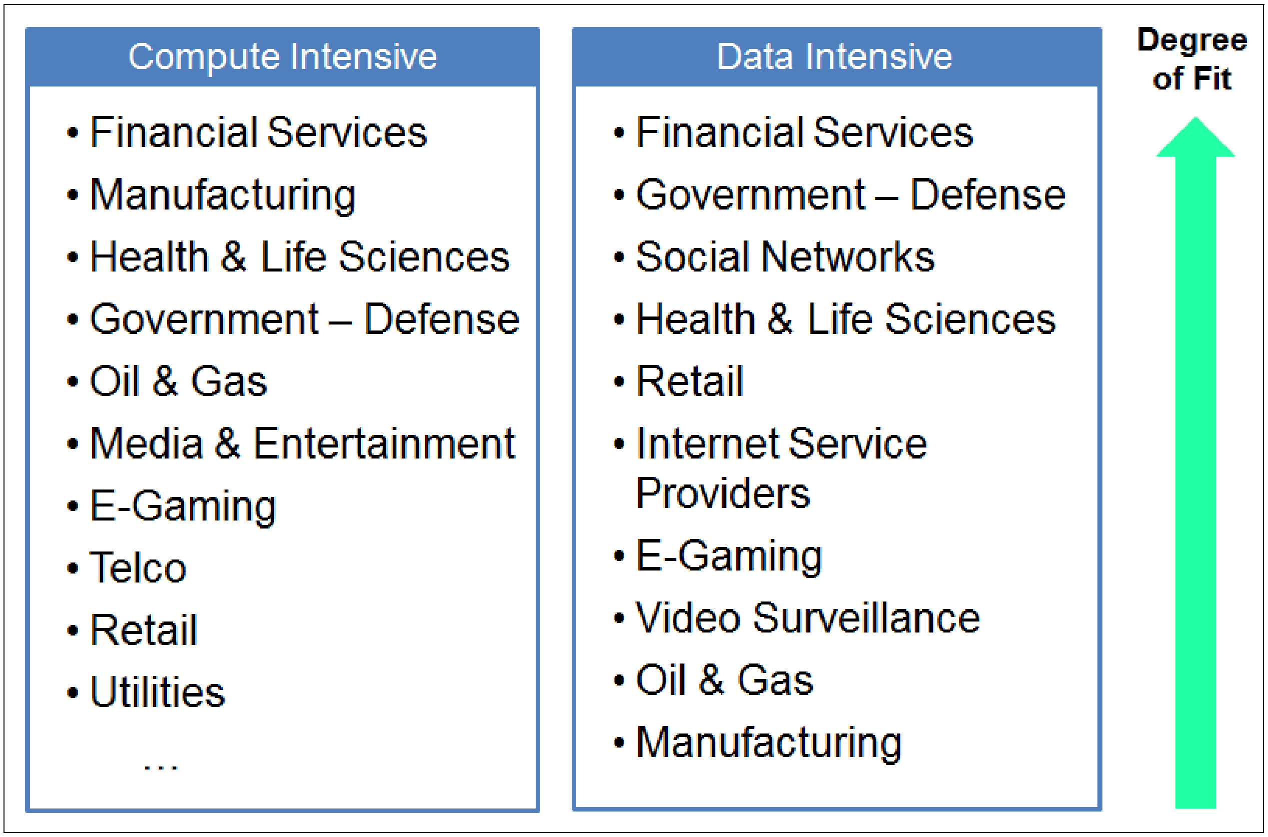


Figure 3-1 Target IBM Platform Symphony markets

3.2 How it operates

Platform Symphony is enterprise-class software that distributes and virtualizes compute-intensive application services and processes across existing heterogeneous IT resources. Platform Symphony creates a shared, scalable, and fault-tolerant infrastructure, delivering faster, more reliable application performance while reducing cost.

Platform Symphony provides an application framework that allows you to run distributed or parallel applications in a scaled-out grid environment.

> **Note:** For more details about how IBM Platform Symphony works internally, download the IBM Platform Symphony Foundations document from the following web page:
>
> `http://publibfp.dhe.ibm.com/epubs/pdf/c2750652.pdf`

Figure 3-2 illustrates the Platform Symphony application framework.

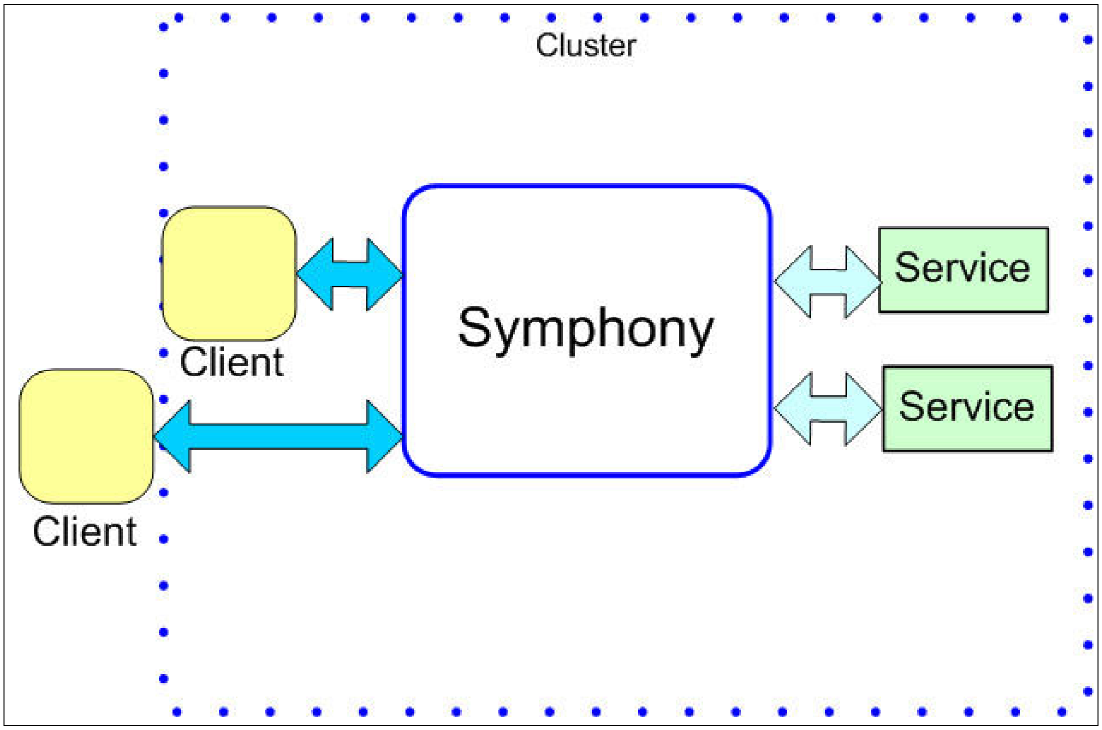


Figure 3-2 Platform Symphony application framework

3.2.1 Cluster management

A *cluster* is a logical grouping of hosts that provides a distributed environment in which to run applications. Platform Symphony manages the resources and the workload in the cluster.

When using Platform Symphony, resources are virtualized. It dynamically and flexibly assigns resources, provisions, them and makes them available for applications to use.

Platform Symphony can assign resources to an application on demand when the work is submitted, or assignment can be predetermined and preconfigured.

Cluster components

A Platform Symphony cluster manages both workload and resources. Platform Symphony maintains historical data, includes a web interface for administration and configuration, and has a command-line interface for administration.

3.2.2 Application

A Platform Symphony service-oriented application uses a client-service architecture. It consists of two programs: The *client*, which provides the client logic to submit work, retrieve and process results, and the *service*, which comprises the business logic (the computation). The service-oriented application uses parallel processing to accelerate computations.

Platform Symphony receives requests to run applications from a client and manages the scheduling and running of the work. The client is not affected by where the application runs.

Client

The client sends compute requests and collects results by using the Platform Symphony client APIs. The client can run on a machine that is part of the cluster or outside of the cluster. The client can use a service without detecting what programming language was used to create the service.

The client submits an input data request to Platform Symphony, which then initiates the service that processes the client requests, receives results from the service, and passes the results back to the client.

Service

The service is a self-contained business function that accepts requests from a client, performs a computation, and returns responses to the client. The service uses computing resources and must be deployed to the cluster. Multiple instances of a service can run concurrently in the cluster.

The service is initiated and run by Platform Symphony upon receipt of a client request. The service runs on a machine that is part of the Platform Symphony cluster. The service runs on the cluster resources that are dynamically provisioned by Platform Symphony, which monitors the running of the service and passes the results back to the client.

3.2.3 How workload management differs from resource management

A workload manager interfaces directly with the application, receives work, processes it, and returns the results. It provides a set of APIs, or it might interface with additional runtime components to enable the application components to communicate and perform work. The workload manager is aware of the nature of the applications that it supports, using terminology and models consistent with the specified class of the workload. In a service-oriented application environment, the workload is expressed in terms of messages, sessions, and services.

A resource manager provides the underlying system infrastructure to enable multiple applications to operate within a shared resource infrastructure. It manages the computing resources for all types of workloads.

How Platform Symphony supplies resources

To understand how Platform Symphony supplies resources to meet workload requests, consider the analogy that follows.

A bank customer does not withdraw funds directly from the bank vaults. The customer accesses an account and requests a withdrawal from that account. The bank recognizes the customer by the account number and determines whether the customer has sufficient funds to make a withdrawal. See Figure 3-3 on page 32.

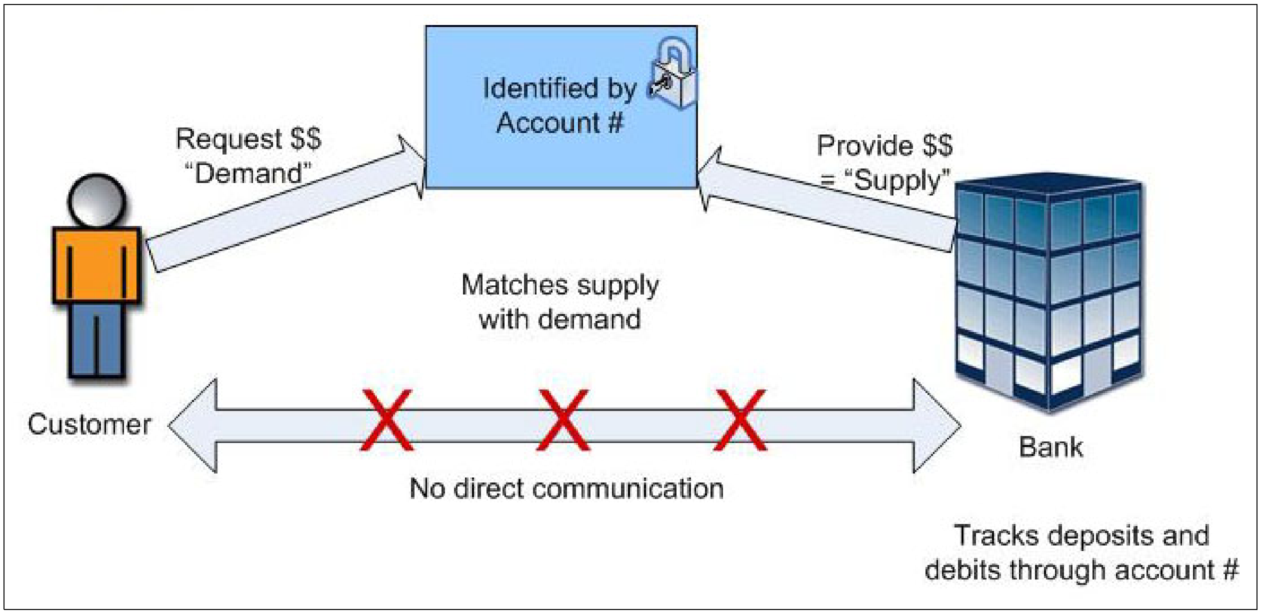
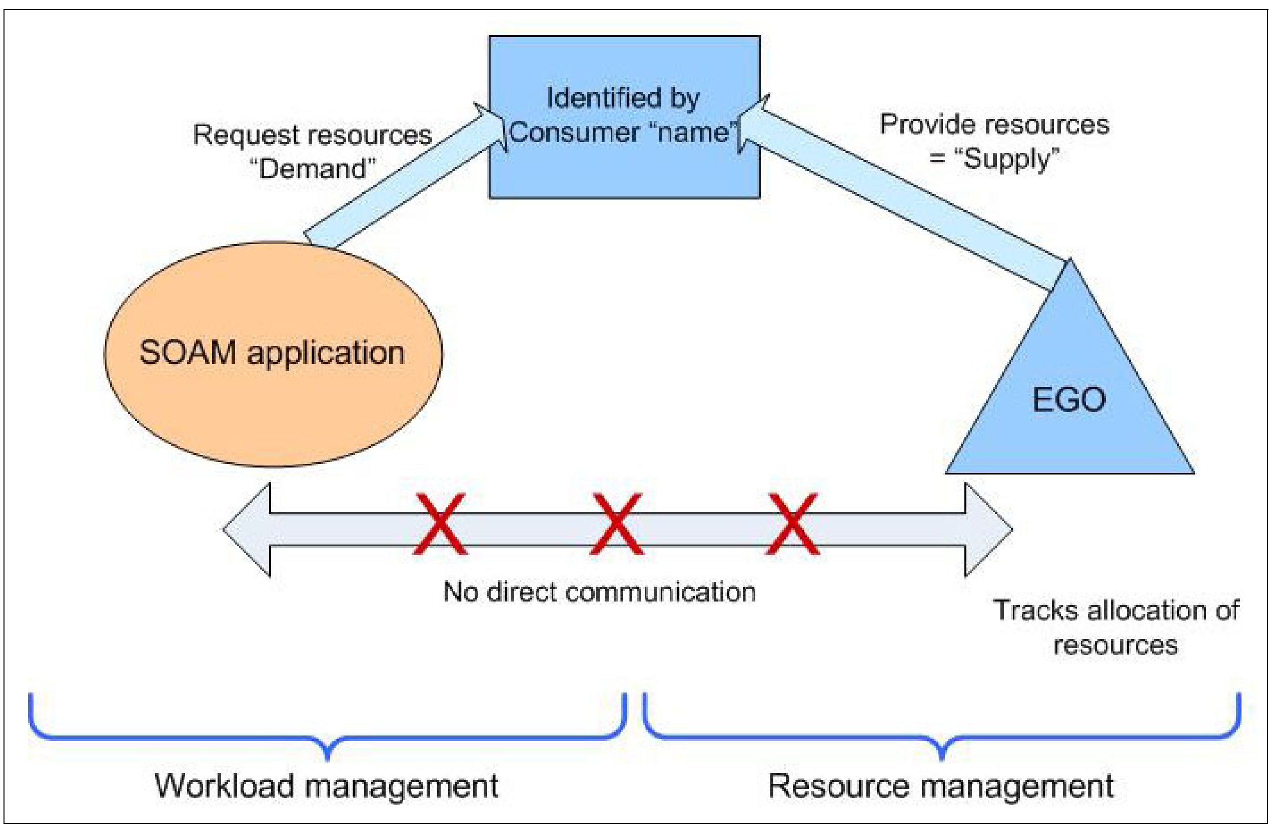


Figure 3-3 This picture illustrates how Platform Symphony supplies resources

Figure 3-4 on page 37 shows that when a Platform Symphony application requires resources, it does not communicate directly with EGO and has no direct access to resources. The application is associated with a consumer and requests resources through that account name. EGO recognizes the consumer and, through it, allocates resources to the application.



This diagram illustrates how Platform Symphony supplies resources

EGO resource manager

Enterprise Grid Orchestrator (EGO) manages the supply and distribution of resources and makes them available to applications. EGO provides resource provisioning, remote execution, high availability, and business continuity.

EGO also provides cluster management tools and the ability to manage supply according to demand to meet service-level agreements.

SOA middleware workload manager

The SOA middleware (SOAM) manages service-oriented application workloads within the cluster, creating a demand for cluster resources. When a client submits an application request, the request is received by SOAM. Then, SOAM manages the scheduling of the workload to its assigned resources, requesting additional resources as required to meet service-level agreements. SOAM transfers input from the client to the service and returns results to the client. It releases excess resources to the resource manager.

3.2.4 Platform Management Console

The Platform Management Console (PMC) is your window to Platform Symphony, providing resource monitoring capability, application service-level monitoring and control, and configuration tools.

Platform Symphony stores a wide variety of historical data for reporting and diagnostic purposes. Multiple reports capture and summarize the data.

3.3 IBM Platform Symphony for multitenant designs

Multitenancy is an architecture in which a single instance of a software application serves multiple customers. Each customer is called a *tenant*. Tenants might be given the ability to customize some parts of the application, such as color of the user interface (UI) or business rules, but they cannot customize the application's code.

3.3.1 The narrow view of multitenancy

> **Note:** In a multitenancy environment, multiple customers share the same application that runs on the same operating system, on the same hardware, with the same data-storage mechanism.

Big data and analytics infrastructure silos are inefficient. Platform Symphony helps you achieve the best results by using multitenancy.

3.3.2 Advantages and challenges

By using a shared infrastructure environment, this service reduces hardware, software, and environmental costs while maintaining a secured infrastructure through isolated LPARs and IBM's comprehensive managed services. The solution offers an allocation-based consumption model that further reduces costs, so you pay for only what is allocated to you. The savings are from extending the cost of hardware and software across the entire multitenant customer environment. The service also provides dynamic capacity to meet peak workload requirements and growth as business needs change.

There are several challenges associated with this solution:

- Increasing cost of analytics
- Addressing challenges associated with the ETL (extract, transform, load) process
- Accommodating data warehouse volume growth
- Delivering needed information in a timely manner
- Ensuring that information is available when needed
- Management of the Hadoop environment

Ever-increasing expectations increase technical needs, too:

- Increased performance to support business demands
- Increased scalability to address huge and growing volumes of data
- Optimized use of existing resources for scaled performance
- Efficient data management to remove data bottlenecks
- Support for new, cloud-native application workload patterns
- Effective operational management: monitoring, alerts, diagnostics, and security

3.3.3 Multitenant designs

In general, multitenancy implies multiple unrelated users or customers of a set of services. Within a single organization, this can be multiple business units with resources and data that must remain separate for legal or compliance reasons. Most hosting companies require multitenancy as a core attribute of their business model. This might include a dedicated physical infrastructure for each hosted customer or logical segmentation of a shared infrastructure by using software-defined technologies.

In Platform Symphony Advanced Edition, up to 300 MapReduce runtime engines (*job trackers*) can coexist and use the same infrastructure.

Users can define multiple MapReduce applications and associate them with resource consumers by "cloning" the default MapReduce application. Each application has its separate and unique job tracker called session manager (SSM). When there are multiple instances of SSMs, they are balanced on the available management nodes.

Within each application, simultaneous job management is possible because of the design that implements sophisticated scheduling of multiple sessions on the resources that are allocated for an application. This function is possible by separating the job control function (workload manager) from the resource allocation and control (Enterprise Grid Orchestrator). The new YARN, Apache Hadoop 2, has a similar feature, but the release is still in alpha stage. The stable release of Hadoop MapReduce offers only one job tracker per cluster.

Moreover, multitenancy is much more than just multiple job trackers. It is about user security, shared and controlled access to the computing resources and to the whole environment, monitoring and reporting features, and so on. These multitenancy features are addressed as they are implemented by the Platform Symphony solution.

3.3.4 Requirements gathering

Requirements gathering helps you determine how the consumer can be aware of the hosted applications and how is the best way to request access to these hosted services.

While gathering requirements, get answers the following questions:

- Will consumers use accounts that the host creates, or will they use accounts that they use internally to access services?
- Is one consumer allowed to be aware of other consumer's identities, or is a separation required?
- Can multiple consumers share a physical infrastructure?
- Can traffic from multiple consumers share a common network?
- Can software-defined isolation meet the requirements?
- How far into the infrastructure must authentication, authorization, and accounting be maintained for each consumer?

There are also segmentation options to consider as part of a multitenant infrastructure:

- Physical separation by customer (dedicated hosts, network, storage)
- Logical separation by customer (shared physical infrastructure with logical segmentation)
- Data separation
- Network separation (virtual LANS, or VLANs)
- Performance separation (shared infrastructure but guaranteed capacity)

3.3.5 Building a multitenant big data infrastructure

Platform Symphony provides a platform for robust, multi-computer automation for all elements of a data center, including servers, operating systems, storage, and networking. It also provides centralized administration and management capabilities, such as deploying roles and features remotely to physical and virtual servers and deploying roles and features to virtual hard disk drives, even when they are offline.

3.4 Platform Symphony concepts

Although you are probably familiar with Hadoop and various commercial distributions, you might be less familiar with Platform Symphony. It is commercial grid workload and resource management software that has been used to share resources among diverse applications in multitenant environments for more than a decade. Platform Symphony is widely deployed as a shared services infrastructure in some of the world's largest investment banks.

3.4.1 Session manager

Service-oriented applications in Platform Symphony are managed by a session manager. The session manager is responsible for dispatching tasks to service instances and for collecting and assembling results. It provides a function that is similar in concept to a Hadoop application manager, although it has considerably more capabilities. Platform Symphony implements job tracker function by using the session manager.

> **Note:** In this IBM Redbooks publication, the terms *job tracker*, *application manager*, and *session manager* are used interchangeably.

Although the concept of multiple concurrent application managers in Hadoop is new with YARN. Platform Symphony has always featured a multitenant design.

3.4.2 Resource groups

Unlike Hadoop clusters, Platform Symphony does not make assumptions about the capabilities of hosts that participate in the cluster. Although Hadoop generally assumes that member nodes are 64-bit Linux hosts running Java, Platform Symphony supports a variety of hardware platforms and operating environments. It allows hosts to be grouped in flexible ways into different resource groups, and different types of applications can share these underlying resource groups in flexible ways.

3.4.3 Applications

The term *application* can be a bit confusing as it is applied to Platform Symphony. Platform Symphony views an application as the combination of the client-side and service-side code that comprise a distributed application. This is a more expanded definition than most people are used to using.

By this definition, an instance of InfoSphere BigInsights might be viewed as a single application. All of these are examples of Platform Symphony applications:

- Custom applications written in C++
- A commercial ISV application, such as IBM Algorithmics, Calypso, or Murex
- A commercial or open source Hadoop application, such as Cloudera, IBM InfoSphere BigInsights, or open source Hadoop

Platform Symphony views applications as being an instance of middleware. Various client side tools associated with a particular version of Hadoop (Pig, Hive, Sqoop, and so on) can all run against a single Hadoop application definition. An important concept for those not familiar with Platform Symphony is that Platform Symphony provisions service instances associated with different applications dynamically. As a result, there is nothing technically stopping a Platform Symphony cluster from supporting multiple instances of Hadoop and non-Hadoop environments concurrently.

Figure 3-4 on page 37 shows the result of choosing the **Workload** → **Symphony** → **Applications** menu.

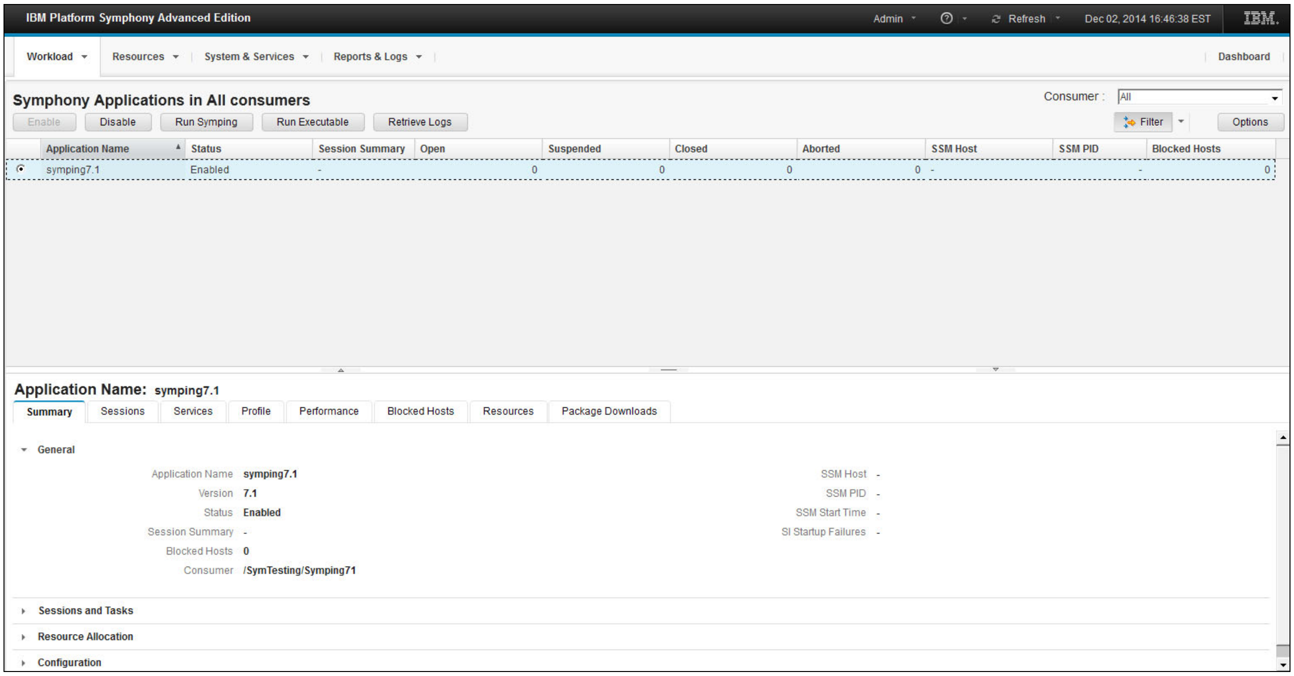


Figure 3-4 Applications menu

3.4.4 Application profiles

As explained previously in this chapter, applications in Platform Symphony are flexible and highly configurable constructs. An application profile in Platform Symphony defines the characteristics of an application and various behaviors at runtime.

Figure 3-5 shows result of selecting **Workload** → **Symphony** → **Application Profiles** menu.

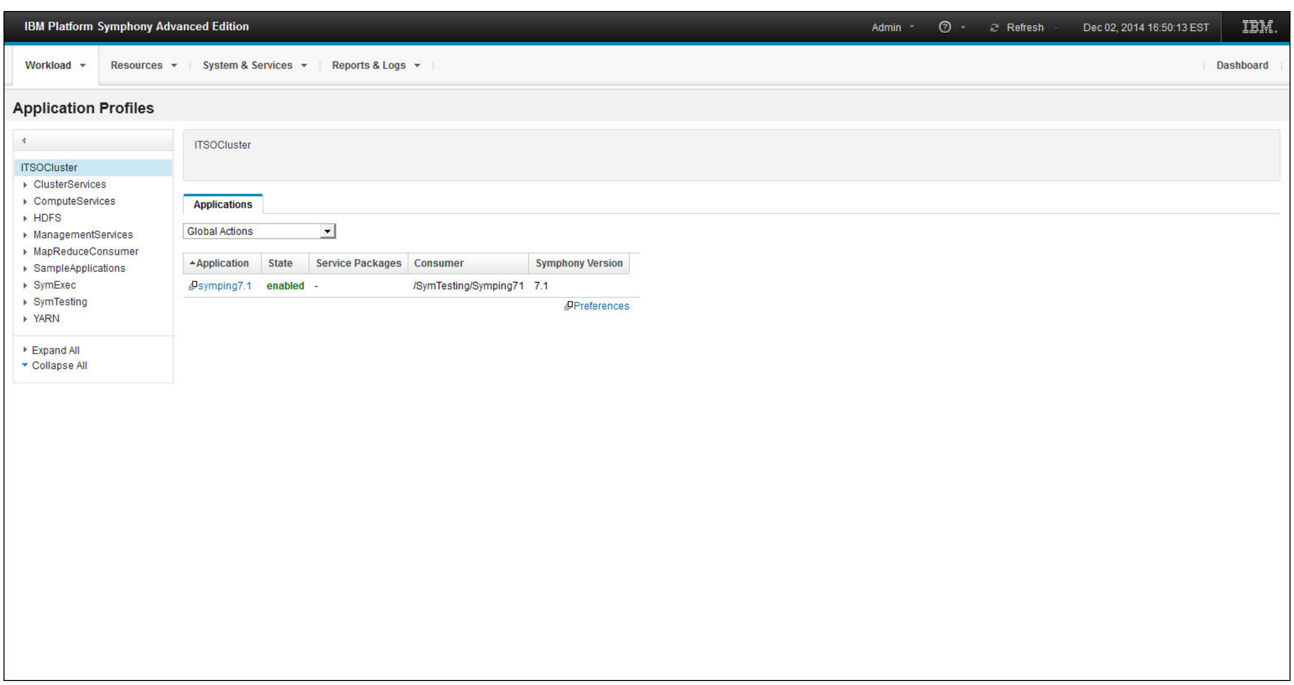


Figure 3-5 Application profiles

3.4.5 Consumers

From the viewpoint of a resource manager, an application or tenant in the cluster is defined as something that needs particular types of resources at runtime. Platform Symphony uses the term *consumer* to define these users of resources. The software provides capabilities to define hierarchical consumer trees and express business rules about how consumers share various types of resources that are collected into resource groups. The leaf nodes in consumer trees map to a Platform Symphony application.

3.4.6 Services

Services are the portions of applications that run on cluster nodes. In a Hadoop context, administrators might think of services as equating to a task tracker that runs Map and Reduce logic. Here again, Platform Symphony takes a broader view.

Platform Symphony services are generic. A service might be a task tracker associated with a particular version of Hadoop or it might be something else entirely. When the MapReduce framework is used in Platform Symphony, the Hadoop service-side code that implements the task tracker logic is dynamically provisioned by Platform Symphony.

Platform Symphony is named for this ability to orchestrate a variety of services quickly and dynamically according to sophisticated sharing policies.

3.4.7 Sessions

A *session* in Platform Symphony equates to a job in Hadoop.

A client application in Platform Symphony normally opens a connection to the cluster, selects an application, and opens a session. Behind the scenes, Platform Symphony provisions a session manager to manage the lifecycle of the job. A single Platform Symphony session manager might support multiple sessions (Hadoop jobs) concurrently.

A Hadoop job is a special case of a Platform Symphony job. The Hadoop client starts a session manager that provides *job tracker* functions. Platform Symphony uses the job tracker and task tracker code provided in a Hadoop distribution. However, it uses its own low-latency middleware to more efficiently orchestrate these services in a shared cluster.

3.4.8 Repositories

As explained in previous sections, Platform Symphony dynamically orchestrates service-side code in response to application demand. The binary code that comprises an application service is stored in a repository. Normally, for Platform Symphony applications, services are distributed to compute nodes from a repository service. For Hadoop applications, code can be distributed via either the repository service or the HDFS or the IBM Spectrum Scale File Placement Optimizer (FPO) file system.

3.4.9 Tasks

Platform Symphony jobs are collections of tasks. They are managed by a session manager that runs on a management host. The session manager makes sure that instances of the needed service are running on compute nodes/data nodes in the cluster. Services instances run under the control of a Symphony Service Instance Manager (SIM).

MapReduce jobs in Platform Symphony work the same way, but, in this example, the Platform Symphony service is essentially the Hadoop task tracker logic.

In Hadoop clusters, slots are normally designated as running either map logic or reduce logic. Again, in Platform Symphony this is fluid because services are orchestrated dynamically, so service instances can be either Map or Reduce tasks.

This is an advantage because it allows full use of the cluster as the job progresses. At the start of a job, the majority of slots can be allocated to Map tasks. Toward the end of the job, the slot function can be shifted to perform the Reduce function.

3.5 Benefits of using Platform Symphony

This section describes the benefits of implementing Platform Symphony in your environment.

First, the highlights:

- Monitoring of the cluster and Hadoop jobs
- Configuration and management of physical resources
- Failover and recovery logic for Hadoop jobs
- Reporting framework
- Enhanced Hadoop MR processing framework

You also benefit from the following advantages:

- Sophisticated scheduling engine
  - Priority-based scheduling
  - Preemptive scheduling
  - Fair share proportional scheduling
  - Threshold based scheduling
  - Task reclaim logic
  - Administrative control of running jobs
- Configuration and management features
  - Resource group-based and slot-based allocation
  - Consumer allocation
  - Shared resources, heterogeneous application support
  - GUI management console
  - Real-time monitoring and management of hosts: All global assets
- High availability
  - Failover scenarios
    - Host running job tracker fails
    - Host running map task fails
    - Host running reduce task fails
    - Job recovery
    - Services failover
- Enhanced MapReduce implementation
  - Low latency with immediate map allocation
  - Fast workload allocation
  - Small overhead to start jobs
  - Tools for meeting necessary service level objectives (SLO) and business continuity

3.5.1 Summary

Platform Symphony supports advanced multitenancy. Therefore, you can share a broader set of application types and scheduling patterns on a common resource foundation. Key advantages are better performance and better resource management.

3.6 Product edition highlights

Platform Symphony is available in four editions that are each tailored to different business requirements:

Developer Build and test applications without the need for a full-scale grid

Express The ideal solution for departmental clusters

Standard Enterprise-class performance and scalability

Advanced Ideal for distributed computing- and data-intensive applications that require Hadoop MapReduce or benefit from the advanced capabilities of the Application Service Controller add-on for Platform Symphony

Product add-ons are optional and enhance the functions of the Standard and Advanced editions. Table 3-1 summarizes the features that are associated with each Platform Symphony edition.

Table 3-1 IBM Platform Symphony features and editions

Features	**IBM Platform Symphony edition**			
	Developer	**Express**	**Standard**	**Advanced**
Low-latency HPC SOA	X	X	X	X
Agile service and task scheduling	X	X	X	X
Dynamic resource orchestration		X	X	X
Standard and custom reporting			X	X
Desktop, server, and virtual server harvesting capability			X	X
Data affinity				X
MapReduce framework	X			X

Table 3-2 shows the Platform Symphony add-ons that are available.

Table 3-2 IBM Platform Symphony add-ons for each edition

Add-ons	IBM Platform Symphony edition			
	Developer	Express	Standard	Advanced
Platform Application Service Controller				X
Desktop Harvesting			X	X
Server and Virtual Server Harvesting			X	X
Graphics processing units (GPU)			X	X
IBM Spectrum Scale (GPFS)			X	X
Spectrum Scale Shared Nothing Cluster (FPO)				X

3.6.1 Platform Symphony Developer Edition

Platform Symphony Developer Edition provides a complete test environment for application developers to grid-enable, test, and run their service-oriented applications.

The Developer Edition simulates the grid environment provided by Platform Symphony. Developers can test their clients and services in their own cluster of machines before deploying them to the grid.

Platform Symphony Developer Edition includes the following features:

- Easy-to-use APIs and rich design patterns to seamlessly grid-enable all types of service-oriented applications with minimal changes
- A MapReduce framework to run MapReduce applications with minimal changes:
  - Different modes for debugging MapReduce applications:
    - The standalone mode in which the entire MapReduce workflow runs in a single Java process on the local host
    - The pseudo-distributed mode in which each MapReduce daemon runs in separate Java processes on the local host
  - Support for distributed file systems, such as the open-source Apache Hadoop Distributed File System (HDFS), Cloudera's CDH open source distribution that includes Apache Hadoop, Appistry Cloud IQ, and IBM Spectrum Scale
  - Command-line utility called `mrsh` that automatically sets up the environment when submitting MapReduce jobs
  - A MapReduce service class definition that you can customize to implement custom lifecycle event handlers
  - A Java class wrapper that defines buffers as data containers, enabling you to copy and view large files as a sequence of bytes
- A web interface for monitoring and controlling your test environment for Platform Symphony and MapReduce workloads
- A knowledge center for easy access to documentation

To run a Platform Symphony workload on the grid, the application developer creates a service package and adds the service executable file into the package. No additional code changes are required.

Note: The Platform Symphony Developers Edition does not include the Enterprise Grid Orchestrator (EGO) resource management component. However, Platform Symphony Developers Edition includes an EGO stub to simulate basic EGO resource distribution.

Platform Management Console

The Platform Management Console (PMC) is the web interface to Platform Symphony and Application Service Controller for Platform Symphony. For Platform Symphony, the PMC provides a single point of access to the key system components for cluster and workload monitoring and control, configuration, and troubleshooting. For the Application Service Controller for Platform Symphony, PMC also provides a single point of access to manage and monitor your application instances.

Cluster and workload health dashboards

The dashboard page opens when you log into PMC. This page provides a quick overview of the health of your cluster. The dashboard shows a summary of the workload in the cluster, a summary of hosts use and status, and links to key pages in the Platform Management Console.

Note: The d*ashboard* only displays the information when the console is used to access the grid. The dashboard does not appear in Platform Symphony Developers Edition.

The *Common Tasks* menu is available on the upper right corner of the dashboard page and provides links to key pages in the PMC such as Platform Symphony Workload, Resources, Cluster Settings, System Logs, and Reports. Based on your entitlement, extra pages such as MapReduce workload and Application Service Controller workload also appear.

3.6.2 Platform Symphony Advanced Edition

Across a range of industries, organizations are collecting tremendous volumes of data, generated by a wide variety of sources, often at extreme speeds. Analyzing this big data can produce key insights for improving the customer experience, enhancing marketing effectiveness, increasing operational efficiencies, reducing financial risks and so on. Platform Symphony Advanced Edition software can help address the challenges of achieving outstanding performance for analyzing big data while controlling costs.

Platform Symphony Advanced Edition includes a best-of breed runtime engine for MapReduce applications that is fully compatible with popular MapReduce distributions including Hadoop and Spark MapReduce. Platform Symphony Advanced Edition delivers enterprise-class distributed computing capabilities for the MapReduce programming model. The solution meets enterprise IT requirements by delivering high resource use, availability, scalability, manageability and compatibility. These benefits all leads to the ability to deliver a higher quality of service aligned to customer service level requirements at a lower cost. As a platform for distributed MapReduce workloads, Platform Symphony Advanced Edition provides an open application architecture for both applications and file systems. It provides client and server side APIs for both MapReduce and non MapReduce applications supporting multiple programming languages. Also, its open architecture supports connections to multiple data types and storage file systems, including full compatibility with the open source Hadoop Distributed File System (HDFS).

A high level view of architecture of Platform Symphony Advanced Edition and the MapReduce framework is provided in Figure 3-6.

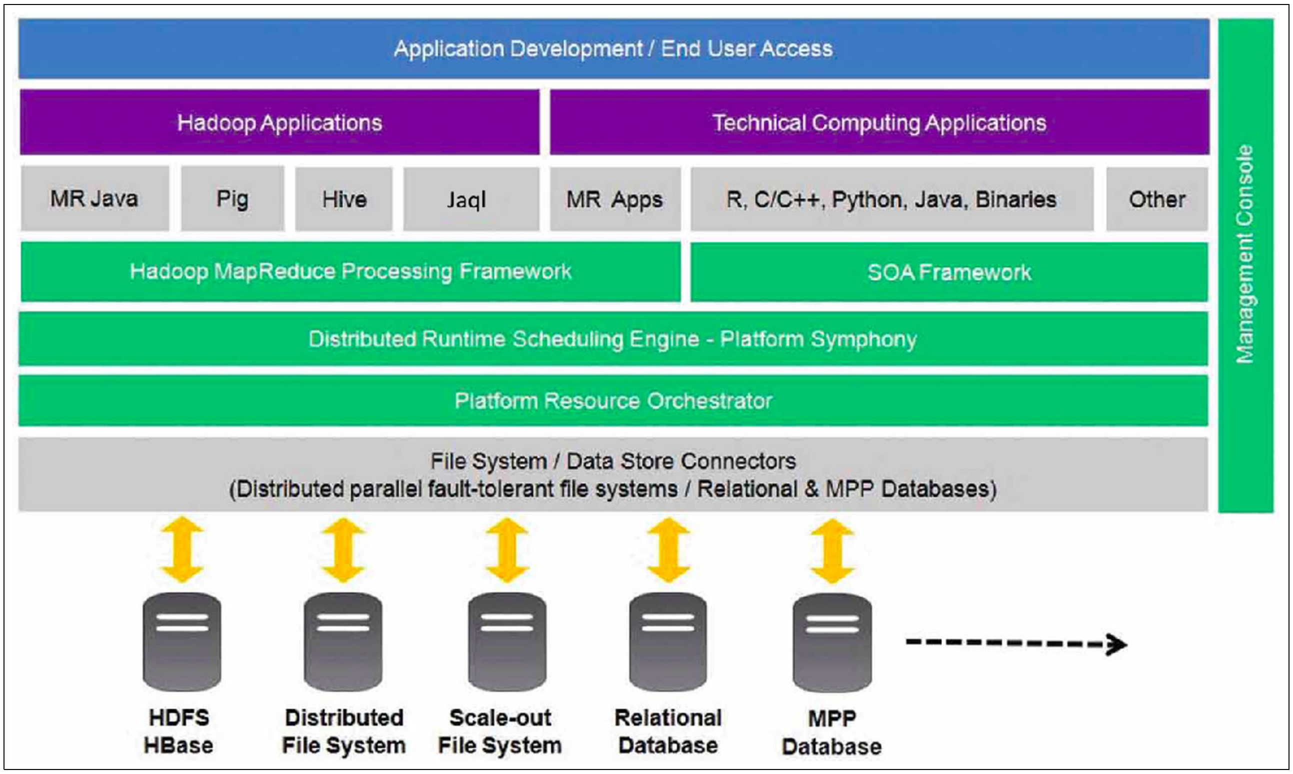


Figure 3-6 Platform Symphony Advanced Edition MapReduce framework

Platform Symphony Advanced Edition provides a variety of client and server side application programming interfaces to facilitate easy application integration and execution. These include a MapReduce APIs fully compatible with open source Hadoop as well as a variety of capabilities supporting commercial application integrations. These services allow developers to use the open source Hadoop logic and projects and easily port the resulting applications into the Platform Symphony's MapReduce architecture. Platform Symphony Advanced Edition also provides developers with a much richer set of tools to avoid performance bottlenecks and optimize performance by taking advantage of advanced Platform Symphony features like multi-core optimization, direct data transfer, and data-affinity.

3.7 Optional applications to extend Platform Symphony capabilities

Several add-on tools and complementary products can be used with both Platform Symphony Standard and Advanced editions. These products are all designed to help you do more while spending less.

- Desktop Harvesting

  This add-on harnesses the resources from available idle desktops, adding them to the pool of potential candidates to help complete tasks. Platform Symphony services do not interfere with other applications running on the desktops; and harvested resources are managed directly through the integrated management interface.

- Server/VM Harvesting

  To take full advantage of your enterprise's resources, the add-on taps idle or under-used servers and virtual machines (VMs). Instead of requiring new infrastructure investments, Platform Symphony locates and aggregates these server resources as part of the grid whenever additional capacity is needed to handle larger workloads, or when the speed of results is critical.

- GPU Harvesting

  To unleash the power of general-purpose graphic processing units (GPUs), this add-on enables applications to share expensive GPU resources more effectively and to scale beyond the confines of a single GPU. Sharing GPUs more efficiently among multiple applications, and detecting and addressing GPU-specific issues at run time helps improve service levels and reduce capital spending.

- Platform Analytics

  Platform Analytics is an advanced analysis and visualization tool for analyzing massive amounts of workload and infrastructure usage data collected from Platform Symphony clusters. This add-on enables you to easily correlate job, resources and license data from multiple Platform Symphony clusters for data driven decision making.

- Application Service Controller

  The Application Service Controller, available only in the Advanced Edition, extends the Platform Symphony grid to provide a shared-service backbone for a broad portfolio of distributed software frameworks. By enabling a wide variety of applications to share resources and coexist on the same infrastructure, the Application Service Controller helps organizations reduce cost, simplify management, increase efficiency and improve performance.

The next section describes the Application Service Controller extension.

3.8 Overview of the Application Service Controller add-on

IBM Platform Symphony v7.1 and the Application Service Controller for Platform Symphony help to exceed performance goals with a fast, efficient grid and analytic computing environment. At this version, Platform Symphony offers increased scaling and performance. Application Service Controller for Platform Symphony enables to better manage cloud-native distributed computing environments.

The Application Service Controller for Platform Symphony Advanced Edition is a generalized service controller for complex long running application services.

The Platform Application Service Controller for Platform Symphony add-on extends the Symphony grid to enable a shared-service backbone for a broad portfolio of distributed software frameworks. Designed specifically to address the requirements of a new generation of distributed application workloads that stem from the wide adoption of *born-on-the-cloud* technology, it increases resource use, minimizes application silos, and offers increased resiliency and high availability.

The Application Service Controller for Platform Symphony add-on is available for Platform Symphony Advanced Edition. Application Service Controller offers the following benefits:

- Increases use of existing hardware resources:
  - Reduces server idle time across a broader set of distributed applications, including a new generation of cloud-native workloads
  - Shares resources across applications, users, and lines of business
  - Defers the need for incremental capital investment
- Increases application performance:
  - Obtains bare metal performance with dynamic runtime elasticity: Manage demand at run time rather than at build time
  - Gains application isolation without virtual machines
  - Reduces application wait time
- Increases resiliency and high availability
- Improves management efficiencies: Reduces administration overhead for visualization, monitoring, alerting, reporting, application deployment, and lifecycle management.

Application Service Controller for Platform Symphony v7.1 is supported on the following operating system platforms:

- Microsoft Windows
- IBM PowerLinux™
- xLinux

3.8.1 Application Service Controller lifecycle

With Application Service Controller for Platform Symphony, you can create application instances using an application template. Figure 3-7 illustrates the basic tasks that are typically associated with using Application Service Controller for Platform Symphony.

> **Note:** Application Service Controller for Platform Symphony features can be configured through a RESTful API.

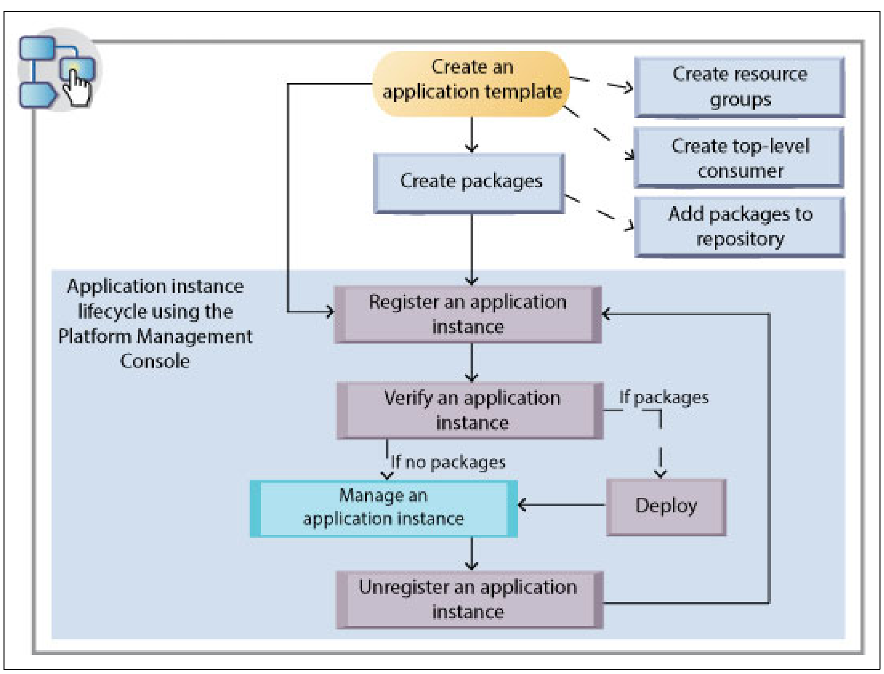


Figure 3-7 Tasks associated with Application Service Controller for Platform Symphony

First, create an application template. Next, you create the package(s) that are based on the application template. When you register an application instance, you can add the created package(s) to the Platform Symphony repository and you can specify consumers that you want to use.

Alternatively, if you want to define your own consumers and resource groups that are available to the application instance, you can create the resource groups, consumers, and add the packages to the repository ahead of time so that they can be used by multiple application instances.

After you register the application instance, verify that it was registered correctly. If the application instance has packages, deploy the application instance first, and then manage it.

If the application instance does not include packages, you start managing it, as there is no need for deployment. If you need to update your application template, you must unregister it.

3.8.2 Application framework integrations

Application Service Controller for Platform Symphony can be integrated with any distributed application framework to manage and run them on a scalable and shared grid.

The following are some of the application frameworks that are integrated with Application Service Controller for Platform Symphony:

- Apache Hadoop
- Apache Spark
- Apache YARN
- Cassandra
- Cloudera
- Hadoop
- Hortonworks
- MongoBD

> **Note:** For more information on the latest set of application frameworks integrations, see the following websites:
>
> http://ibm.github.io/
> https://hub.jazz.net/learn/

3.8.3 Basic concepts

To understand the Application Service Controller for Platform Symphony, the following concepts are described:

Application instance

An application instance is a collection of services and service groups that are associated with a top-level consumer. You can monitor and manage an application instance and drill down to manage the related services and service instances. You create (register) an application instance from an application template.

Application Service Controller service

Application Service Controller services can be part of an application instance or independent. If you create a service, select the type as ASC to enable the features of the Application Service Controller for Platform Symphony. An Application Service Controller service can be either stateful or stateless.

Application template

An application template is defined in YAML Ain't Markup Language (YAML) and contains all of the parameters, resources, and outputs that are required to register application instances.

Consumer

A consumer is a unit within the representation of an organizational structure. The structure creates the association between the workload demand and the resource supply.

EGO Service Controller

The EGO service controller (egosc) is the first service that runs on top of the EGO kernel. It functions as a bootstrap mechanism for starting other services in the cluster, and also monitors and recovers other services. It is somewhat analogous to `init` on UNIX systems or Service Control Manager on Windows systems. After the kernel boots, it reads a configuration file to retrieve the list of services to be started. There is one egosc per cluster, and it runs on the master host.

Process information manager (PIM)

PIM collects resource use of the process that runs on the local host.

Platform Management Console (PMC)

The PMC is the web interface to the Application Service Controller for Platform Symphony. The PMC provides a single point of access to manage and monitor your application instances.

Resource

Resources are physical and logical entities that are used by application instances to run. CPU slots are the most important resources.

Resource group

A resource group is a logical group of hosts. A resource group can be specified by resource requirements in terms of operating system, memory, swap space, CPU factor, and so on. Or, it can be explicitly listed by host names.

Service

A service is a self-contained business function that accepts one or more requests and returns one or more responses through a well-defined, and standard interface. The service performs work for a client program. It is a component capable of performing a task, and is identified by a name. Symphony runs services on hosts in the cluster.

The service is the part of your application instance that does the actual calculation. The service encapsulates the business logic.

Service instance

When a service is running on a host, service instances are created. When the service is stopped, there are no longer service instances.

Stateful Application Service Controller service

An Application Service Controller service typically stores data locally to a disk on the host that it runs on. Application Service Controller for Platform Symphony aims to keep the service running on that host, and enables optional decommission of the service to properly handle the data for that service when it is removed.

Stateless Application Service Controller service

This is an Application Service Controller service that does not store data locally on the host that is runs on. A stateless service can be safely restarted on a different host if necessary. By default, all Application Service Controller services are stateless.

3.8.4 Key prerequisites

In order to deploy the Application Service Controller for Platform Symphony, an Platform Symphony Advanced Edition license is required. The following are additional requirements:

- A physical grid computing environment that consists of any of the following:
  - IBM Power Systems
  - IBM PureSystems®
  - IBM System x servers
  - Similar servers from third-party companies
- Cluster nodes pre installed with supported operating environments
- Cluster nodes that are connected via a fast TCP/IP network infrastructure
- Management hosts on the cluster that ideally share a common network file system (enable recovery of grid sessions in case of failure)

Hardware requirements

Platform Symphony v7.1 is supported on Lenovo System x iDataPlex and other rack-based servers as well as non-IBM x64 servers. Also supported are IBM Power Systems servers running PowerLinux operating environments. PowerLinux support is for Big Endian only.

IBM Power System servers running AIX® can integrate with Platform Symphony, but from a client perspective only.

Other platforms include:

- Microsoft Windows 64-bit
- Linux x86-64
- Linux on IBM Power Systems
- Solaris x86-64
- IBM AIX 64-bit: C++ software development kit (SDK) and Java client
- SPARC Solaris 10-64: C++ and Java SDK
- Co-processor harvesting: Client, SDK, and compute nodes

Software requirements

A high-level summary of operating environments supported by Platform Symphony are listed as follows:

- Microsoft Windows Server 2008 SE, 2008 EE, 2008 R2 SE, and 2008 R2 EE (64-bit)
- Windows HPC Server 2008 and 2008 R2 (64-bit)
- Windows Server 2012 Standard and Datacenter, and 2012 R2 Standard and Datacenter (64-bit)
- Windows 7, 8 (64-bit)
- Red Hat Enterprise Linux (RHEL) AS 5, 6, 6.4, 6.5 (x86-64)
- RHEL AS 7 (x86-64)
- CentOS 6 (x86-64)
- SUSE Linux Enterprise Server (SLES) 10 and 11 (x86-64)
- SLES 11 SP2 (x86-64)
- RHEL on IBM POWER6®, POWER7®, and POWER8™
- Oracle Solaris SPARC 10-64 and x86-64 11 (64-bit) (with limitations)
- IBM AIX 7.1 (64-bit) (with limitations)

IBM Knowledge Center: You can find more information in “Supported Platform Symphony and Platform Symphony Developer Edition system configurations” in IBM Knowledge Center:

http://ibm.co/1JjNQed

3.8.5 Application Service Controller application templates

The Application Service Controller for Platform Symphony provides application template samples to help you create your own application templates. The following template samples are available to customize:

- asc_sample_minimal.yaml
- asc_sample.yaml

There are also application template samples available to customize that are specific to the following application frameworks:

- Ambari
- Cassandra
- Cloudera manager
- Hadoop 2.4.1
- Hadoop 2.4.1 with Docker support
- MongoDB
- Spark
- Spark and HDFS with Docker support
- ZooKeeper

All of the application template samples are available in this directory:

```
$EGO_CONFDIR/../../asc/conf/samples
```

Spark and Cassandra solutions are covered in the sections that follow.

Note: To use the application template samples, you must start the Application Service Controller as the root user. Changes in the scripts are required if another user starts the controller.

3.9 Platform Symphony application implementation

As explained in 3.2.2, “Application” on page 30, Platform Symphony service-oriented applications consist of a client application and a service. When the application runs, a session is created that contains a group of tasks. The application profile provides information about the application.

This section provides details about how to deploy Platform Symphony.

3.9.1 Planning for Platform Symphony

This section describes the necessary planning steps.

For examples in this book, we configured a mixed cluster environment with IBM Platform Symphony v7.1 on four virtual machines. We tested Hadoop MapReduce framework with the WordCount application, which is suitable for production use or small-scale application testing.

Chapter 4, “Mixed IBM Power Systems and Intel environment for big data” on page 69 includes a full description of the installation steps.

Components

The following are the essential components:

- Two virtual guests hosted by X3850X5 7145-AC1 running VMware ESXi
- Two virtual guests hosted by IBM PowerLinux 7R2 8246-L2C
- Red Hat Enterprise Linux Server release 6.5 (Santiago) x86, 64-bit
- Apache Hadoop release 1.1.1 x86, 64-bit
- IBM Platform Symphony release 7.1 x86, 64-bit
- Spectrum Scale release 4.1.0 with Fix Pack GPFS_STD-4.1.0.4 x86, 64-bit
- WordCount v1.0
- Oracle Java:
  - jre-6u45-linux-x64 on an Intel-based platform
  - ibm-java-jre-6.0-16.2-linux-ppc64 on IBM PowerLinux

Installation prerequisites

You need to set variables and fulfill the prerequisites before you can start installing the IBM Platform Symphony 7.1:

1. Choose the root operating system account for the installation. This choice provides the flexibility to use different execution accounts for different grid applications.
2. Set the grid administrator operating system account to `egoadmin`. We created this account in Lightweight Directory Access Protocol (LDAP) before we started the installation process:

   ```
   useradd egoadmin
   ```

3. Grant root privileges to the cluster administrator. We set up the cluster and a host:

   ```
   A new cluster <ITSOCluster> has been created. The host <pw4302-l2> is themaster
   host.
   ```

4. If the number of CPUs in the cluster plus the number of client connections to the cluster exceeds 1000, increase the cluster scalability. As root user, run the following command:

   ```
   /opt/ibm/platformsymphony/profile.platform
   ```

5. Add this line to the `/etc/security/limits.conf` file:

   ```
   * hard nofile 6400
   ```

6. Before you start the EGO configuration, connect your IP with your host name in the `/etc/hosts` directory as root to avoid host name errors. It is also best to shut down the `iptables service` to avoid connection failures.
7. Set the variables that are necessary to install correctly:

   ```
   export CLUSTERADMIN=egoadmin
   export CLUSTERNAME=ITSOCluster
   export JAVA_HOME=/usr/java/latest
   export SIMPLIFIEDWEM=N
   ```

8. To run **`egoconfig`** and complete the cluster configuration, log in as `egoadmin`. The cluster uses configuration files under the `EGO_CONFDIR/.../` directory. The value of the *`EGO_CONFDIR`* environment variable changes if the cluster keeps configuration files on a shared file system. When your documentation refers to this environment variable, substitute the correct directory.

9. Configure the mixed cluster environment for Platform Symphony.

   On PowerKVM and Intel based Linux platforms. We decided to configure the first host as the master host and the second host as the master candidate for failover. We do not set the cluster to handle failover, so the cluster uses configuration files in the installation directory:

   `EGO_CONFDIR=EGO_TOP/kernel/conf`

10. Enable automatic startup, grant root privileges to the cluster administrator, and start EGO.

We used the following settings for the Platform Symphony installation:

- Workload Execution Mode (WEM): **Advanced**
- Cluster Administrator: `egoadmin`
- Cluster Name: `ITSOCluster`
- Installation Directory: `/opt/ibm/platformsymphony`
- Connection Base Port: `7869`

After installation, you can run **`egoconfig setbaseport`** on every host in the cluster to change the ports used by the cluster. If you want to add more compute hosts, follow the installation instructions for IBM PowerKVM or for Intel based Linux in Platform Symphony documentation on IBM Knowledge Center:

`http://ibm.co/1Ncoeh9`

Ports

The default base port used by Platform Symphony is 7869. We suggest that you use the default value unless you have systems that run other services using that port. Remember that Platform Symphony requires seven consecutive ports that start from the base port, for example, 7869 - 7875. Ensure that all ports in that range are available before you start installation.

Important: On all hosts in the cluster, you must have the same set of ports available.

If you need to set a different base port, use the **`BASEPORT`** environment variable when you define the cluster properties for installation. For example, to use 17869 as the base port, define `BASEPORT=17869` in the `install.config` file.

Platform Symphony also requires more ports for services and daemons. Table 3-3 describes the required ports for each service.

Table 3-3 Additional port requirements

Service	Required ports
Web server	8080, 8005, and 8009
Service director	53
Web service	9090
Loader controller	4046
Derby database	1527

Workload execution mode

During the installation, it is necessary to decide whether a single user (non-root) is the primary user of the grid. If so, use the Simplified Workload Execution Mode (WEM) approach where the Platform Symphony applications run under one user account.

Otherwise, to provide more flexibility and to allow different applications and users to run applications from the grid, use the Advanced WEM approach. Platform Symphony applications run under the workload execution account of the consumer, which is a configurable account. Different consumers can have different workload execution accounts.

Note: Do not let the Advanced configuration name confuse you during the installation, because the default values from Platform Symphony can run most workloads.

Cluster name

You must customize the installation if you want to specify your own unique cluster name. Do not use a valid host name as the cluster name.

Important: The cluster name is permanent; you cannot change it after you install it.

To specify the cluster name and not use cluster1, set the environment variable with `CLUSTERNAME=<Name>`.

Multi-head installations

Platform Symphony requires a configuration parameter named `OVERWRITE_EGO_CONFIGURATION`. The default is `No`. If this parameter is set to `Yes`, the defulat Platform Symphony configuration overwrites the EGO configuration. For example, it overwrites EGO `ConsumerTrees.xml`, adds `sd.xml` in the EGO service `conf` directory and overwrites the EGO Derby DB data files.

If you plan a multi-head cluster, which is a cluster that runs both Platform Symphony and IBM Platform Load Sharing Facility (LSF), it is acceptable for IBM Platform LSF and Platform Symphony workloads to share EGO resources in the cluster. In that case, avoid overwriting the EGO configuration.

The environment that is planned in this section is single-headed, so we ensure that the variable `OVERWRITE_EGO_CONFIGURATION` is set to `Yes`.

Software packages

Check that you have all of the required software packages and entitlement files available (see Table 3-4).

Table 3-4 Software packages and entitlement file list

Type	File name
Platform Symphony package	`symSetup7.1.0_lnx26-lib23-x64.bin`
EGO package	`ego-lnx26-lib23-x64-3.1.0.rpm`
SOAM (SOA middleware) package	`soam-lnx26-lib23-x64-7.1.0.rpm`

3.9.2 Accessing the Platform Symphony management console

The Platform Symphony console is usually on the same host, if you follow the installation recommendations described in the previous section. Port 8080 is the default, so you should be able to log into the Platform Symphony management console with the following URL:

http://*<master-host>*:8080/platform

The default user administrator login and password for Platform Symphony are both `admin`. Figure 3-8 shows the IBM Platform Symphony v7.1 login screen.

Figure 3-8 IBM Platform Symphony login screen

In production clusters, there are usually multiple Platform Symphony management hosts. Setting this up is covered in Chapter 4, “Mixed IBM Power Systems and Intel environment for big data” on page 69. You can also review “Installing a Platform Symphony cluster” in IBM Knowledge Center:

http://ibm.co/1cxDk4n

If you are having trouble connecting to the Platform Symphony web console, use the following command to see details about the web service:

```
egosh service view WEBGUI
```

The WEBGUI services should be started automatically by EGO, but if it necessary to start or stop the service, use the following command:

```
egosh service start WEBGUI
egosh service stop WEBGUI
egosh logon
```

Enter `admin` as the username and the password when prompted:

The WEBGUI service is implemented using Apache TomCat. If there are problems with the WEBGUI, inspect the logs for information about what might be wrong with the service by using the following command:

```
${EGO_TOP}/gui/logs/catalina.out
```

If you cannot connect to the Platform Symphony console, the cause might be your firewall configuration. You can disable your firewall temporarily to see if this is the cause by using following command:

```
service iptables stop
```

After a user logs in to the Platform Symphony console on port 8080, the main dashboard shown in Figure 3-9 opens. This view is mostly used to monitor the high-level status of the various applications and tenants in the Platform Symphony cluster.

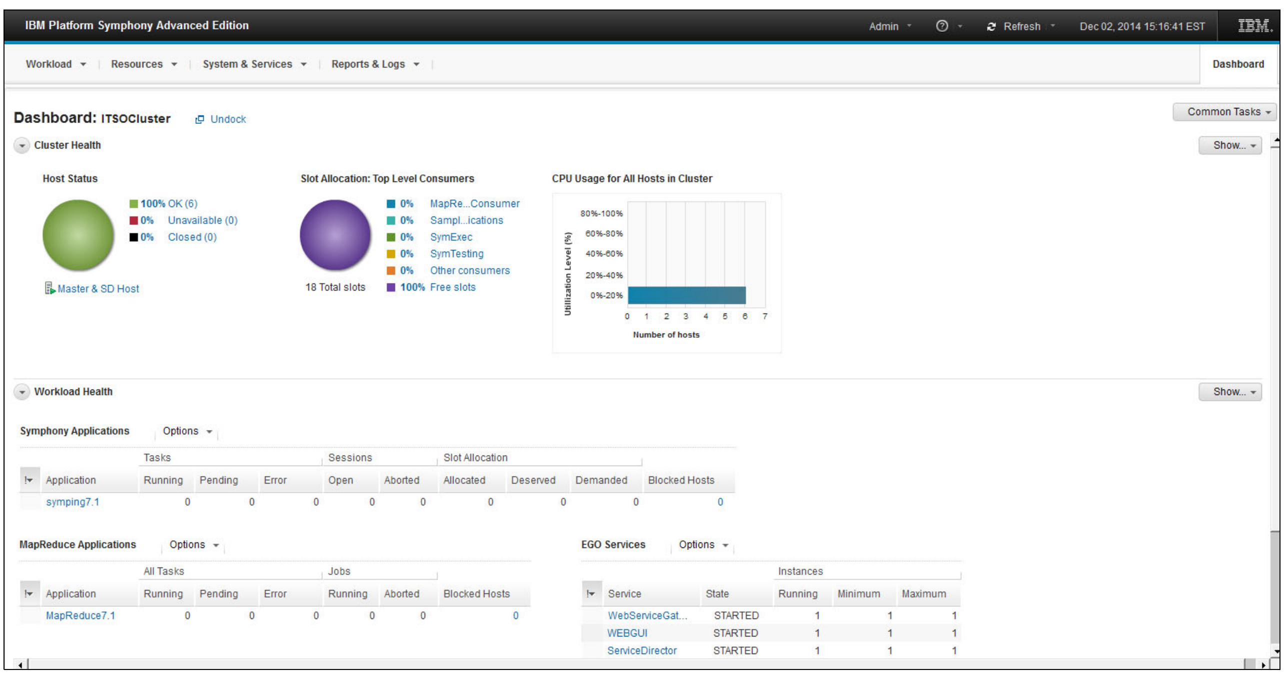


Figure 3-9 Platform Symphony dashboard view after login

3.9.3 Configuring a cluster for multitenancy

Platform Symphony has two workload execution modes:

- Simple Workload Execution Mode
- Advanced Workload Execution Mode

This is normally an installation option with Platform Symphony. The Platform Symphony Enterprise Edition installation automatically installs Platform Symphony in Advanced Workload Execution Mode. This term is frequently abbreviated as WEM in the Platform Symphony documentation. In advanced workload execution mode, core Platform Symphony services run as root so that as application administrators are able to control the user ID that clustered applications run under. Platform Symphony is frequently deployed in secure environments, where these capabilities are important.

Configuring OS groups for the multitenant environment

Those who are using Platform Symphony (both named users and the user IDs that applications run under, via impersonation), these IDs need to be part of the operating system (OS) group that owns the Platform Symphony installation.

Users and security

To allow users to use resources when running their applications in a managed way, Platform Symphony implements a hierarchical model of consumers. This allows association of users and roles with applications and with grid resources. Policies for the distribution of resources among multiple applications run by different users can be configured to share the resources in the grid. MapReduce applications and other non-MapReduce applications, such as standard SOA compute-intensive applications, within Platform Symphony can use the same infrastructure.

A multi-head installation of both Platform LSF and Platform Symphony is supported. This installation allows batch jobs from Platform LSF and compute-intensive and data-intensive applications from Platform Symphony to share the hardware grid infrastructure. A security model is enforced for the authentication and authorization of various users to the entitled applications and to isolate them when they try to access the environment.

You can create user accounts within the Platform Symphony environment, as shown in Figure 3-10. Then, assign to the user accounts predefined or user created roles. User accounts include optional contact information, a name, and a password.

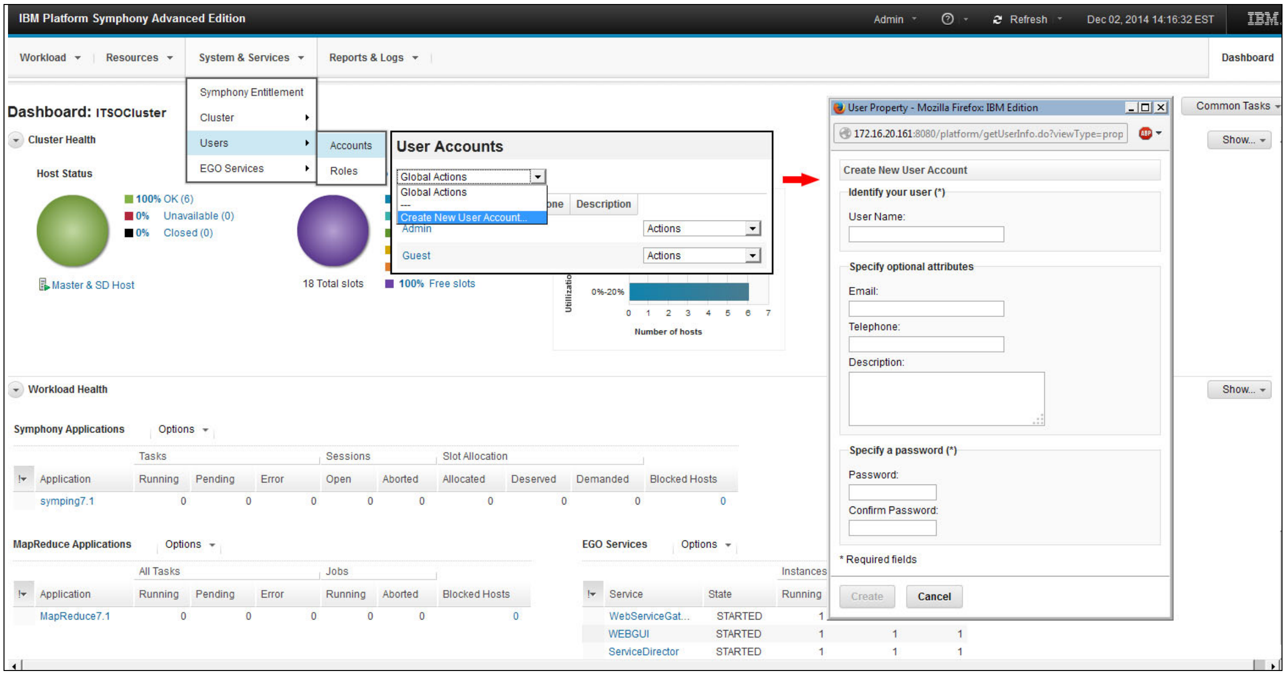


Figure 3-10   Create user accounts

Platform Symphony has four predefined user roles that can be assigned to a user account:

- Cluster administrator

  A user with this role can perform any administrative or workload-related task, and has access to all areas of the Platform Management Console and to all actions within it.

- Cluster administrator (read only)

  This user role allows read-only access to any cluster information, but cannot perform any add, delete, or change action.

- Consumer administrator

  Users with this role are assigned to a top-level consumer in the consumer hierarchy, and can administer all subconsumers in that branch of the tree.

- Consumer user

  Consumer users are assigned to individual consumers on the tree, and have access and control only over their own workload units.

To submit a workload for an enabled application, a user must have appropriate roles and permissions assigned. When more roles are added to a user account, the permissions are merged. To configure such a setup, you need an Administrator role with the correct permissions.

Sharing resources

An application can be used only after it is registered and enabled. You can only register an application as a leaf consumer (a consumer that has no subconsumers). Only one application can be enabled per consumer. Before you can register an application, you must create at least one consumer and deploy the service package of the application to the intended consumer. You can deploy the service package to a non-leaf consumer so that all applications registered to child leaf consumers are able to share the service package. A service package is created that puts all developed and compiled service files and any dependent files associated with the service in a package.

Resource distribution plan

In this step, you relate the resources themselves to the consumer tree and introduce the resource distribution plan that details how the cluster resources are allocated among consumers. The resource orchestrator distributes the resources at each scheduling cycle according to this resource distribution plan. The resource plan takes into account the differences between consumers and their needs, resource properties, and various policies about consumer ranking or prioritization when allocating resources.

You must initially assign bulk resources to consumers in the form of resource groups to simplify their management. You can change this assignment later.

Resource groups are logical groups of hosts. A *host* in a resource group is characterized by a number of slots. The number of slots is a variable parameter. When you choose a value for it, the value must express the degree of specific workload that the host is able to serve. A typical slot assignment is the allocation of one slot per processor core.

After it is created, a resource group can be added to each top-level consumer to make it available to all subconsumers. Figure 3-11 shows an example of a consumer tree with all of its top-level consumers and their assigned resource groups and users. Platform Symphony provides a default top-level consumer, MapReduceConsumer, and a leaf-consumer.

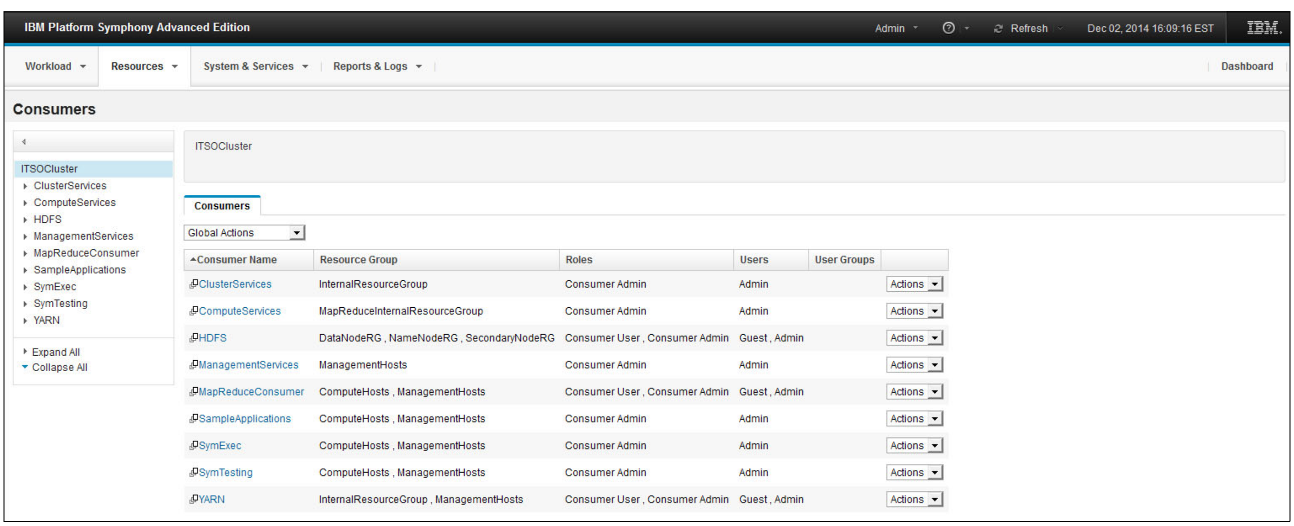


Figure 3-11 Platform Symphony consumer tree

There are several concepts used within a resource distribution plan:

Ownership The guaranteed allocation of a minimum number of resources to a consumer.

Borrowing and lending The temporary allocation of owned resources from a lending consumer to a consumer with an unsatisfied demand.

Sharing The temporary allocation of unowned resources from a shared pool to a consumer with an unsatisfied demand.

Reclaiming This defines the criteria under which the lender reclaims its owned resources from borrowers. The policy can specify a grace period before starting the resource reclaim, or the policy can specify to stop any running workload and reclaim the resources immediately.

Rank The order in which policies are applied to consumers. Rank also determines the order in which the distribution of resources is processed. The highest-ranking consumer receives resources first, borrows resources first, and returns borrowed resources last.

This is shown in Figure 3-12.

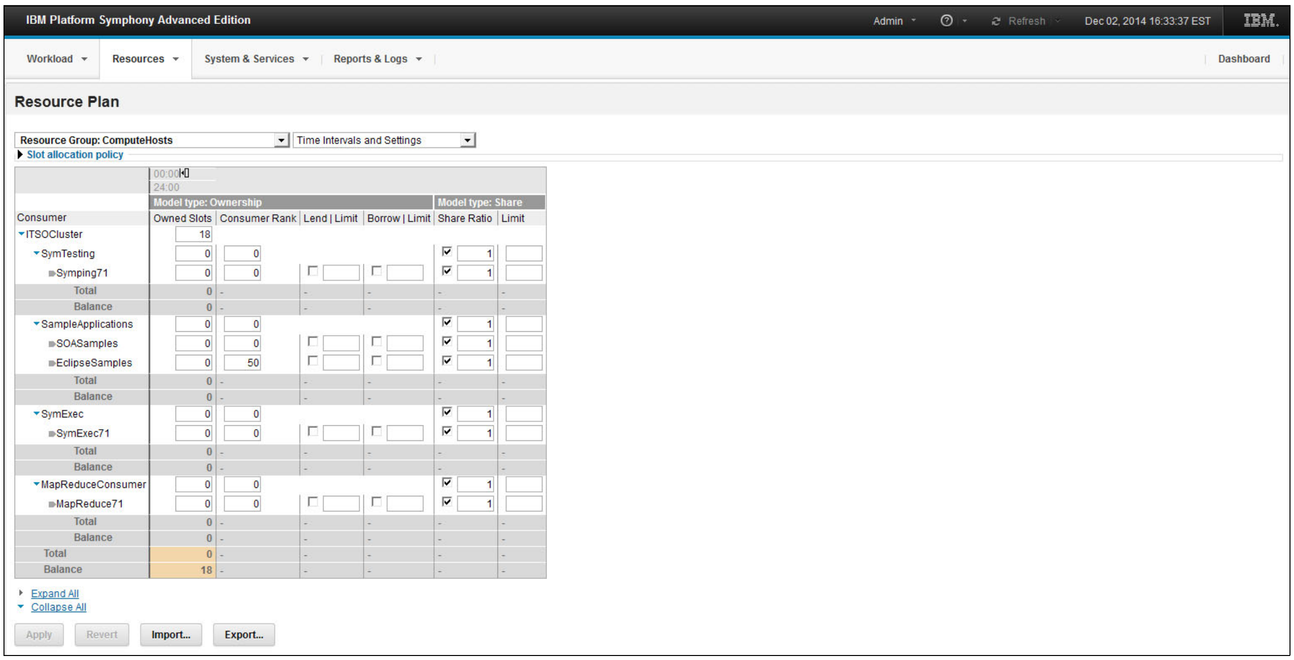


Figure 3-12 Resource plan

The first allocation priority is to satisfy each consumer’s reserved ownership. Remaining resources are then allocated to consumers that still have demands. Unused owned resources from consumers that can lend them are allocated to demanding consumers that are entitled to borrow. The resource orchestrator then allocates the unowned resources from the shared pool to consumers with unsatisfied demands that are entitled to the type of resources.

The resources from the “family” pool (any unowned resources within a particular branch in the consumer tree) are allocated first. After the family pool is exhausted, the system distributes resources from other branches in the consumer tree. The free resources in the shared pools are distributed to competing consumers according to their configured share ratio. A consumer that still has an unsatisfied demand and has loaned resources reclaims them at this stage.

Owned resources are reclaimed first, followed by the entitled resources from the shared pool that is currently used by consumers with a smaller share ratio. This is the default behavior. The default behavior can be changed so that owned resources are recalled first before trying to borrow from other consumers.

The resource orchestrator updates the resource information at a frequency cycle that is determined by `EGO_RESOURCE_UPDATE_INTERVAL` in the `ego.conf` file. Its default value is 60 seconds. At each cycle, the resource orchestrator detects any newly added resource or unavailable resource in the cluster and any changes in workload indexes for the running jobs.

Each resource group must have its own plan. Also, you can define different resource plans for distinct time intervals of the day. This enables you to better adapt them to workload patterns. At the time interval boundary, the plan change might determine important resource reclaiming.

Enabling Platform Symphony repository services

By default, when Platform Symphony is installed, the repository service in Platform Symphony is disabled. The function of the repository service is to store the application services and distribute the code that implements services dynamically to service instances on the cluster.

By default, the MapReduce framework in Platform Symphony distributes the application service code (specifically, the application logic that implements the task tracker function and JAR files that implement map and reduce logic) by copying them to Hadoop Distributed File System (HDFS) with a high block replication factor so that the files will be accessible on all nodes.

If you are planning to add and remove application profiles in Platform Symphony or consumers, start the Platform Symphony repository service. Otherwise, you will encounter errors, because some of these services assume that the repository service in Platform Symphony is running. Using the web interface, select **System & Services** → **Ego Services** → **Services**. This System Services view, shown in Figure 3-13, lists system services that EGO is managing.

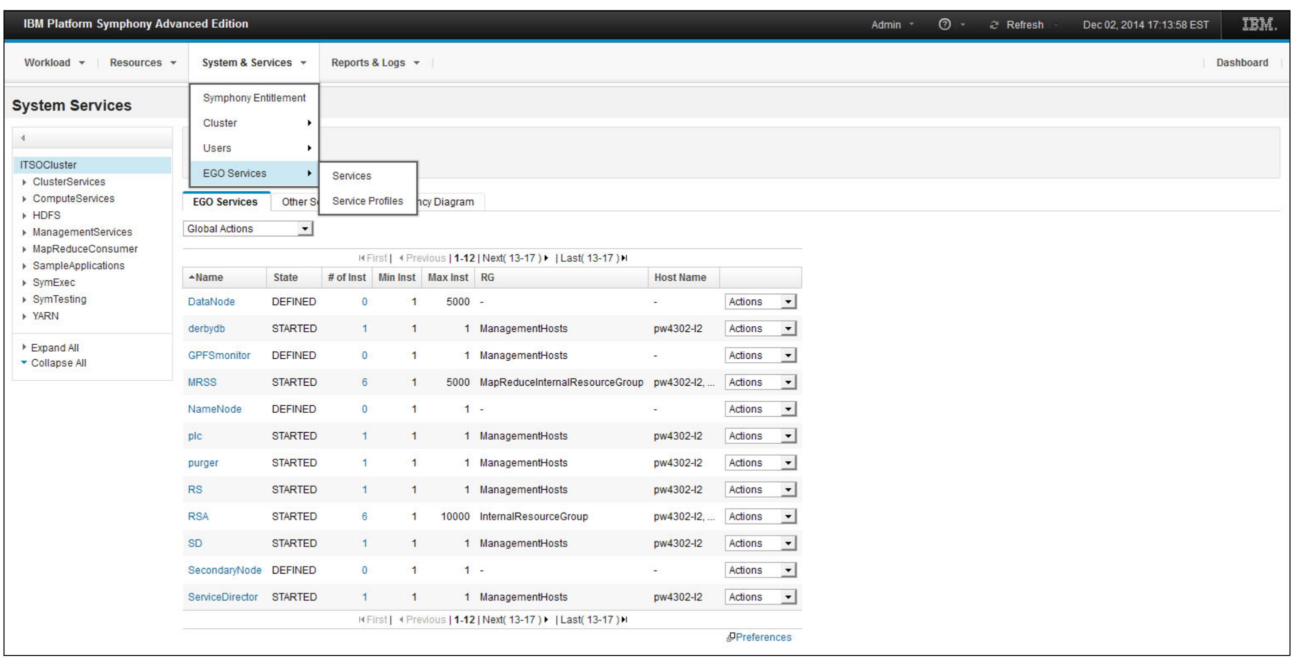


Figure 3-13 System services

3.9.4 Adding a new application or tenant

Fundamental to the design of open source Hadoop is the idea that there is only a single instance of the Hadoop cluster. However, Platform Symphony supports multiple applications sharing the same cluster. Platform Symphony is also flexible enough to support multiple instances of an application environment.

There are several kinds of tenants that you might want to add:

- A native Platform Symphony application written with the Platform Symphony APIs
- A batch-oriented workload (when Platform LSF is installed as an add-on to Platform Symphony)
- A distinct Hadoop MapReduce environment
- A separate Hadoop MapReduce application instance that shares resources between applications but shares the same Hadoop binaries and file system instance
- Third-party applications

To add an application profile, from the pull down menu select **Workload** → **MapReduce** → **Application Profile**. The Add Application screen opens as shown in Figure 3-14.

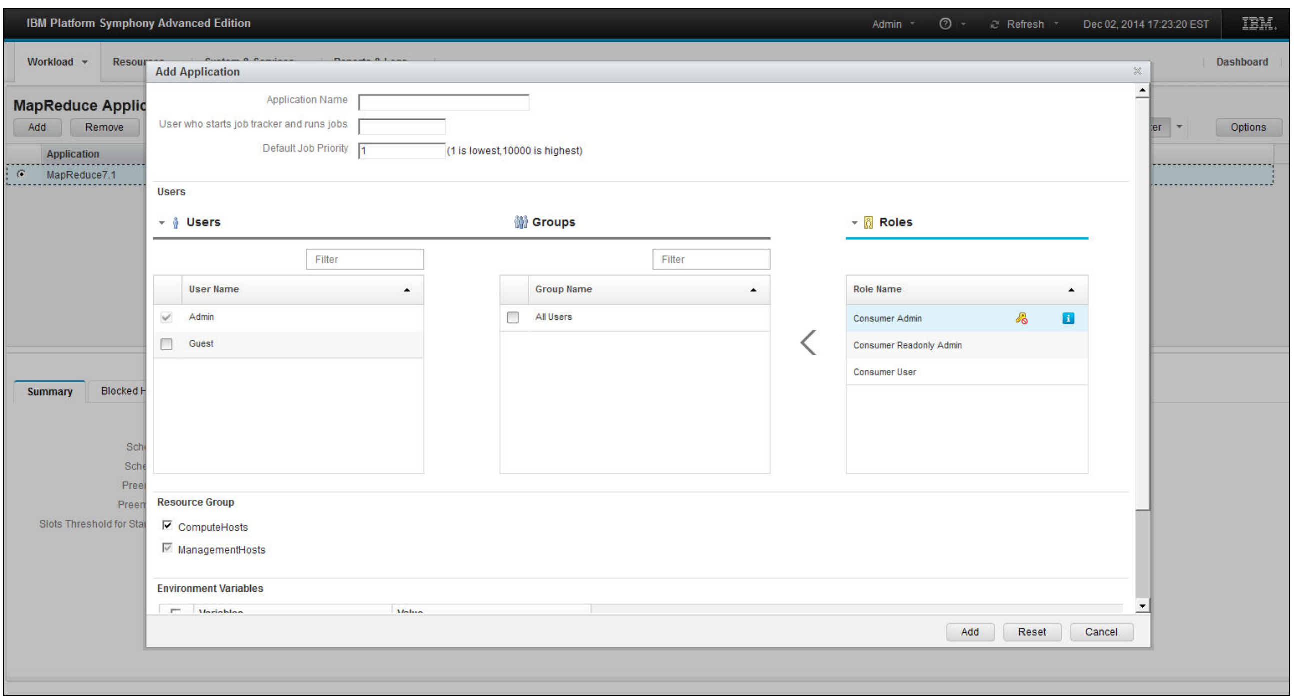


Figure 3-14 Add Application screen

There is already an application profile defined for MapReduce. This is installed automatically with Platform Symphony. To add a new application profile to support a new tenant, click the *Add* button.

The following parameters need to be filled in:

- Application name.
- The user-ID that starts the job tracker and runs jobs. This is the impersonation feature. You must define what operating system ID the application has to run under.

- Platform Symphony has 10,000 priority levels. By default, you can submit your application jobs as having a low priority and later increase the level if necessary.
- Configure user accounts that have access to this application. Include all users in specialized group access to the application, along with named operating system and Platform Symphony users.

> **Note:** To choose any operating system ID, it is mandatory to run Advanced Workload Execution Mode (WEM), which can be defined when you install Platform Symphony. If using Simplified WEM, the workload runs using the grid manager user account.

Based on this information, Platform Symphony adds an application with a set of reasonable defaults for a Hadoop MapReduce job.

The next step is to edit the configuration of the tenant as necessary to suit the unique needs of the application. Select **Workload** → **MapReduce** → **Application Profiles**, and then define as many separate applications as you want.

3.9.5 Configuring application properties

When applications profiles are created for each new application, a default template is used to represent reasonable settings for a MapReduce workload. The next step is to configure application profiles to meet the unique requirements of each application workload.

Application profiles are covered in detail in “Managing the Platform Symphony cluster and application” in IBM Knowledge Center:

`http://ibm.co/1MMzLnY`

To configure application properties for Apache Sqoop, modify the application profile by selecting: **Workload** → **MapReduce** → **Application Profiles** from the top menu on the MapReduce applications screen. Select the application profile definition for the application created earlier, and select **Modify**.

A new window opens that allows detailed settings for the application to be changed. The web interface affects the application service profile definitions that are stored in the `$EGO_TOP/data/soam/profiles` directory on the Platform Symphony master host. Enabled profiles reside in a subdirectory called `enabled`, and disabled profiles reside in a directory called `disabled`.

The first tab in the interface, called Application Profile, is where you can adjust application profile settings. Use the second tab, Users, to modify the users and groups that have access to the application profile.

Remember the following information application profiles:

- Application profile names must be unique.
- An application profile can be associated with only a single consumer.
- In the consumer tree, MapReduce applications are placed under the MapReduceConsumer tree by default.

The application profile can be viewed in an Advanced Configuration, a Basic Configuration, or in a Dynamic Configuration Update mode.

The General settings area includes settings such as where metadata associated with jobs and job history are stored, the default service definition to use (MapReduce for MapReduce applications), and resource requirements.

The Platform Symphony application profile definition provides precise control over how MapReduce workloads run, and this is useful to advanced users.

Another nice feature of Platform Symphony is the execution logic. It is provisioned dynamically, so slots are interchangeable between mappers and reducers. Settings allow this to be configured along with preferences for default ratios between mappers and reducers and precise configuration on a per-resource-group basis.

Platform Symphony allows multiple service definitions to exist for each application, and the service definition section provides granular control over this capability. This is useful for applications written to Platform Symphony's native APIs and might be useful for Hadoop developers. Also, it has already implemented a service called RunMapReduce, which is started by service-instance managers to handle MapReduce workloads. The process of starting this service is automatic for the MapReduce service.

Heterogeneous applications support

Platform Symphony supports heterogeneous applications. It does not matter whether application clients or services are written in C/C++, Java, scripting languages, or even C# in Microsoft .NET environments. The versatility to handle all types of workloads is what makes Platform Symphony powerful as a multitenant environment.

Another unique capability that Platform Symphony brings to Hadoop is *recoverable sessions*. This concept does not exist in open source Hadoop where the job tracker is implemented in a simple way. In standard Hadoop, if the JobTracker fails at run time, the job must be restarted.

However, the Platform Symphony SOAM middleware has long supported journaling transactions, so Hadoop MapReduce jobs become inherently recoverable. If the software service that is running the JobTracker logic fails (and restarts on the same host or a different host), the Platform Symphony job can recover and resume from where it left off. This is a major advantage if you have long-running Hadoop jobs that need to complete within specific batch windows.

This and other configuration capabilities are very important for specific workloads. As another example, if you have execution logic where the reducer is multithreaded, you can control the ratio of reducer services to slots, thereby giving a reducer multiple slots if it can take advantage of them.

3.9.6 Associating applications with consumers

In the Platform Symphony architecture, resources are not actually allocated to applications directly. They are allocated to consumer definitions which, in turn, map to applications.

This is an important distinction because, although an application space is essentially *flat* (if you have multiple applications and versions of applications of different types), the structure of consumers is usually hierarchical. This is because most organizational structures are hierarchical, for example:

- A bank might have several lines of business, each with various departments or application groups
- A service provider might have multiple tenant customers, and might provide different application services for each tenant

- A government agency might have different divisions, each running different applications with a particular need to segment data access

Platform Symphony allows consumer trees to be setup in flexible ways to accommodate the needs of almost any organization. A key concept to understand is that the leaf nodes of consumer trees are linked to the application definitions.

To view consumer definitions, from the MapReduce screen in Platform Symphony select **Resources** → **Resource Planning** → **Consumers**. This is the interface that is used to manage the consumer tree.

Setting up the consumer tree is reasonably straightforward. The left panel controls where you are on the tree, and the right side of the interface allows you to perform operations relative to that segment on the tree. Notice the hierarchical notion of consumers in Platform Symphony.

Advanced users might find it easier to manually edit the consumer tree. Platform Symphony stores consumer tree definitions in an XML file:

```
$EGO_TOP/kernel/conf in the file ConsumerTrees.xml
```

If you manually edit this file, restart EGO services to synchronize the web-based view with the actual contents of the XML files where these settings are persisted.

After editing the `ConsumerTrees.xml` file, while logged in as the cluster administrator, stop and restart EGO services to make sure that changes are reflected in the Platform Symphony console.

3.9.7 Summary

This chapter described a customer use case involving a multitenant implementation of Platform Symphony that permits the following:

- Concurrent execution of different Hadoop applications (including different versions of code) in the same physical cluster.
- Dynamic sharing of resources between tenants in a way to maximize performance and resource use while respecting individual SLAs.
- Support for applications other than Hadoop MapReduce to maximize flexibility and allow capital investments to be re-purposed for multiple requirements.
- Security isolation between tenants, removing a major barrier to sharing in many commercial organizations.

These advances are significant. Although Hadoop is advancing and growing, open source solutions are still far away from commercial distributions that are real multitenancy offerings. In fact, Platform Symphony brings up practical solutions for supporting multiple workloads in a shared infrastructure.

The economic arguments in favor of resource sharing are compelling. Analytic applications are increasingly comprised of multiple software components that rely on distributed services. Rather than deploying separate application infrastructure silos, Platform Symphony provides the option to consolidate these different application instances on a common foundation, thereby increasing infrastructure use, boosting service levels, and helping significantly in reducing costs.

3.10 Application Service Controller in a big data solution

This section describes the possibility of a complete big data environment using Platform Symphony Advanced Edition v7.1 and one add-on feature, called Application Service Controller (ASC) as a complement to Adaptive MapReduce.

Since IBM Platform Symphony version 6.1 was released, an Apache Hadoop-compatible MapReduce implementation optimized for low latency was included in Platform Symphony Advanced Edition, bringing to the market reliability and resource sharing that has been demonstrated in an audited benchmark delivering on average four times the performance of open source Hadoop.

IBM Platform Symphony MapReduce solution

IBM Platform Symphony MapReduce is an enterprise-class distributed runtime engine that integrates with open source and commercial Hadoop-based applications, for example, IBM InfoSphere BigInsights and Cloudera CDH3. The Platform Symphony MapReduce framework addresses several problems that typical Hadoop clusters experience. With it, you can incorporate robust high-availability (HA) features, enhanced performance during job initiation, sophisticated scheduling, and real-time resource monitoring.

3.10.1 Hadoop implementations in IBM Technology

In fact, Hadoop is a standard choice for large-scale data processing across almost every industry and enterprise, with numerous vendors providing Hadoop distributions in conjunction with enterprise-grade support services.

In 2009, IBM's Information Management division created a value-added Hadoop implementation called InfoSphere BigInsights that includes Apache Hadoop and various other open source components, as well as IBM-developed tools aimed at simplifying management, application development, and data integration. Although InfoSphere BigInsights customers continue to use the Hadoop MapReduce API and higher-level tools such as PIG, Apache HBase, and HIVE, they have the option of using proprietary components in addition to or in place of the open source Hadoop components.

Hadoop scales computation and storage across inexpensive commodity servers and allows other applications to run on top of both of these. Spark is one of these applications. Spark runs on top of existing Hadoop clusters to provide enhanced and additional functions.

Although Hadoop is effective for storing vast amounts of data inexpensively, the computations that it enables with MapReduce are quite limited. MapReduce can execute only simple computations and uses a high-latency batch model.

Adaptive MapReduce re-implements the standard Hadoop job tracker, task tracker, and shuffle services on a low-latency grid middleware implementation provided by IBM Platform Computing. Adaptive MapReduce provides even better production-oriented benefits than Hadoop's grid management and scheduling components. One of those benefits is superior performance.

Comparison of IBM InfoSphere BigInsights Enterprise Edition with Adaptive MapReduce and Apache Hadoop

In an audited benchmark conducted on October 2013 by STAC, the Securities Technology Analysis Center, InfoSphere BigInsights for Hadoop was found to deliver an approximate 4x performance gain on average over open source.

In jobs derived from production Hadoop traces, InfoSphere BigInsights accelerated Hadoop by an average of approximately 4x. The speed advantage of InfoSphere BigInsights was most closely related to the shuffle size. Much of the InfoSphere BigInsights advantage appears to be due to better scheduling latency.

In a test of scheduling speed, the InfoSphere BigInsights configuration outperformed the Hadoop configuration by approximately 11x in warm runs. Default settings for the Hadoop core and for InfoSphere BigInsights were used.

Nevertheless, it is possible that different settings for Hadoop or InfoSphere BigInsights could achieve different results.

> **Note:** The full report, titled *STAC Report: IBM BigInsights vs Apache Hadoop, using SWIM* (document IML14386USEN), can be downloaded in PDF format from the following web page:
>
> http://ibm.co/1FPUWQl

3.10.2 Adding Application Service Controller to improve a big data cluster

Typically, stand-alone Hadoop clusters, which are deployed as resource silos, so they cannot function in a shared services model. They cannot host different workload types, users, and applications.

IBM Platform Symphony v7.1 revolutionizes big data environments by providing the possibility of sharing an infrastructure across many application frameworks, with elasticity and easy control of many popular big data applications. The Application Service Controller for Platform Symphony add-on can integrate with many distributed application frameworks to manage and run them on a scalable and shared grid.

ASC allows you to deploy, run, and manage complex, long-running application instances in the Platform Symphony cluster. These can be application servers, InfoSphere BigInsights instances, Spark, Cassandra, MongoDB, HBase, and so on. You can monitor and manage application instances and drill down to manage the related services and service instances.

You can also have your application instances running in Docker containers with tools provided by the Application Service Controller. For the latest set of application framework integrations and templates that can be used to improve your daily work, see the IBM Platform Application Service Controller page in IBM Bluemix DevOps Services on the Jazz™.net website:

https://hub.jazz.net/user/ibmasc

3.10.3 Advantages of Spark technology

Spark Core is the underlying general execution engine for the Spark platform that all other functions are built on. It provides in-memory computing capabilities to deliver speed, a generalized execution model to support a wide variety of applications, and Java, Scala, and Python APIs for ease of development.

Spark provides simple and easy-to-understand programming APIs that can be used to build applications at a rapid pace in Java, Python, or Scala. Data scientists and developers alike can benefit from Spark by building rapid prototypes and workflows that reuse code across batch, interactive, and streaming workloads. For instance, users can load tables in Spark programs by using Shark, call machine learning library routines in graph processing, or use the same code for batch and stream processing.

Spark is not a direct replacement for the Hadoop stack, because all of the development already done for the MapReduce framework can continue to run. The Spark technology enhances data managers' experiences and enriches the capabilities of analysis. From the outset, Spark was designed to read and write data from and to HDFS, as well as other storage systems. Hadoop users can enrich their processing capabilities by combining Spark with Hadoop MapReduce, HBase, and other existing big data frameworks by using Platform Symphony with the Application Service Controller.

Note: An IBM Application Service Controller (ASC) application template to quickly deploy a Spark cluster is available at the following website:

https://hub.jazz.net/project/ibmasc/asc-spark-docker/overview

To create Spark with HDFS, you can find the template at the following website:

https://hub.jazz.net/project/ibmasc/asc-spark-hdfs-docker/overview

3.11 Application Service Controller as the attachment for a cloud-native framework: Cassandra

DataGrid solutions are still in use to optimize data distribution in HPC environments, because NoSQL solutions are being used mostly as inbound/outbound data stores for the core compute engines. These solutions are likely to progressively start to override the DataGrid market, mainly for cost reasons. However, it has become increasingly complex to maintain these two different technologies that are dedicated to data management.

NoSQL solutions are starting to be adopted across the board in investment banking either using MongoDB, Cassandra or HDFS/Hadoop as part of their compute stack.

The Application Service Controller uses proven technologies that are widely deployed at scale in some of the world's largest production clusters to enable increased asset use and improved application performance. It is designed to be flexible and to accommodate distributed cloud-native frameworks, such as Hadoop, Cassandra, and MongoDB.

Apache Cassandra is a massively scalable open source NoSQL database. Cassandra is a good choice for managing large amounts of data across multiple data centers and the cloud. It offers continuous availability, linear scalability, and operational simplicity across many commodity servers with no single point of failure, along with a powerful data model designed for maximum flexibility and fast response times.

Cassandra has a *masterless* architecture, meaning that all nodes are the same. It provides automatic data distribution across all nodes that participate in database cluster. There is nothing programmatic that a developer or administrator needs to do or code to distribute data across a cluster, because data is transparently partitioned across all nodes in a cluster.

Cassandra also features customizable replication. Therefore, if any node in a cluster goes down, one or more copies of that node's data is still available on other machines in the cluster. Replication can be configured to work across one data center, many data centers, and multiple cloud availability zones.

3.11.1 Cassandra architecture

Cassandra handles big data workloads across multiple nodes with no single point of failure. Cassandra addresses the problem of failures by employing a peer-to-peer distributed system across homogeneous nodes where data is distributed among all nodes in the cluster. Each node exchanges information across the cluster every second. A sequentially written commit log on each node captures write activity to ensure data durability. Data is then indexed and written to an in-memory structure, called a *memtable*, which resembles a write-back cache.

Cassandra is a row-oriented database. Its architecture allows any authorized user to connect to any node in any data center and access data using the CQL language. For ease of use, CQL uses a syntax similar to SQL. From the CQL perspective, the database consists of tables. Typically, a cluster has one keyspace per application. Developers can access CQL through *cqlsh* and access via drivers for application languages.

Client read or write requests can be sent to any node in the cluster. When a client connects to a node with a request, that node serves as the coordinator for that particular client operation. The coordinator acts as a proxy between the client application and the nodes that own the data being requested. The coordinator determines which nodes in the ring should get the request based on how the cluster is configured.

Cassandra supplies linear scalability, meaning that capacity can be added simply by adding new nodes online. This can be easily accomplished using Platform Symphony with ASC because deploying a new node is merely a matter of few steps. Or, you can schedule automated deployments in case You can also create a schedule to automate a new node deployment if more nodes are needed due to performance issues.

3.11.2 Cassandra and multitenancy

Most users of Cassandra create a cluster for each application or related set of applications, because that is much simpler to tune and troubleshoot. Cassandra's development team has made it possible to support more multitenant capabilities, such as scheduling and authorization. However the most common path is single-tenant. You can bypass this problem by using Cassandra as an application controlled by the Application Service Controller.

> **Note:** Companies that run their applications on Apache Cassandra have achieved results that have directly improved their business. Read the following web page to learn how businesses have successfully deployed Apache Cassandra software in their environments:
>
> `http://planetcassandra.org/apache-cassandra-use-cases/`

3.12 Summary

For the most recent information about Platform Symphony, see IBM Knowledge Center:

`http://ibm.co/1eI6wY9`

See what's new and what has changed in IBM Platform Symphony v7.1:

- What's new

  `http://ibm.co/1QJaBHg`

- What's changed

  `http://ibm.co/1eOS39i`

For developers: To learn about what is new, what has changed, known challenges, and documentation updates for Platform Symphony and Platform Symphony Developer Edition, see the following website:

http://ibm.co/1FHeEyk

4

Mixed IBM Power Systems and Intel environment for big data

In this chapter we build a mixed cluster environment with IBM Platform Symphony on four virtual machines and test Hadoop MapReduce framework with WordCount.

This chapter includes the following sections:

4.1 System components and default settings in the test environment

Our solution works with the following system components:

- Four virtual guests hosted by X3850X5 7145-AC1 running VMware ESXi
- Two virtual guests hosted by IBM PowerLinux 7R2 8246-L2C
- Each machine has one 100 GB system disk, one 1 GB disk for IBM Spectrum Scale metadata, and one 10 GB disk for Spectrum Scale data
- Red Hat Enterprise Linux Server release 6.5, 64-bit
- Apache Hadoop release 1.1.1
- IBM Platform Symphony release 7.1 64-bit
- IBM Spectrum Scale standard edition (GPFS) release 4.1.0, 64-bit with Fix Pack GPFS_STD-4.1.0.1
- IBM Java SDK 7, 64-bit

The following are the default settings on the machines:

- Set variables and install prerequisite software before you install Platform Symphony. You can see prerequisites in Example 4-3, Example 4-8, and Example 4-10.
- If you have Platform Symphony already installed, uninstall it, and then remove its directory from `/opt/ibm/platformsymphony`.
- Shut down the `iptables` service to avoid connection failures on Master, Management, and Compute hosts, and set the SElinux to *disabled*.
- Use the `/etc/hosts` file, as shown in Example 4-1.

Example 4-1 /etc/hosts file on all nodes

```
[root@pw4302-l2 ~]# cat /etc/hosts
172.16.20.161pw4302-l2
172.16.20.160pw4302-l1
172.16.20.164pw4302-l5
172.16.20.165   pw4302-l6
172.16.21.162symphony1
172.16.21.163symphony2
127.0.0.1localhost
[root@pw4302-l2 ~]#
```

- You need Oracle Java installed and exported according to your environment. In this scenario and during the build, use the profile shown in Example 4-2.

Example 4-2 Profile used during the build

```
[root@pw4302-l2 ~]# cat /etc/profile|tail -4
export PATH=$PATH:/usr/lpp/mmfs/bin
export HADOOP_HOME=/usr/share/hadoop
export JAVA_HOME=/usr/ibm_java_7
export PATH=$PATH:$JAVA_HOME/bin
[root@pw4302-l2 ~]#
```

4.2 Supported system configurations

Before the system installation, check the prerequisites, which are shown in Figure 4-1, Figure 4-2, and Figure 4-3.

Linux x86 64-bit

Operating Systems	Management Node	Compute Node	Client Node	Development Edition / SDK			MapReduce
				C++	Java	Python	
Red Hat Enterprise Linux AS 5.x (Kernel 2.6.x compiled with glibc 2.3.x)	Yes	Yes (Support both 32-bit and 64-bit applications)		Gcc 3.4, 4.0[e], 4.1, 4.6, 4.7 Intel C++ 12.0.2[f] on RHEL 5 and RHEL 6	Oracle JVM: 1.5, 1.6, 1.7 IBM JVM: 1.5, 1.6, 1.7 JRockit: 1.5, 1.6 Eclipse 4.3x	No	Yes Java: Oracle JVM:1.6, 1.7 IBM JVM:1.6, 1.7
Red Hat Enterprise Linux AS 6.x (Kernel 2.6.x compiled with glibc 2.3.x)						Yes Python 2.7.2 and 3.4	
Red Hat Enterprise Linux AS 7.x (Kernel 3.10 compiled with glibc 2.17.x)							
CentOS 6.x (Kernel 2.6.x compiled with glibc 2.3.x)	Same as RHEL 6 support					No	No
SUSE Linux Enterprise Server 10.x (Kernel 2.6.x compiled with glibc 2.3.x)	Yes	Yes (Support both 32-bit and 64-bit applications)		gcc 3.4, 4.0, 4.1, 4.6, 4.7, Intel C++ 12.0.2	Oracle JVM: 1.5, 1.6, 1.7 IBM JVM: 1.5, 1.6, 1.7 JRockit: 1.5, 1.6 (Eclipse 4.3.x)	No	No
SUSE Linux Enterprise Server 11.x (Kernel 2.6.x compiled with glibc 2.3.x)						Yes Python 2.7.2 and 3.4	Yes MapReduce Java support: Oracle JVM:1.6, 1.7 IBM JVM:1.6, 1.7
SUSE Linux Enterprise Server 11.x SP2 and **SP3** (Symphony compiled on kernel 2.6.x running on kernel 3.x)							
Ubuntu 8.04 Server Edition[c] (Kernel 2.6.x compiled with glibc 2.3.x)	On Demand since Symphony 7.1.					No	No
Ubuntu 12.04 Server Edition[c] (Symphony compiled on kernel 2.6.x running on kernel 3.2.0)							

Figure 4-1 Operating system prerequisites on Intel platforms

Linux on PowerPC 64-bit

Operating Systems	Management Node	Compute Node	Client Node	DE / SDK			MapReduce
				C++	Java	Python	
PowerLinux 6 (Kernel 2.6.x compiled with glibc 2.5.x)	Yes	Yes (Support both 32-bit and 64-bit applications)		gcc 4.1.2, 4.6, 4.7	IBM JVM: 1.5, 1.6, 1.7 (Eclipse 4.3.x)	Yes Python 2.7.2 and 3.4	Yes Java: IBM JVM: 1.6, 1.7
PowerLinux 7 (Kernel 2.6.x compiled with glibc 2.5.x) Note: Power 775 (supercomputing) is certified.							
PowerLinux 8 (Kernel 2.6.x compiled with glibc 2.5.x)							

Figure 4-2 Operating system prerequisites on IBM PowerLinux

Supported File Systems for Symphony HA

File System	Version	Notes
NFS		
IBM GPFS	4.1.1	Symphony HA on GPFS is supported on Windows, Linux and Power Linux
Veritas	VxFS 6.0	

Supported Distributed File Systems for MapReduce

Application	Version	Notes
IBM GPFS-FPO	3.5.0.10 3.5.0.11 3.5.0.17 4.1.1	GPFS-FPO on Power Linux: v3.5.0.11, v3.5.0.17 and 4.1.1 GPFS FPO on Linux x86-64: v3.5.0.10, v3.5.0.17 and 4.1.1
Apache Hadoop HDFS	1.1.1 2.4	• Open Source Hadoop 2.4 distribution with both MRv1 and MRv2 applications is supported in standalone Symphony environment. • BigInsights Hadoop 2.4 distribution is supported in Symphony enabled BigInsights environment.
Cloudera CDH	5.1.x	
MapR	1.0.3	
Hortonworks HDP	1.3 2.1	
Appistry Cloud IQ	4.5	On Demand since Symphony 7.1.

Figure 4-3 Supported application releases

4.3 IBM Platform Symphony installation steps

The cluster is built with one master host and five compute hosts. Platform Symphony also has the capability to install a management node for failover, but in this scenario, install the cluster without it. The goal is to check how Intel x86 and IBM PowerPC® work together in a clustered environment. The following installation components are described in this section:

- Master host installation
- Compute host installation

4.3.1 Master host installation

Choose one of the Linux machines on X3850X5 as a master host. First, we show the installation and prerequisite setting steps of the master host, as shown in Example 4-3.

Example 4-3 Installation of Platform Symphony v7.1 with prerequisites

```
[root@pw4302-l2 install_data]# useradd egoadmin
[root@pw4302-l2 symphony]# passwd egoadmin
Changing password for user egoadmin.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@pw4302-l2 ~]# export HADOOP_HOME=/usr/share/hadoop
[root@pw4302-l2 ~]# export LD_LIBRARY_PATH=/usr/lib64
[root@pw4302-l2 ~]# export HADOOP_VERSION=1_1_1
[root@pw4302-l2 ~]# export JAVA_HOME=/usr/java
[root@pw4302-l2 ~]# export CLUSTERADMIN=egoadmin
[root@pw4302-l2 ~]# export CLUSTERNAME=ITSOCluster
[root@pw4302-l2 ~]# export SIMPLIFIEDWEM=N
[root@pw4302-l2 symphony71]# ./symSetup7.1.0_lnx26-lib23-x64.bin

Extracting files... done.
```

```
International License Agreement for Early Release of
Programs

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON
AN "ACCEPT" BUTTON, OR OTHERWISE USING THE PROGRAM,
LICENSEE AGREES TO THE TERMS OF THIS AGREEMENT. IF YOU ARE
ACCEPTING THESE TERMS ON BEHALF OF LICENSEE, YOU REPRESENT
AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND LICENSEE
TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

Press Enter to continue viewing the license agreement, or
enter "1" to accept the agreement, "2" to decline it, "3"
to print it, "4" to read non-IBM terms, or "99" to go back
to the previous screen.
1
This program will use following command to install the EGO and Symphony RPM
packages on the system.
rpm --prefix /opt/ibm/platformsymphony -ivh  ego-lnx26-lib23-x64-3.1.0.rpm
rpm --prefix /opt/ibm/platformsymphony -ivh  soam-lnx26-lib23-x64-7.1.0.rpm
Do you want to continue?(Y/N)Y
Start install...
Preparing...                ########################################### [100%]


Warning
=======
The /etc/services file contains one or more services which are using
the same ports as 7869.  The entry is:
mobileanalyzer  7869/tcp                # MobileAnalyzer& MobileMonitor
Continuing with installation. After installation, you can run egoconfig
setbaseport on every host in the cluster to change the ports used by the cluster.

Warning
=======
The /etc/services file contains one or more services which are using
the same ports as 7870.  The entry is:
rbt-smc         7870/tcp                # Riverbed Steelhead Mobile Service
Continuing with installation. After installation, you can run egoconfig
setbaseport on every host in the cluster to change the ports used by the cluster.
The installation will be processed using the following settings:
Workload Execution Mode (WEM): Advanced
Cluster Administrator: egoadmin
Cluster Name: ITSOCluster
Installation Directory: /opt/ibm/platformsymphony
Connection Base Port: 7869

   1:ego-lnx26-lib23-x64    ########################################### [100%]

Platform EGO $prdversion is installed successfully.
Install the SOAM package to complete the installation process.
Source the environment and run the <egoconfig> command to complete the setup after
installing the SOAM package.Preparing...
########################################### [100%]
```

```
   1:soam-lnx26-lib23-x64   ########################################### [100%]

IBM Platform Symphony 7.1.0 is installed at /opt/ibm/platformsymphony.
[root@pw4302-l2 symphony71]#
```

Before you configure EGO, connect your IP with your host name as root in the `/etc/hosts` directory, as shown in Example 4-4.

Example 4-4 Sample /etc/hosts file to avoid host name error

```
[root@pw4302-l2 symphony71]# cat /etc/hosts
172.16.20.161 pw4302-l2
172.16.20.160 pw4302-l1
172.16.20.164 pw4302-l5
172.16.20.165 pw4302-l6
172.16.21.162 symphony1
172.16.21.163 symphony2
127.0.0.1 localhost
[root@pw4302-l2 symphony71]#
```

Without a connection, you get the error shown in Example 4-5.

Example 4-5 Master hosts config error without correct /etc/hosts file

```
[egoadmin@pw4302-l2 ~]$ egoconfig join pw4302-l2
[pw4302-l2] is not a valid host name.
Command failed.
[egoadmin@pw4302-l2 ~]$
```

To run **egoconfig** and complete the cluster configuration, it is necessary to log in as `egoadmin`. The configuration procedure is shown in Example 4-6. The login details are for `soamlogon`, using `admin` for both user ID and password.

Example 4-6 EGO configuration

```
[root@pw4302-l2 ~]# su - egoadmin
[egoadmin@pw4302-l2 ~]$ . /opt/ibm/platformsymphony/profile.platform
[egoadmin@pw4302-l2 ~]$ egoconfig join pw4302-l2
You are about to create a new cluster with this host as the master host.  Do you
want to continue? [y/n]y
A new cluster <ITSOCluster> has been created. The host <pw4302-l2> is the master
host.
You should run <egoconfig setentitlement "entitlementfile"> before using the
cluster.
[egoadmin@pw4302-l2 ~]$ egoconfig setentitlement
/symphony/install_data/symphony71/platform_sym_adv_entitlement.dat
Successfully set entitlement.
[egoadmin@pw4302-l2 ~]#
[egoadmin@pw4302-l2 ~]$ soamlogon
user account:Admin
password:
Logged on successfully
[egoadmin@pw4302-l2 ~]$
```

Note: The cluster uses configuration files under the directory indicated by (EGO_CONFDIR/../..). The value of the environment variable EGO_CONFDIR changes if the cluster keeps the configuration files in a shared file system. When your documentation refers to this environment variable, substitute the correct directory.

In this scenario, do not set the cluster to handle failover, so the cluster uses the configuration files under the installation directory.

Note: Increase cluster scalability if the number of CPUs in the cluster plus the number of client connections to the cluster exceeds 1000. For example:

As root:

```
. /opt/ibm/platformsymphony/profile.platform
```

Add the following line to the /etc/security/limits.conf:

```
* hard nofile 6400
```

Specify * in the domain column instead of group name, and add hard in the type column for hard limits.

Enable automatic start, grant root privileges to the cluster administrator, and then start the Enterprise Grid Orchestrator (EGO). Example 4-7 shows the commands.

Example 4-7 Finishing the installation

```
[root@pw4302-l2 symphony71]# egosetrc.sh
Egosetrc successful
[root@pw4302-l2 symphony71]# egosetsudoers.sh
Egosetsudoers successful
[root@pw4302-l2 symphony71]# egosh ego start
Start up LIM on <pw4302-l2> ...... done
```

Note: If you ran the **`egosetsudoers.sh`** earlier, you need to run it again with **`egosetsudoers.sh -p`**.

4.3.2 Compute host installation

We want to prove that Platform Symphony can work in a mixed cluster environment. This section shows the installation and login steps:

- "Installation on IBM PowerPC"
- "Platform Symphony installation on Intel based Linux systems" on page 79
- "Log in to the Platform Management Console" on page 81

Installation on IBM PowerPC

In this example, we must install `gettext` for the Platform Symphony installation. Example 4-8 shows the steps.

Example 4-8 Installing gettext and IBM Platform Symphony

```
[root@symphony1 install_data]# cat /etc/hosts
172.16.20.161 pw4302-l2
172.16.20.160 pw4302-l1
172.16.20.164 pw4302-l5
172.16.20.165 pw4302-l6
172.16.21.162 symphony1
172.16.21.163 symphony2
127.0.0.1 localhost
[root@symphony1 install_data]# yum -y install gettext
Loaded plugins: product-id, security, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use
subscription-manager to register.
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package gettext.ppc64 0:0.17-16.el6 will be installed
--> Processing Dependency: libgomp.so.1(GOMP_1.0)(64bit) for package:
gettext-0.17-16.el6.ppc64
--> Processing Dependency: cvs for package: gettext-0.17-16.el6.ppc64
--> Processing Dependency: libgomp.so.1()(64bit) for package:
gettext-0.17-16.el6.ppc64
--> Running transaction check
---> Package cvs.ppc64 0:1.11.23-16.el6 will be installed
---> Package libgomp.ppc64 0:4.4.7-4.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=================================================================================
==
 Package        Arch          Version               Repository
Size
=================================================================================
======
Installing:
 gettext        ppc64         0.17-16.el6            xCAT-rhels6.5-path0
1.9 M
Installing for dependencies:
 cvs            ppc64         1.11.23-16.el6         xCAT-rhels6.5-path0
714 k
 libgomp        ppc64         4.4.7-4.el6            xCAT-rhels6.5-path0
121 k

Transaction Summary
=================================================================================
======
Install       3 Package(s)

Total download size: 2.7 M
Installed size: 8.5 M
```

```
Downloading Packages:
(1/3): cvs-1.11.23-16.el6.ppc64.rpm                             | 714 kB
00:00
(2/3): gettext-0.17-16.el6.ppc64.rpm                            | 1.9 MB
00:00
(3/3): libgomp-4.4.7-4.el6.ppc64.rpm                            | 121 kB
00:00
--------------------------------------------------------------------------------
------
Total                                                    74 MB/s | 2.7 MB
00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : libgomp-4.4.7-4.el6.ppc64
1/3
  Installing : cvs-1.11.23-16.el6.ppc64
2/3
  Installing : gettext-0.17-16.el6.ppc64
3/3
  Verifying  : cvs-1.11.23-16.el6.ppc64
1/3
  Verifying  : gettext-0.17-16.el6.ppc64
2/3
  Verifying  : libgomp-4.4.7-4.el6.ppc64
3/3

Installed:
  gettext.ppc64 0:0.17-16.el6

Dependency Installed:
  cvs.ppc64 0:1.11.23-16.el6              libgomp.ppc64 0:4.4.7-4.el6

Complete!
[root@symphony1 install_data]# useradd egoadmin
[root@symphony1 install_data]# passwd egoadmin
Changing password for user egoadmin.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@symphony1 ~]# export HADOOP_HOME=/usr/share/hadoop
[root@symphony1 ~]# export LD_LIBRARY_PATH=/usr/lib64
[root@symphony1 ~]# export HADOOP_VERSION=1_1_1
[root@symphony1 ~]# export JAVA_HOME=/usr/java
[root@symphony1 ~]# export CLUSTERADMIN=egoadmin
[root@symphony1 ~]# export CLUSTERNAME=ITSOCluster
[root@symphony1 ~]# export SIMPLIFIEDWEM=N
root@symphony1 install_data]# cd /mnt/install_data/symphony71/
[root@symphony1 symphony71]# ./symcompSetup7.1.0_lnx26-lib25-ppc64.bin

Extracting files... done.
This program will use following command to install the EGO and Symphony RPM
packages on the system.
rpm --prefix /opt/ibm/platformsymphony -ivh  egocomp-lnx26-lib25-ppc64-3.1.0.rpm
```

```
rpm --prefix /opt/ibm/platformsymphony -ivh  soam-lnx26-lib25-ppc64-7.1.0.rpm
Do you want to continue?(Y/N)Y
Start install...
Preparing...                ########################################### [100%]


Warning
=======
The /etc/services file contains one or more services which are using
the same ports as 7869.  The entry is:
mobileanalyzer  7869/tcp                # MobileAnalyzer& MobileMonitor
Continuing with installation. After installation, you can run egoconfig
setbaseport on every host in the cluster to change the ports used by the cluster.

Warning
=======
The /etc/services file contains one or more services which are using
the same ports as 7870.  The entry is:
rbt-smc         7870/tcp                # Riverbed Steelhead Mobile Service
Continuing with installation. After installation, you can run egoconfig
setbaseport on every host in the cluster to change the ports used by the cluster.
The installation will be processed using the following settings:
Workload Execution Mode (WEM): Advanced
Cluster Administrator: egoadmin
Cluster Name: ITSOCluster
Installation Directory: /opt/ibm/platformsymphony
Connection Base Port: 7869

   1:egocomp-lnx26-lib25-ppc########################################### [100%]


Platform EGO 3.1.0 (compute host package) is installed at
/opt/ibm/platformsymphony.
Remember to use the egoconfig command to complete the setup process.
Preparing...                ########################################### [100%]

   1:soam-lnx26-lib25-ppc64 ########################################### [100%]


IBM Platform Symphony 7.1.0 is installed at /opt/ibm/platformsymphony.
[root@symphony1 symphony71]#
```

After Platform Symphony has been installed, configure EGO and start the services. Example 4-9 shows the steps.

Example 4-9 Configuring EGO and starting the services

```
[root@symphony1 symphony71]# su - egoadmin
[egoadmin@symphony1 ~]$ . /opt/ibm/platformsymphony/profile.platform
[egoadmin@symphony1 ~]$ egoconfig join pw4302-l2
You are about to join this host to a cluster with master host pw4302-l2.  Do you
want to continue? [y/n]y
The host symphony1 has joined the cluster ITSOCluster.
[egoadmin@symphony1 ~]$ soamlogon
user account:Admin
```

```
password:
Logged on successfully
[egoadmin@symphony1 ~]$ egosh resource list -l
NAME                   status        mem    swp    tmp   ut    it    pg   r1m
r15s  r15m  ls
pw4302-l5              ok            14G  8044M    75G   1%   170     0     0
0     0   1
pw4302-l1              ok            15G  8044M    38G   1%  1216     0     0
0     0   0
pw4302-l2              ok          6893M  8047M    29G   1%    10     0     0
4.1     0   1
symphony2              ok          3726M  1023M    84G   3%    19     0   0.1
0   0.1   0
[egoadmin@symphony1 ~]$ exit
logout
[root@symphony1 symphony71]# . /opt/ibm/platformsymphony/profile.platform
[root@symphony1 symphony71]# egosetrc.sh
Egosetrc successful
[root@symphony1 symphony71]# egosetsudoers.sh
Ego_setsudoer_success
[root@symphony1 symphony71]# egosh ego start
Start up LIM on <symphony1> ...... done
[root@symphony1 symphony71]#
```

Platform Symphony installation on Intel based Linux systems

Example 4-10 shows the complete Platform Symphony installation steps for and Intel based Linux system.

Example 4-10 Installation on an Intel based Linux system

```
[root@pw4302-l6 install_data]# cat /etc/hosts
172.16.20.161 pw4302-l2
172.16.20.160 pw4302-l1
172.16.20.164 pw4302-l5
172.16.20.165 pw4302-l6
172.16.21.162 symphony1
172.16.21.163 symphony2
127.0.0.1 localhost
[root@pw4302-l6 install_data]# useradd egoadmin
[root@pw4302-l6 install_data]# passwd egoadmin
Changing password for user egoadmin.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@pw4302-l6 ~]# export HADOOP_HOME=/usr/share/hadoop
[root@pw4302-l6 ~]# export LD_LIBRARY_PATH=/usr/lib64
[root@pw4302-l6 ~]# export HADOOP_VERSION=1_1_1
[root@pw4302-l6 ~]# export JAVA_HOME=/usr/java
[root@pw4302-l6 ~]# export CLUSTERADMIN=egoadmin
[root@pw4302-l6 ~]# export CLUSTERNAME=ITSOCluster
[root@pw4302-l6 ~]# export SIMPLIFIEDWEM=N
[root@pw4302-l6 install_data]# cd /mnt/install_data/symphony71/
[root@pw4302-l6 symphony71]# ./symcompSetup7.1.0_lnx26-lib23-x64.bin

Extracting files... done.
```

```
This program will use following command to install the EGO and Symphony RPM
packages on the system.
rpm --prefix /opt/ibm/platformsymphony -ivh  ego-lnx26-lib23-x64-3.1.0.rpm
rpm --prefix /opt/ibm/platformsymphony -ivh  soam-lnx26-lib23-x64-7.1.0.rpm
Do you want to continue?(Y/N)Y
Start install...
Preparing...                ########################################### [100%]


Warning
=======
The /etc/services file contains one or more services which are using
the same ports as 7869.  The entry is:
mobileanalyzer  7869/tcp                # MobileAnalyzer& MobileMonitor
Continuing with installation. After installation, you can run egoconfig
setbaseport on every host in the cluster to change the ports used by the cluster.

Warning
=======
The /etc/services file contains one or more services which are using
the same ports as 7870.  The entry is:
rbt-smc         7870/tcp                # Riverbed Steelhead Mobile Service
Continuing with installation. After installation, you can run egoconfig
setbaseport on every host in the cluster to change the ports used by the cluster.
The installation will be processed using the following settings:
Workload Execution Mode (WEM): Advanced
Cluster Administrator: egoadmin
Cluster Name: ITSOCluster
Installation Directory: /opt/ibm/platformsymphony
Connection Base Port: 7869

   1:egocomp-lnx26-lib23-x64 ########################################### [100%]


Platform EGO 3.1.0 (compute host package) is installed at
/opt/ibm/platformsymphony.
Remember to use the egoconfig command to complete the setup process.
Preparing...                ########################################### [100%]

   1:soam-lnx26-lib23-x64 ########################################### [100%]


IBM Platform Symphony 7.1.0 is installed at /opt/ibm/platformsymphony.
[root@pw4302-l6 symphony71]#
[root@pw4302-l6 symphony71]# su - egoadmin
[egoadmin@pw4302-l6 ~]$ . /opt/ibm/platformsymphony/profile.platform
[egoadmin@pw4302-l6 ~]$ egoconfig join pw4302-l2
You are about to join this host to a cluster with master host pw4302-l2.  Do you
want to continue? [y/n]y
The host pw4302-l6 has joined the cluster ITSOCluster.
[egoadmin@pw4302-l6 ~]$ soamlogon
user account:Admin
password:
Logged on successfully
[egoadmin@pw4302-l6 ~]$ exit
```

```
logout
[root@pw4302-l6 symphony71]# . /opt/ibm/platformsymphony/profile.platform
[root@pw4302-l6 symphony71]# egosetrc.sh
Egosetrc successful
[root@pw4302-l6 symphony71]# egosetsudoers.sh
Ego_setsudoer_success
[root@pw4302-l6 symphony71]# egosh ego start
Start up LIM on <pw4302-l6> ...... done
[root@pw4302-l6 symphony71]#
```

Log in to the Platform Management Console

Now, log in to the Master server console via the web browser, using the following address:

`http://Host_M:8080/platform`

`Admin` is the default login for both user ID and password, as shown in Figure 4-4.

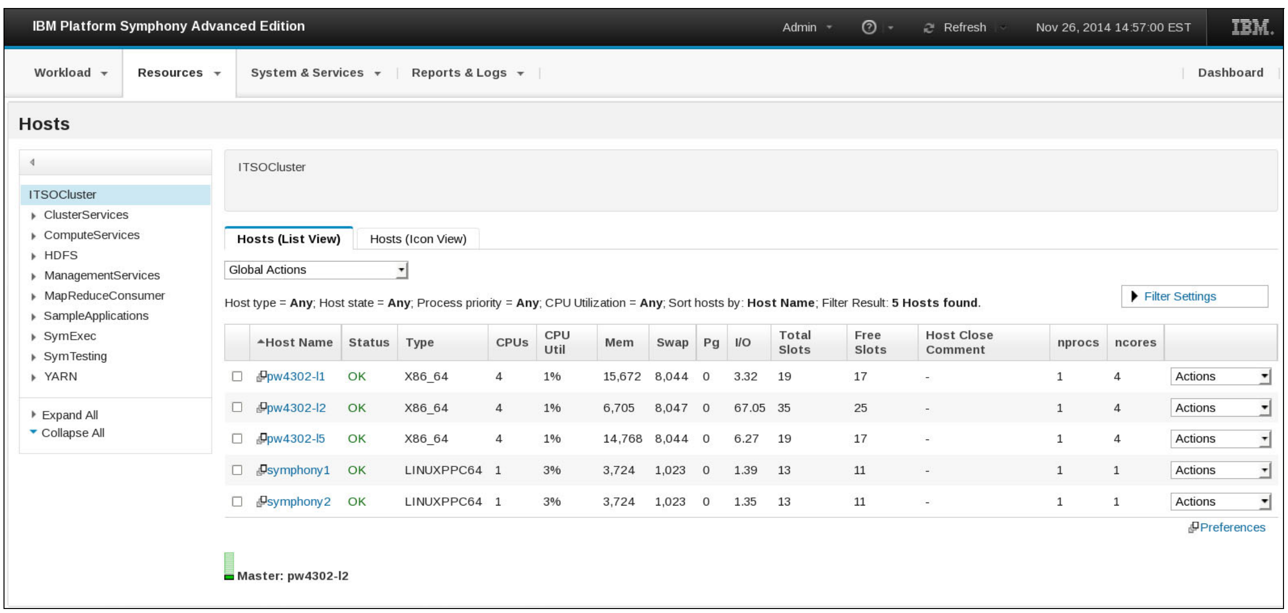


Figure 4-4 Accessing the master server console

Figure 4-4 shows the list of the compute nodes and the master nodes. After you add a new node to the cluster, wait a few minutes for it to appear in the console display.

If you want to add more compute hosts, follow “Installation on IBM PowerPC” on page 76 or “Platform Symphony installation on Intel based Linux systems” on page 79.

4.4 Compiling and installing Hadoop 1.1.1 for IBM PowerPC

When this publication was written, there was no Hadoop 1.1.1 for IBM PowerPC. So, we compiled it to build our cluster. In this section, we show the necessary steps in these subsections:

- Dependencies
- Building Hadoop

4.4.1 Dependencies

Install the dependencies first:

- Red Hat Enterprise Linux dependencies
- Java
- Apache dependencies
- Build the commons-daemon source

Red Hat Enterprise Linux dependencies

Use these commands to install the Red Hat Enterprise Linux (RHEL) dependencies:

```
yum groupinstall "Development Tools"
yum install subversion openssl openssl-devel zlib zlib-devel gzip
compat-libstdc++-33
```

Java

Use the `ibm-java-ppc64-sdk-7.1-2.0.bin` file, which you can download from the Linux Download page on IBM developerWorks®:

http://www.ibm.com/developerworks/java/jdk/linux/download.html

Compile it with the following command default settings:

```
/tmp/ibm-java-ppc64-sdk-7.1-2.0.bin
```

Apache dependencies

You need the following binary files to compile and build Hadoop for PowerPC. Check the respective Apache projects to get these binary files:

- `apache-ant-1.9.4` from http://ant.apache.org
- `apache-forrest-0.9` from http://forrest.apache.org

We placed the decompressed directories in the `/usr/local` directory.

Build the commons-daemon source

You also need to build the Apache commons-daemon binary `jsvc file`. Follow the instructions on the Apache website page titled “Ref: HADOOP-8273 Update url for commons daemon ppc64 binary tarball:”

https://issues.apache.org/jira/browse/DAEMON-249

1. Download the `commons-daemon-1.0.2-native-src.zip` file (or the appropriate version for the build):

   http://archive.apache.org/dist/commons/daemon/source/commons-daemon-1.0.2-native-src.zip

2. Extract the downloaded file in a directory called `/home/hadoop/commons` (or a directory of your choice).
3. Change directory to `/home/hadoop/commons/commons-daemon-1.0.2-native-src/unix`.
4. Run **`./configure`**.

During the configuration phase, we got the error shown in Example 4-11.

Example 4-11 Error during the configuration phase

```
[root@hadooponpower unix]# ./configure
*** Current host ***
checking build system type... ./support/config.guess: unable to guess system type

This script, last modified 2001-04-20, has failed to recognize
the operating system you are using. It is advised that you
download the most up to date version of the config scripts from

    ftp://ftp.gnu.org/pub/gnu/config/

If the version you run (./support/config.guess) is already up to date, please
send the following data and any information you think might be
pertinent to <config-patches@gnu.org> in order to provide the needed
information to handle your system.

config.guess timestamp = 2001-04-20

uname -m = ppc64
uname -r = 2.6.32-431.el6.ppc64
uname -s = Linux
uname -v = #1 SMP Sun Nov 10 22:17:43 EST 2013

/usr/bin/uname -p =
/bin/uname -X     =

hostinfo               =
/bin/universe          =
/usr/bin/arch -k       =
/bin/arch              = ppc64
/usr/bin/oslevel       =
/usr/convex/getsysinfo =

UNAME_MACHINE = ppc64
UNAME_RELEASE = 2.6.32-431.el6.ppc64
UNAME_SYSTEM  = Linux
UNAME_VERSION = #1 SMP Sun Nov 10 22:17:43 EST 2013
configure: error: cannot guess build type; you must specify one
[root@hadooponpower unix]#
```

To correct the error, change the `./support/config.guess` file. The original configuration file is shown in Example 4-12.

Example 4-12 Original configuration file

```
#ifdef __MIPSEL__
  printf ("%sel-unknown-linux-gnu\n", argv[1]);
#endif
  return 0;
}
EOF
        $CC_FOR_BUILD $dummy.c -o $dummy 2>/dev/null && ./$dummy "${UNAME_MACHIN
E}" && rm -f $dummy.c $dummy && exit 0
```

```
        rm -f $dummy.c $dummy
        ;;
    ppc:Linux:*:*)
        # Determine Lib Version
        cat >$dummy.c <<EOF
#include <features.h>
```

We changed the configuration to the value shown in Example 4-13.

Example 4-13 Changing the configuration values

```
#ifdef __MIPSEL__
  printf ("%sel-unknown-linux-gnu\n", argv[1]);
#endif
  return 0;
}
EOF
        $CC_FOR_BUILD $dummy.c -o $dummy 2>/dev/null && ./$dummy "${UNAME_MACHIN
E}" && rm -f $dummy.c $dummy && exit 0
        rm -f $dummy.c $dummy
        ;;
    ppc64:Linux:*:*)
        # Determine Lib Version
        cat >$dummy.c <<EOF
#include <features.h>
```

After the change, it runs. You can see the output in Example 4-14.

Example 4-14 Successful configuration run after the changes

```
[root@hadooponpower unix]# ./configure
*** Current host ***
checking build system type... powerpc-unknown-linux-gnu
checking host system type... powerpc-unknown-linux-gnu
checking cached host system type... ok
*** C-Language compilation tools ***
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for ranlib... ranlib
*** Host support ***
checking C flags dependant on host system type... ok
*** Java compilation tools ***
checking for sablevm... NONE
checking for kaffe... NONE
checking for javac... /opt/ibm/java-ppc64-71/bin/javac
/opt/ibm/java-ppc64-71/bin/javac
checking wether the Java compiler (/opt/ibm/java-ppc64-71/bin/javac) works... yes
checking for jar... /opt/ibm/java-ppc64-71/bin/jar
gcc flags added
```

```
*** Writing output files ***
configure: creating ./config.status
config.status: creating Makefile
config.status: creating Makedefs
config.status: creating native/Makefile
*** All done ***
Now you can issue "make"
[root@hadooponpower unix]#
```

5. Build the `jsvc` binary file by typing the **`make`** command.
6. Create a build directory and copy the `jsvc` binary that was just built into a new directory called `/home/hadoop/commons/commons-daemon-1.0.2-native-src/unix/build`.
7. Copy the following text files from the `/home/hadoop/commons/commons-daemon-1.0.2-native-src` directory to the `/home/hadoop/commons/commons-daemon-1.0.2-native-src/unix/build` directory:
   - LICENSE.txt
   - NOTICE.txt
   - RELEASE-NOTES.txt
8. Change to the `home/hadoop/commons/commons-daemon-1.0.2-native-src/unix/build` directory.
9. Create the binary `.tar` file for IBM PowerPC by issuing the following command:

   ```
   tar -czvf commons-daemon-1.0.2-bin-linux-ppc64.tar.gz *
   ```

10. Copy the file to an appropriate directory where this can be accessed from the hadoop-common build.
11. For branch-1, edit the `build.xml` file on the build root to point to the binary `.tar` file that was just built:

   ```
   <property name="jsvc.location"
   value="file:///home/hadoop/commons-daemon-1.0.2-bin-linux-ppc64.tar.gz"/>
   ```

See Example 4-15.

Example 4-15 Checking the configuration changes

```
[root@hadooponpower src]# cat /usr/local/hadoop-1.1.1/build.xml |grep 'property
name="jsvc.location'
  <property name="jsvc.location"
value="file:///usr/local/build/commons-daemon-1.0.2-bin-linux-ppc64.tar.gz" />
[root@hadooponpower src]#
```

4.4.2 Building Hadoop

The topics in this section show you how you can build Hadoop from the source and how to install it (we assume that your machine has an Internet connection):

- "Download the Hadoop source code"
- "Set the environment variables" on page 86
- "Compile Hadoop with the output .tar file" on page 86
- "Compile Hadoop with the output rpm" on page 87
- "Install Hadoop" on page 88

Download the Hadoop source code

You are now ready to build Hadoop from the Hadoop source code.

Download Hadoop version 1.1.1 from this website:

http://svn.apache.org/repos/asf/hadoop/common/tags/release-1.1.1/

Move it to the `/usr/ local` directory:

```
mv ./hadoop-1.1.1 /usr/local/
```

Set the environment variables

You can see our `/etc/profile` file in Example 4-16, which relates to the installation.

Example 4-16 /etc/profile file

```
...
export JAVA_HOME=/usr/ibm_java_7
export ANT_HOME=/usr/local/apache-ant-1.9.4
export PATH=$PATH:${JAVA_HOME}/bin:${ANT_HOME}/bin
...
```

Note: If you change your profile file, reread it with the following command:

```
. ./etc/profile
```

Change the file `/usr/local/hadoop-1.1.1/src/contrib/fuse-dfs/src/Makefile.am` to reflect the correct Java virtual machine (JVM) and library path for `libhdfs`. In this case, you can see the correct line in Example 4-17.

Example 4-17 Hadoop Makefile.am

```
AM_LDFLAGS= -L$(HADOOP_HOME)/build/c++/Linux-ppc64-64/lib -lhdfs
-L$(FUSE_HOME)/lib -lfuse -L$(JAVA_HOME)/jre/lib/$(OS_ARCH)/j9vm -ljvm
```

Before compiling it, install fuse with the following command:

```
yum install fuse.ppc64 fuse-devel.ppc64 fuse-libs.ppc64
```

Compile Hadoop with the output .tar file

If you do not install Fuse in your system, you get the error in Example 4-18 during the build.

Example 4-18 Error during the build

```
BUILD FAILED
/usr/local/hadoop-1.1.1/build.xml:706: The following error occurred while
executing this line:
/usr/local/hadoop-1.1.1/src/contrib/build.xml:30: The following error occurred
while executing this line:
/usr/local/hadoop-1.1.1/src/contrib/fuse-dfs/build.xml:64: exec returned: 2
```

Now you are ready to build Hadoop and generate the binaries by using the **ant** command. It is important to compile the C/C++ libraries and code for the ppc64 platform. Therefore, you need to add those options to your build:

```
ant -Dlibhdfs=true -Dcompile.native=true -Dfusedfs=true -Dcompile.c++=true
-Dforrest.home=/usr/local/apache-forrest-0.9 compile-core-native compile-c++
compile-c++-examples task-controller tar record-parser compile-hdfs-classes
package -Djava5.home=/opt/ibm/java-ppc64-71
```

The last lines of the compilation are shown in Example 4-19.

Example 4-19 Showing the last line of the compilation

```
jsvc:
      [get] Getting:
file:/usr/local/build/commons-daemon-1.0.2-bin-linux-ppc64.tar.gz
      [get] To: /usr/local/hadoop-1.1.1/build/jsvc.ppc64/jsvc.ppc64.tar.gz
    [untar] Expanding: /usr/local/hadoop-1.1.1/build/jsvc.ppc64/jsvc.ppc64.tar.gz
into /usr/local/hadoop-1.1.1/build/jsvc.ppc64
     [copy] Copying 1 file to
/usr/local/hadoop-1.1.1/build/hadoop-1.1.2-SNAPSHOT/libexec
     [copy] Copying /usr/local/hadoop-1.1.1/build/jsvc.ppc64/jsvc to
/usr/local/hadoop-1.1.1/build/hadoop-1.1.2-SNAPSHOT/libexec/jsvc.ppc64

BUILD SUCCESSFUL
Total time: 9 minutes 4 seconds
[root@hadooponpower hadoop-1.1.1]#
```

You should find the `.jar` files and a tarball of Hadoop tools in the `/build` directory.

Compile Hadoop with the output rpm

After the successful tarball file build, build an rpm to install Hadoop easily to other machines. The **ant** command creates the rpm:

```
ant -Dlibhdfs=true -Dcompile.native=true -Dfusedfs=true -Dcompile.c++=true
-Dforrest.home=/usr/local/apache-forrest-0.9 compile-core-native compile-c++
compile-c++-examples task-controller rpm record-parser compile-hdfs-classes
package -Djava5.home=/opt/ibm/java-ppc64-71
```

The **ant** command creates the rpmbuild files but does not build it because of the error shown in Example 4-20.

Example 4-20 Error before the build

```
/tmp/hadoop_package_build_root/BUILDROOT/hadoop-1.1.0-1.ppc64/etc/hadoop/taskcontr
oller.cfg
      [rpm]     File not found:
/tmp/hadoop_package_build_root/BUILDROOT/hadoop-1.1.0-1.ppc64/etc/hadoop/fair-sche
duler.xml
      [rpm]     File not found:
/tmp/hadoop_package_build_root/BUILDROOT/hadoop-1.1.0-1.ppc64/usr
      [rpm]     File not found:
/tmp/hadoop_package_build_root/BUILDROOT/hadoop-1.1.0-1.ppc64/usr/libexec
      [rpm]     File not found:
/tmp/hadoop_package_build_root/BUILDROOT/hadoop-1.1.0-1.ppc64/etc/rc.d/init.d
      [rpm] Processing files: hadoop-1.1.0-1.${os
      [rpm]
```

```
      [rpm]
      [rpm] RPM build errors:

BUILD FAILED
/usr/local/hadoop-1.1.1/build.xml:1869: '/usr/bin/rpmbuild' failed with exit code
1

Total time: 3 minutes 36 seconds
[root@hadooponpower hadoop-1.1.1]#
```

The files are under `/tmp/hadoop_package_build_root/`. Perform the step as shown in Example 4-21.

Example 4-21 One step before the build

```
[root@hadooponpower hadoop_package_build_root]# pwd
/tmp/hadoop_package_build_root
[root@hadooponpower hadoop_package_build_root]# cp -pr * /root/rpmbuild/
```

Change the INSTALL SECTION in the /tmp/hadoop_package_build_root/SPECS/hadoop.spec file, as shown in Example 4-22.

Example 4-22 Changing the installation section

```
#########################
#### INSTALL SECTION ####
#########################
%install
mv ${RPM_BUILD_DIR}/%{_final_name}/etc/hadoop/* ${RPM_BUILD_DIR}%{_conf_dir}
mv ${RPM_BUILD_DIR}/%{_final_name}/* ${RPM_BUILD_DIR}%{_prefix}

if [ "${RPM_BUILD_DIR}%{_conf_dir}" != "${RPM_BUILD_DIR}/%{_prefix}/conf" ]; then
  rm -rf ${RPM_BUILD_DIR}/%{_prefix}/etc
fi
/bin/cp -fpr /tmp/hadoop_package_build_root/BUILD/usr
/tmp/hadoop_package_build_root/BUILD/etc /tmp/hadoop_package_build_root/BUILD/var
/root/rpmbuild/BUILDROOT/hadoop-1.1.0-1.ppc64/
```

Then, run the following command:

```
rpmbuild -bb SPECS/hadoop.spec
```

The command creates the rpm in this directory:

```
/root/rpmbuild/RPMS/ppc64/hadoop-1.1.0-1.ppc64.rpm
```

Install Hadoop

For installation on PowerPC, see Example 4-23.

Example 4-23 Installation on PPC

```
[root@symphony1 install_data]# rpm -ivh hadoop-1.1.0-1.ppc64.rpm
Preparing...                ########################################### [100%]
   1:hadoop                 ########################################### [100%]
[root@symphony1 install_data]#
```

The installation is the same as on x86 Linux.

4.5 IBM Spectrum Scale installation and configuration

This topic shows how to create a Spectrum Scale cluster in our environment. The installation steps are for IBM PowerPC, although the methodology is the same for Intel x86:

- 4.5.1, “Installation steps” on page 89
- 4.5.2, “Cluster configuration steps” on page 99
- 4.5.3, “Spectrum Scale network shared disk creation” on page 101

4.5.1 Installation steps

Follow these steps to install IBM Spectrum Scale, which is scalable, high-performance data and file management software that is based upon IBM General Parallel File System or GPFS technology:

1. First, download the IBM Spectrum Scale Standard Edition version 4.1, 64-bit:

   http://w3.ibm.com/software/xl/download/ticket.do

2. Download the latest applicable Fix Pack from the IBM Fix Central website:

   http://ibm.co/1NDEAUH

3. Extract the downloaded installation files as shown in Example 4-24.

Example 4-24 Unpack Spectrum Scale install files

```
[root@symphony1 gpfs_power]# ./gpfs_install-4.1.0-0_ppc64 --text-only

Extracting License Acceptance Process Tool to /usr/lpp/mmfs/4.1 ...
tail -n +456 ./gpfs_install-4.1.0-0_ppc64 | /bin/tar -C /usr/lpp/mmfs/4.1 -xvz
--exclude=*rpm --exclude=*tgz --exclude=*deb 2> /dev/null 1> /dev/null

Installing JRE ...
tail -n +456 ./gpfs_install-4.1.0-0_ppc64 | /bin/tar -C /usr/lpp/mmfs/4.1
--wildcards -xvz  ./ibm-java*tgz 2> /dev/null 1> /dev/null

Invoking License Acceptance Process Tool ...
/usr/lpp/mmfs/4.1/ibm-java-ppc64-60/jre/bin/java -cp
/usr/lpp/mmfs/4.1/LAP_HOME/LAPApp.jar com.ibm.lex.lapapp.LAP -l
/usr/lpp/mmfs/4.1/LA_HOME -m /usr/lpp/mmfs/4.1 -s /usr/lpp/mmfs/4.1  -text_only
International Program License Agreement

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON
AN "ACCEPT" BUTTON, OR OTHERWISE USING THE PROGRAM,
LICENSEE AGREES TO THE TERMS OF THIS AGREEMENT. IF YOU ARE
ACCEPTING THESE TERMS ON BEHALF OF LICENSEE, YOU REPRESENT
AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND LICENSEE
TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

* DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, CLICK ON AN
"ACCEPT" BUTTON, OR USE THE PROGRAM; AND

* PROMPTLY RETURN THE UNUSED MEDIA, DOCUMENTATION, AND

Press Enter to continue viewing the license agreement, or
```

```
enter "1" to accept the agreement, "2" to decline it, "3"
to print it, "4" to read non-IBM terms, or "99" to go back
to the previous screen.
1

License Agreement Terms accepted.

Extracting Product RPMs to /usr/lpp/mmfs/4.1 ...
tail -n +456 ./gpfs_install-4.1.0-0_ppc64 | /bin/tar -C /usr/lpp/mmfs/4.1
--wildcards -xvz  ./gpfs.base-4.1.0-0.ppc64.rpm ./gpfs.crypto-4.1.0-0.ppc64.rpm
./gpfs.docs-4.1.0-0.noarch.rpm ./gpfs.ext-4.1.0-0.ppc64.rpm
./gpfs.gnr-4.1.0-0.sles.ppc64.rpm ./gpfs.gpl-4.1.0-0.noarch.rpm
./gpfs.gskit-8.0.50-16.ppc64.rpm ./gpfs.msg.en_US-4.1.0-0.noarch.rpm 2>
/dev/null 1> /dev/null

   - gpfs.base-4.1.0-0.ppc64.rpm
   - gpfs.crypto-4.1.0-0.ppc64.rpm
   - gpfs.docs-4.1.0-0.noarch.rpm
   - gpfs.ext-4.1.0-0.ppc64.rpm
   - gpfs.gnr-4.1.0-0.sles.ppc64.rpm
   - gpfs.gpl-4.1.0-0.noarch.rpm
   - gpfs.gskit-8.0.50-16.ppc64.rpm
   - gpfs.msg.en_US-4.1.0-0.noarch.rpm

Removing License Acceptance Process Tool from /usr/lpp/mmfs/4.1 ...

rm -rf  /usr/lpp/mmfs/4.1/LAP_HOME /usr/lpp/mmfs/4.1/LA_HOME

Removing JRE from /usr/lpp/mmfs/4.1 ...

rm -rf /usr/lpp/mmfs/4.1/ibm-java*tgz
[root@symphony1 gpfs_power]# ll /usr/lpp/mmfs/4.1/
total 25892
-rw-r--r-- 1 root root 14790184 Apr 25  2014 gpfs.base-4.1.0-0.ppc64.rpm
-rw-r--r-- 1 root root   213059 Apr 25  2014 gpfs.crypto-4.1.0-0.ppc64.rpm
-rw-r--r-- 1 root root   292356 Apr 25  2014 gpfs.docs-4.1.0-0.noarch.rpm
-rw-r--r-- 1 root root  1503907 Apr 25  2014 gpfs.ext-4.1.0-0.ppc64.rpm
-rw-r--r-- 1 root root  4424323 Apr 25  2014 gpfs.gnr-4.1.0-0.sles.ppc64.rpm
-rw-r--r-- 1 root root   573469 Apr 25  2014 gpfs.gpl-4.1.0-0.noarch.rpm
-rw-r--r-- 1 root root  4557142 Apr 25  2014 gpfs.gskit-8.0.50-16.ppc64.rpm
-rw-r--r-- 1 root root   131507 Apr 25  2014 gpfs.msg.en_US-4.1.0-0.noarch.rpm
drwxr-xr-x 3 root root     4096 Dec  3 14:11 license
[root@symphony1 gpfs_power]#
```

4. Install the unpacked files as shown in Example 4-25.

Example 4-25 Installing Spectrum Scale

```
[root@symphony1 gpfs_power]# yum install gpfs.base-4.1.0-0.ppc64.rpm
gpfs.crypto-4.1.0-0.ppc64.rpm gpfs.docs-4.1.0-0.noarch.rpm
gpfs.ext-4.1.0-0.ppc64.rpm gpfs.gpl-4.1.0-0.noarch.rpm
gpfs.gskit-8.0.50-16.ppc64.rpm gpfs.msg.en_US-4.1.0-0.noarch.rpm
Loaded plugins: product-id, security, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use
subscription-manager to register.
Setting up Install Process
```

```
Examining gpfs.base-4.1.0-0.ppc64.rpm: gpfs.base-4.1.0-0.ppc64
Marking gpfs.base-4.1.0-0.ppc64.rpm to be installed
Examining gpfs.crypto-4.1.0-0.ppc64.rpm: gpfs.crypto-4.1.0-0.ppc64
Marking gpfs.crypto-4.1.0-0.ppc64.rpm to be installed
Examining gpfs.docs-4.1.0-0.noarch.rpm: gpfs.docs-4.1.0-0.noarch
Marking gpfs.docs-4.1.0-0.noarch.rpm to be installed
Examining gpfs.ext-4.1.0-0.ppc64.rpm: gpfs.ext-4.1.0-0.ppc64
Marking gpfs.ext-4.1.0-0.ppc64.rpm to be installed
Examining gpfs.gpl-4.1.0-0.noarch.rpm: gpfs.gpl-4.1.0-0.noarch
Marking gpfs.gpl-4.1.0-0.noarch.rpm to be installed
Examining gpfs.gskit-8.0.50-16.ppc64.rpm: gpfs.gskit-8.0.50-16.ppc64
Marking gpfs.gskit-8.0.50-16.ppc64.rpm to be installed
Examining gpfs.msg.en_US-4.1.0-0.noarch.rpm: gpfs.msg.en_US-4.1.0-0.noarch
Marking gpfs.msg.en_US-4.1.0-0.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package gpfs.base.ppc64 0:4.1.0-0 will be installed
---> Package gpfs.crypto.ppc64 0:4.1.0-0 will be installed
---> Package gpfs.docs.noarch 0:4.1.0-0 will be installed
---> Package gpfs.ext.ppc64 0:4.1.0-0 will be installed
---> Package gpfs.gpl.noarch 0:4.1.0-0 will be installed
---> Package gpfs.gskit.ppc64 0:8.0.50-16 will be installed
---> Package gpfs.msg.en_US.noarch 0:4.1.0-0 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

===============================================================================
===============================================================================
=================
 Package                                 Arch
Version                        Repository
Size
===============================================================================
===============================================================================
=================
Installing:
 gpfs.base                               ppc64
4.1.0-0                       /gpfs.base-4.1.0-0.ppc64
45 M
 gpfs.crypto                             ppc64
4.1.0-0                       /gpfs.crypto-4.1.0-0.ppc64
478 k
 gpfs.docs                               noarch
4.1.0-0                       /gpfs.docs-4.1.0-0.noarch
1.3 M
 gpfs.ext                                ppc64
4.1.0-0                       /gpfs.ext-4.1.0-0.ppc64
4.9 M
 gpfs.gpl                                noarch
4.1.0-0                       /gpfs.gpl-4.1.0-0.noarch
2.3 M
 gpfs.gskit                              ppc64
8.0.50-16                     /gpfs.gskit-8.0.50-16.ppc64
16 M
```

```
 gpfs.msg.en_US                          noarch
4.1.0-0                       /gpfs.msg.en_US-4.1.0-0.noarch
503 k

Transaction Summary
================================================================================
================================================================================
=================
Install       7 Package(s)

Total size: 70 M
Installed size: 70 M
Is this ok [y/N]: y^CExiting on user Command
[root@symphony1 gpfs_power]# ^C
[root@symphony1 gpfs_power]# ^C
[root@symphony1 gpfs_power]# ^C
[root@symphony1 gpfs_power]# ^C
[root@symphony1 gpfs_power]# yum install gpfs.base-4.1.0-0.ppc64.rpm
gpfs.docs-4.1.0-0.noarch.rpm gpfs.ext-4.1.0-0.ppc64.rpm
gpfs.gpl-4.1.0-0.noarch.rpm gpfs.gskit-8.0.50-16.ppc64.rpm
gpfs.msg.en_US-4.1.0-0.noarch.rpm
Loaded plugins: product-id, security, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use
subscription-manager to register.
Setting up Install Process
Examining gpfs.base-4.1.0-0.ppc64.rpm: gpfs.base-4.1.0-0.ppc64
Marking gpfs.base-4.1.0-0.ppc64.rpm to be installed
Examining gpfs.docs-4.1.0-0.noarch.rpm: gpfs.docs-4.1.0-0.noarch
Marking gpfs.docs-4.1.0-0.noarch.rpm to be installed
Examining gpfs.ext-4.1.0-0.ppc64.rpm: gpfs.ext-4.1.0-0.ppc64
Marking gpfs.ext-4.1.0-0.ppc64.rpm to be installed
Examining gpfs.gpl-4.1.0-0.noarch.rpm: gpfs.gpl-4.1.0-0.noarch
Marking gpfs.gpl-4.1.0-0.noarch.rpm to be installed
Examining gpfs.gskit-8.0.50-16.ppc64.rpm: gpfs.gskit-8.0.50-16.ppc64
Marking gpfs.gskit-8.0.50-16.ppc64.rpm to be installed
Examining gpfs.msg.en_US-4.1.0-0.noarch.rpm: gpfs.msg.en_US-4.1.0-0.noarch
Marking gpfs.msg.en_US-4.1.0-0.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package gpfs.base.ppc64 0:4.1.0-0 will be installed
---> Package gpfs.docs.noarch 0:4.1.0-0 will be installed
---> Package gpfs.ext.ppc64 0:4.1.0-0 will be installed
---> Package gpfs.gpl.noarch 0:4.1.0-0 will be installed
---> Package gpfs.gskit.ppc64 0:8.0.50-16 will be installed
---> Package gpfs.msg.en_US.noarch 0:4.1.0-0 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
================================================================================
=================
 Package                                 Arch
Version                         Repository
Size
```

```
================================================================================
================================================================================
=================
Installing:
 gpfs.base                                  ppc64
4.1.0-0                     /gpfs.base-4.1.0-0.ppc64
45 M
 gpfs.docs                                  noarch
4.1.0-0                     /gpfs.docs-4.1.0-0.noarch
1.3 M
 gpfs.ext                                   ppc64
4.1.0-0                     /gpfs.ext-4.1.0-0.ppc64
4.9 M
 gpfs.gpl                                   noarch
4.1.0-0                     /gpfs.gpl-4.1.0-0.noarch
2.3 M
 gpfs.gskit                                 ppc64
8.0.50-16                   /gpfs.gskit-8.0.50-16.ppc64
16 M
 gpfs.msg.en_US                             noarch
4.1.0-0                     /gpfs.msg.en_US-4.1.0-0.noarch
503 k

Transaction Summary
================================================================================
================================================================================
=================
Install       6 Package(s)

Total size: 70 M
Installed size: 70 M
Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
 Installing : gpfs.base-4.1.0-0.ppc64
1/6
/sbin/ldconfig: /usr/lib64/libhdfs.so.0 is not a symbolic link

/sbin/ldconfig: /usr/lib64/libhadoop.so.1 is not a symbolic link

 Installing : gpfs.ext-4.1.0-0.ppc64
2/6
 Installing : gpfs.gpl-4.1.0-0.noarch
3/6
 Installing : gpfs.msg.en_US-4.1.0-0.noarch
4/6
 Installing : gpfs.docs-4.1.0-0.noarch
5/6
 Installing : gpfs.gskit-8.0.50-16.ppc64
6/6
 Verifying  : gpfs.ext-4.1.0-0.ppc64
1/6
```

```
 Verifying : gpfs.gskit-8.0.50-16.ppc64
2/6
 Verifying : gpfs.base-4.1.0-0.ppc64
3/6
 Verifying : gpfs.docs-4.1.0-0.noarch
4/6
 Verifying : gpfs.gpl-4.1.0-0.noarch
5/6
 Verifying : gpfs.msg.en_US-4.1.0-0.noarch
6/6

Installed:
  gpfs.base.ppc64 0:4.1.0-0  gpfs.docs.noarch 0:4.1.0-0  gpfs.ext.ppc64
0:4.1.0-0  gpfs.gpl.noarch 0:4.1.0-0  gpfs.gskit.ppc64 0:8.0.50-16
gpfs.msg.en_US.noarch 0:4.1.0-0

Complete!
[root@symphony1 gpfs_power]#
```

5. Update the version of Spectrum Scale to 4.1.0.1 as shown in Example 4-26.

Example 4-26 Update Spectrum Scale

```
[root@symphony1 gpfs_power]# cd 4101/
[root@symphony1 4101]# tar -xvf GPFS-4.1.0.1-power-Linux.advanced.tar.gz
changelog
gpfs.base-4.1.0-1.ppc64.update.rpm
gpfs.crypto-4.1.0-1.ppc64.update.rpm
gpfs.docs-4.1.0-1.noarch.rpm
gpfs.ext-4.1.0-1.ppc64.update.rpm
gpfs.gpl-4.1.0-1.noarch.rpm
gpfs.msg.en_US-4.1.0-1.noarch.rpm
README
[root@symphony1 4101]# yum update *.rpm
Loaded plugins: product-id, security, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use
subscription-manager to register.
Setting up Update Process
Examining gpfs.base-4.1.0-1.ppc64.update.rpm: gpfs.base-4.1.0-1.ppc64
Marking gpfs.base-4.1.0-1.ppc64.update.rpm as an update to
gpfs.base-4.1.0-0.ppc64
Examining gpfs.crypto-4.1.0-1.ppc64.update.rpm: gpfs.crypto-4.1.0-1.ppc64
Package gpfs.crypto not installed, cannot update it. Run yum install to install
it instead.
Examining gpfs.docs-4.1.0-1.noarch.rpm: gpfs.docs-4.1.0-1.noarch
Marking gpfs.docs-4.1.0-1.noarch.rpm as an update to gpfs.docs-4.1.0-0.noarch
Examining gpfs.ext-4.1.0-1.ppc64.update.rpm: gpfs.ext-4.1.0-1.ppc64
Marking gpfs.ext-4.1.0-1.ppc64.update.rpm as an update to
gpfs.ext-4.1.0-0.ppc64
Examining gpfs.gpl-4.1.0-1.noarch.rpm: gpfs.gpl-4.1.0-1.noarch
Marking gpfs.gpl-4.1.0-1.noarch.rpm as an update to gpfs.gpl-4.1.0-0.noarch
Examining gpfs.msg.en_US-4.1.0-1.noarch.rpm: gpfs.msg.en_US-4.1.0-1.noarch
Marking gpfs.msg.en_US-4.1.0-1.noarch.rpm as an update to
gpfs.msg.en_US-4.1.0-0.noarch
Resolving Dependencies
--> Running transaction check
```

```
---> Package gpfs.base.ppc64 0:4.1.0-0 will be updated
---> Package gpfs.base.ppc64 0:4.1.0-1 will be an update
---> Package gpfs.docs.noarch 0:4.1.0-0 will be updated
---> Package gpfs.docs.noarch 0:4.1.0-1 will be an update
---> Package gpfs.ext.ppc64 0:4.1.0-0 will be updated
---> Package gpfs.ext.ppc64 0:4.1.0-1 will be an update
---> Package gpfs.gpl.noarch 0:4.1.0-0 will be updated
---> Package gpfs.gpl.noarch 0:4.1.0-1 will be an update
---> Package gpfs.msg.en_US.noarch 0:4.1.0-0 will be updated
---> Package gpfs.msg.en_US.noarch 0:4.1.0-1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

===============================================================================
===============================================================================
=================
 Package                                 Arch
Version                      Repository
Size
===============================================================================
===============================================================================
=================
Updating:
 gpfs.base                               ppc64
4.1.0-1                     /gpfs.base-4.1.0-1.ppc64.update
45 M
 gpfs.docs                               noarch
4.1.0-1                      /gpfs.docs-4.1.0-1.noarch
1.3 M
 gpfs.ext                                ppc64
4.1.0-1                     /gpfs.ext-4.1.0-1.ppc64.update
4.9 M
 gpfs.gpl                                noarch
4.1.0-1                      /gpfs.gpl-4.1.0-1.noarch
2.3 M
 gpfs.msg.en_US                          noarch
4.1.0-1                     /gpfs.msg.en_US-4.1.0-1.noarch
507 k

Transaction Summary
===============================================================================
===============================================================================
=================
Upgrade       5 Package(s)

Total size: 54 M
Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
 Updating : gpfs.base-4.1.0-1.ppc64
1/10
```

```
/sbin/ldconfig: /usr/lib64/libhdfs.so.0 is not a symbolic link

/sbin/ldconfig: /usr/lib64/libhadoop.so.1 is not a symbolic link

 Updating : gpfs.ext-4.1.0-1.ppc64
2/10
 Updating : gpfs.gpl-4.1.0-1.noarch
3/10
 Updating : gpfs.docs-4.1.0-1.noarch
4/10
 Updating : gpfs.msg.en_US-4.1.0-1.noarch
5/10
 Cleanup  : gpfs.gpl-4.1.0-0.noarch
6/10
 Cleanup  : gpfs.docs-4.1.0-0.noarch
7/10
 Cleanup  : gpfs.msg.en_US-4.1.0-0.noarch
8/10
 Cleanup  : gpfs.ext-4.1.0-0.ppc64
9/10
 Cleanup  : gpfs.base-4.1.0-0.ppc64
10/10
 Verifying : gpfs.base-4.1.0-1.ppc64
1/10
 Verifying : gpfs.msg.en_US-4.1.0-1.noarch
2/10
 Verifying : gpfs.ext-4.1.0-1.ppc64
3/10
 Verifying : gpfs.gpl-4.1.0-1.noarch
4/10
 Verifying : gpfs.docs-4.1.0-1.noarch
5/10
 Verifying : gpfs.ext-4.1.0-0.ppc64
6/10
 Verifying : gpfs.base-4.1.0-0.ppc64
7/10
 Verifying : gpfs.docs-4.1.0-0.noarch
8/10
 Verifying : gpfs.msg.en_US-4.1.0-0.noarch
9/10
 Verifying : gpfs.gpl-4.1.0-0.noarch
10/10

Updated:
  gpfs.base.ppc64 0:4.1.0-1        gpfs.docs.noarch 0:4.1.0-1
gpfs.ext.ppc64 0:4.1.0-1        gpfs.gpl.noarch 0:4.1.0-1
gpfs.msg.en_US.noarch 0:4.1.0-1

Complete!
[root@symphony1 4101]#
```

6. Now, install the Spectrum Scale open source portability layer as shown in Example 4-27. On Linux platforms, Spectrum Scale uses a loadable kernel module that enables the Spectrum Scale daemon to interact with the Linux kernel. Source code is provided for the portability layer so that the Spectrum Scale portability can be built and installed on various

Linux kernel versions and configurations. When Spectrum Scale is installed on Linux, you must build a portability module that is based on your particular hardware platform and Linux distribution to enable communication between the Linux kernel and Spectrum Scale.

Example 4-27 Spectrum Scale open source portability layer building

```
[root@symphony1 4101]# cd /usr/lpp/mmfs/src/
[root@symphony1 src]# make Autoconfig
cd /usr/lpp/mmfs/src/config; ./configure --genenvonly; if [ $? -eq 0 ]; then
/usr/bin/cpp -P def.mk.proto > ./def.mk; exit $? || exit 1; else exit $?; fi
[root@symphony1 src]# make World
Verifying that tools to build the portability layer exist....
cpp present
gcc present
g++ present
ld present
.......
/usr/bin/install -c -m 0500 lxtrace /usr/lpp/mmfs/src/bin/lxtrace-`cat
//usr/lpp/mmfs/src/gpl-linux/gpl_kernel.tmp.ver`
/usr/bin/install -c -m 0500 kdump   /usr/lpp/mmfs/src/bin/kdump-`cat
//usr/lpp/mmfs/src/gpl-linux/gpl_kernel.tmp.ver`
make[1]: Leaving directory `/usr/lpp/mmfs/src/gpl-linux'
[root@symphony1 src]# make InstallImages
(cd gpl-linux; /usr/bin/make InstallImages; \
   exit $?) || exit 1
make[1]: Entering directory `/usr/lpp/mmfs/src/gpl-linux'
Pre-kbuild step 1...
make[2]: Entering directory `/usr/src/kernels/2.6.32-431.el6.ppc64'
  INSTALL /usr/lpp/mmfs/src/gpl-linux/kdump-kern-dummy.ko
  INSTALL /usr/lpp/mmfs/src/gpl-linux/kdump-kern-dwarfs.ko
  INSTALL /usr/lpp/mmfs/src/gpl-linux/mmfs26.ko
  INSTALL /usr/lpp/mmfs/src/gpl-linux/mmfslinux.ko
  INSTALL /usr/lpp/mmfs/src/gpl-linux/tracedev.ko
  DEPMOD  2.6.32-431.el6.ppc64
make[2]: Leaving directory `/usr/src/kernels/2.6.32-431.el6.ppc64'
make[1]: Leaving directory `/usr/lpp/mmfs/src/gpl-linux'
[root@symphony1 src]# make rpm
rm -rf /tmp/rpm
rpmbuild --define "MODKERNEL `cat
//usr/lpp/mmfs/src/gpl-linux/gpl_kernel.tmp.ver`" --define "GPLDIR
/usr/lpp/mmfs/src/gpl-linux" -bb /usr/lpp/mmfs/src/config/gpfs.gplbin.spec
--buildroot=/tmp/rpm
Executing(%install): /bin/sh -e /var/tmp/rpm-tmp.U2DARq
+ umask 022
+ cd /root/rpmbuild/BUILD
+ '[' /tmp/rpm '!=' / ']'
+ rm -rf /tmp/rpm
++ dirname /tmp/rpm
+ mkdir -p /tmp
+ mkdir /tmp/rpm
+ LANG=C
+ export LANG
+ unset DISPLAY
+ '[' x/tmp/rpm = x ']'
+ '[' '' '!=' Linux ']'
+ export GPFS_LPP=/tmp/rpm/usr/lpp/mmfs
```

```
+ GPFS_LPP=/tmp/rpm/usr/lpp/mmfs
+ export GPFS_KO=/tmp/rpm/lib/modules/2.6.32-431.el6.ppc64/extra
+ GPFS_KO=/tmp/rpm/lib/modules/2.6.32-431.el6.ppc64/extra
+ export GPFS_BIN=/tmp/rpm/usr/lpp/mmfs/bin
+ GPFS_BIN=/tmp/rpm/usr/lpp/mmfs/bin
+ export LPPBIN=/usr/lpp/mmfs/src/bin
+ LPPBIN=/usr/lpp/mmfs/src/bin
+ install -d -m 755 /tmp/rpm/usr/lpp/mmfs
+ install -d -m 755 /tmp/rpm/usr/lpp/mmfs/bin
+ install -d -m 755 /tmp/rpm/lib/modules/2.6.32-431.el6.ppc64/extra
+ install -m 500 /usr/lpp/mmfs/src/bin/kdump-2.6.32-431.el6.ppc64
/tmp/rpm/usr/lpp/mmfs/bin/kdump-2.6.32-431.el6.ppc64
+ install -m 500 /usr/lpp/mmfs/src/bin/lxtrace-2.6.32-431.el6.ppc64
/tmp/rpm/usr/lpp/mmfs/bin/lxtrace-2.6.32-431.el6.ppc64
+ install -m 500 /usr/lpp/mmfs/src/gpl-linux/mmfs26.ko
/tmp/rpm/lib/modules/2.6.32-431.el6.ppc64/extra
+ install -m 500 /usr/lpp/mmfs/src/gpl-linux/mmfslinux.ko
/tmp/rpm/lib/modules/2.6.32-431.el6.ppc64/extra
+ install -m 500 /usr/lpp/mmfs/src/gpl-linux/tracedev.ko
/tmp/rpm/lib/modules/2.6.32-431.el6.ppc64/extra
+ exit 0
Processing files: gpfs.gplbin-2.6.32-431.el6.ppc64-4.1.0-1.ppc64
Checking for unpackaged file(s): /usr/lib/rpm/check-files /tmp/rpm
Wrote:
/root/rpmbuild/RPMS/ppc64/gpfs.gplbin-2.6.32-431.el6.ppc64-4.1.0-1.ppc64.rpm
Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp.Y54o5A
+ umask 022
+ cd /root/rpmbuild/BUILD
+ /bin/rm -rf /tmp/rpm
+ exit 0
[root@symphony1 src]#
[root@symphony1 4101]# yum install
gpfs.gplbin-2.6.32-431.el6.ppc64-4.1.0-1.ppc64.rpm
Loaded plugins: product-id, security, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use
subscription-manager to register.
Setting up Install Process
Examining gpfs.gplbin-2.6.32-431.el6.ppc64-4.1.0-1.ppc64.rpm:
gpfs.gplbin-2.6.32-431.el6.ppc64-4.1.0-1.ppc64
Marking gpfs.gplbin-2.6.32-431.el6.ppc64-4.1.0-1.ppc64.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package gpfs.gplbin-2.6.32-431.el6.ppc64.ppc64 0:4.1.0-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

===============================================================================
===============================================================================
=================
 Package                                             Arch
Version                Repository
Size
```

```
================================================================================
================================================================================
=================
Installing:
 gpfs.gplbin-2.6.32-431.el6.ppc64                   ppc64
4.1.0-1            /gpfs.gplbin-2.6.32-431.el6.ppc64-4.1.0-1.ppc64
4.4 M

Transaction Summary
================================================================================
================================================================================
=================
Install       1 Package(s)

Total size: 4.4 M
Installed size: 4.4 M
Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
 Installing:gpfs.gplbin-2.6.32-431.el6.ppc64-4.1.0-1.ppc64
1/1
 Verifying:gpfs.gplbin-2.6.32-431.el6.ppc64-4.1.0-1.ppc64
1/1

Installed:
 gpfs.gplbin-2.6.32-431.el6.ppc64.ppc64 0:4.1.0-1

Complete!
[root@symphony1 4101]#
```

4.5.2 Cluster configuration steps

The following are the cluster configuration steps:

1. Set the key authentication method on all servers because the servers need to communicate with each other. The following command creates the private-public keypairs:

   `ssh-keygen -b 2048`

   The command generates the keys as shown in Example 4-28.

   Example 4-28 SSH keypairs

```
[root@pw4302-l2 ~]# ll /root/.ssh/
total 16
-rw-r--r--  1 root root  397 Dec  5 18:37 authorized_keys
-rw-------. 1 root root 1675 Nov 26 14:40 id_rsa
-rw-r--r--. 1 root root  396 Nov 26 14:40 id_rsa.pub
-rw-r--r--. 1 root root 4751 Dec  1 17:11 known_hosts
[root@pw4302-l2 ~]#
```

Note: Copy the `/root/.ssh` directory into all nodes.

2. Now you are able to configure the cluster on the master node as shown in Example 4-29.

Example 4-29 Cluster creation

```
[root@pw4302-12 .ssh]# mmcrcluster -N
pw4302-12:quorum,pw4302-11:quorum,pw4302-15:quorum,pw4302-16,symphony1,symphony
2 -r /usr/bin/ssh -R /usr/bin/scp
mmcrcluster: Performing preliminary node verification ...
mmcrcluster: Processing quorum and other critical nodes ...
mmcrcluster: Finalizing the cluster data structures ...
mmcrcluster: Command successfully completed
mmcrcluster: Warning: Not all nodes have proper GPFS license designations.
    Use the mmchlicense command to designate licenses as needed.
mmcrcluster: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
[root@pw4302-12 .ssh]#
```

3. Accept the licenses. On quorum servers, use the server license. On file placement optimizer servers, use the FPO license, as shown in Example 4-30.

Example 4-30 Cluster license accept

```
[root@pw4302-12 /]# mmchlicense server --accept -N
pw4302-12,pw4302-11,pw4302-15

The following nodes will be designated as possessing Spectrum Scale server
licenses:
        pw4302-12
        pw4302-11
        pw4302-15
mmchlicense: Command successfully completed
mmchlicense: Warning: Not all nodes have proper GPFS license designations.
    Use the mmchlicense command to designate licenses as needed.
mmchlicense: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
[root@pw4302-12 /]# mmchlicense fpo --accept -N pw4302-16,symphony1,symphony2

The following nodes will be designated as possessing GPFS fpo licenses:
        pw4302-16
        symphony1
        symphony2
mmchlicense: Command successfully completed
mmchlicense: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
[root@pw4302-12 /]#
```

4. Start your cluster as shown in Example 4-31.

Example 4-31 Start the cluster

```
[root@pw4302-12 .ssh]# mmstartup -a
Tue Dec  2 12:23:43 EST 2014: mmstartup: Starting GPFS ...
[root@pw4302-12 .ssh]#
```

5. Check your cluster status. You can see the cluster output in Example 4-32.

Example 4-32 Built cluster

```
[root@pw4302-l2 ~]# mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         pw4302-l2
  GPFS cluster id:           2476174979729679987
  GPFS UID domain:           pw4302-l2
  Remote shell command:      /usr/bin/ssh
  Remote file copy command:  /usr/bin/scp
  Repository type:           CCR

 Node  Daemon node name            IP address     Admin node name  Designation
--------------------------------------------------------------------------------
-
   1   pw4302-l2                   172.16.20.161  pw4302-l2        quorum
   2   pw4302-l1                   172.16.20.160  pw4302-l1        quorum
   3   pw4302-l5                   172.16.20.164  pw4302-l5        quorum
   4   pw4302-l6                   172.16.20.165  pw4302-l6
   5   symphony1                   172.16.21.162  symphony1
   6   symphony2                   172.16.21.163  symphony2

[root@pw4302-l2 ~]# mmgetstate -a

 Node number  Node name        GPFS state
------------------------------------------
       1      pw4302-l2        active
       2      pw4302-l1        active
       3      pw4302-l5        active
       4      pw4302-l6        active
       5      symphony1        active
       6      symphony2        active
[root@pw4302-l2 ~]#
```

4.5.3 Spectrum Scale network shared disk creation

The following steps help to create the network shared disks (NSDs):

1. We use the stanza file, as shown in Example 4-33, to create the NSDs.

Example 4-33 Stanza file for NSDs

```
[root@pw4302-l2 .ssh]# mmcrnsd -F /tmp/config
mmcrnsd: Processing disk sdd
mmcrnsd: Processing disk sdd
mmcrnsd: Processing disk sdd
mmcrnsd: Processing disk sdb
mmcrnsd: Processing disk sdb
mmcrnsd: Processing disk sdb
mmcrnsd: Processing disk sdc
mmcrnsd: Processing disk sdc
mmcrnsd: Processing disk sdc
mmcrnsd: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
[root@pw4302-l2 .ssh]#
```

2. The next step is to create a file system on the disk as shown in Example 4-34.

Example 4-34 File system creation on NSDs

```
[root@pw4302-l2 ~]# mmcrfs bigdatafs -F /tmp/config -A yes -B 1024K -j cluster
-m 3 -M 3 -r 3 -R 3 -T /gpfs_fpo

The following disks of bigdatafs will be formatted on node pw4302-l2:
  gpfs11nsd: size 1024 MB
  gpfs12nsd: size 1024 MB
  gpfs13nsd: size 1024 MB
  gpfs14nsd: size 10240 MB
  gpfs15nsd: size 10240 MB
  gpfs16nsd: size 10240 MB
  gpfs19nsd: size 10240 MB
    gpfs17nsd: size 10240 MB
    gpfs18nsd: size 10240 MB
Formatting file system ...
Disks up to size 52 GB can be added to storage pool system.
Disks up to size 103 GB can be added to storage pool datapool.
Creating Inode File
Creating Allocation Maps
Creating Log Files
Clearing Inode Allocation Map
Clearing Block Allocation Map
Formatting Allocation Map for storage pool system
Formatting Allocation Map for storage pool datapool
Completed creation of file system /dev/bigdatafs.
mmcrfs: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
[root@pw4302-l2 ~]#
```

3. Mount the file systems as shown in Example 4-35.

Example 4-35 File systems mount

```
[root@pw4302-l2 .ssh]# mmmount /gpfs_fpo -a
Tue Dec  2 13:46:12 EST 2014: mmmount: Mounting file systems ...
[root@pw4302-l2 .ssh]#
```

4. Check your configuration as shown in Example 4-36.

Example 4-36 Configuration check

```
[root@pw4302-l2 ~]# mmlsdisk bigdatafs -L
disk         driver   sector     failure holds    holds
storage
name           type       size        group metadata data  status
availability disk id pool          remarks
------------ -------- ------ ----------- -------- ----- -------------
------------ ------- ------------ ---------
gpfs11nsd    nsd       512          1 yes      no    ready        up
1 system        desc
gpfs12nsd    nsd       512          2 yes      no    ready        up
2 system        desc
gpfs13nsd    nsd       512          3 yes      no    ready        up
3 system        desc
```

```
gpfs14nsd   nsd       512      1,0,0 no      yes   ready        up
4 datapool
gpfs15nsd   nsd       512      2,0,0 no      yes   ready        up
5 datapool
gpfs16nsd   nsd       512      3,0,0 no      yes   ready        up
6 datapool
gpfs19nsd   nsd       512      1,1,0 no      yes   ready        up
7 datapool      desc
gpfs17nsd   nsd       512      2,1,0 no      yes   ready        up
8 datapool      desc
gpfs18nsd   nsd       512      3,1,0 no      yes   ready        up
9 datapool
Number of quorum disks: 5
Read quorum value:      3
Write quorum value:     3
[root@pw4302-12 ~]# mmlspool bigdatafs all -L
Pool:
  name                   = system
  poolID                 = 0
  blockSize              = 128 KB
  usage                  = metadataOnly
  maxDiskSize            = 48 GB
  layoutMap              = cluster
  allowWriteAffinity     = no
  writeAffinityDepth     = 0
  blockGroupFactor       = 1

Pool:
  name                   = datapool
  poolID                 = 65537
  blockSize              = 1024 KB
  usage                  = dataOnly
  maxDiskSize            = 575 GB
  layoutMap              = cluster
  allowWriteAffinity     = yes
  writeAffinityDepth     = 1
  blockGroupFactor       = 128

[root@pw4302-12 ~]#
```

5. Now, create Spectrum Scale Data Placement Policy and enable it on all nodes. The steps are shown in Example 4-37.

Example 4-37   Data placement policy

```
[root@pw4302-12 ~]# cat /tmp/policydata
rule default SET POOL 'datapool'
[root@pw4302-12 ~]#
[root@pw4302-12 ~]# mmchpolicy bigdatafs /tmp/policydata -I yes
Validated policy `policydata': Parsed 1 policy rules.
Policy `policydata' installed and broadcast to all nodes.
[root@pw4302-12 ~]#
```

6. Create the **mk_gpfs_local_dirs.sh** script to create the directory tree on all nodes. In the master host, the /Nodes file contains the machine names. We set *WCOLL* (this is the variable that contains the node names for Spectrum Scale) and test the functionality with a **date** command. Example 4-38 shows the steps.

Example 4-38 Directory tree creation

```
[root@pw4302-l2 tmp]# export WCOLL=/Nodes
[root@pw4302-l2 tmp]# mmdsh date
pw4302-l6:  Tue Dec  2 14:23:11 EST 2014
symphony1:  Tue Dec  2 14:10:55 EST 2014
symphony2:  Tue Dec  2 14:10:55 EST 2014
pw4302-l1:  Tue Dec  2 14:23:16 EST 2014
pw4302-l5:  Tue Dec  2 14:23:19 EST 2014
pw4302-l2:  Tue Dec  2 14:23:19 EST 2014
root@pw4302-l2 tmp]# cat mk_gpfs_local_dirs.sh
#!/bin/sh
for nodename in $(mmlsnode -N all); do
mkdir -p /gpfs_fpo/tmp/${nodename}
mkdir -p /gpfs_fpo/mapred/local/${nodename}
done
[root@pw4302-l2 tmp]# sh mk_gpfs_local_dirs.sh
[root@pw4302-l2 tmp]# ll /gpfs_fpo
total 0
drwxr-xr-x. 3 root root 4096 Dec  2 14:15 mapred
drwxr-xr-x. 8 root root 4096 Dec  2 14:15 tmp
[root@pw4302-l2 tmp]#
```

7. Check the replication settings as shown in Example 4-39.

Example 4-39 Replication settings

```
[root@pw4302-l2 ~]# mmlsattr /gpfs_fpo/input/
  replication factors
metadata(max) data(max) file    [flags]
------------- --------- ---------------
      3 (  3)   3 (  3) /gpfs_fpo/input/
[root@pw4302-l2 ~]#
```

Now Spectrum Scale is ready for MapReduce jobs.

4.6 Hadoop configuration

This topic shows how to configure Hadoop to perform MapReduce jobs. The configuration files for this scenario are shown in Example 4-40.

Example 4-40 Hadoop configuration files

```
[root@pw4302-l2 install_data]# cat /etc/hadoop/masters
pw4302-l2
[root@pw4302-l2 install_data]# cat /etc/hadoop/slaves
pw4302-l1
pw4302-l5
pw4302-l6
symphony1
```

```
symphony2
[root@pw4302-l2 ~]# cat /etc/hadoop/hadoop-env.sh|grep -v "^#\|^$"
export JAVA_HOME=/usr/java/default
export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}
export HADOOP_OPTS="-Djava.net.preferIPv4Stack=true $HADOOP_CLIENT_OPTS"
export HADOOP_NAMENODE_OPTS="-Dhadoop.security.logger=INFO,DRFAS
-Dhdfs.audit.logger=INFO,DRFAAUDIT $HADOOP_NAMENODE_OPTS"
HADOOP_JOBTRACKER_OPTS="-Dhadoop.security.logger=INFO,DRFAS
-Dmapred.audit.logger=INFO,MRAUDIT -Dhadoop.mapreduce.jobsummary.logger=INFO,JSA
$HADOOP_JOBTRACKER_OPTS"
HADOOP_TASKTRACKER_OPTS="-Dhadoop.security.logger=ERROR,console
-Dmapred.audit.logger=ERROR,console $HADOOP_TASKTRACKER_OPTS"
HADOOP_DATANODE_OPTS="-Dhadoop.security.logger=ERROR,DRFAS $HADOOP_DATANODE_OPTS"
export HADOOP_SECONDARYNAMENODE_OPTS="-Dhadoop.security.logger=INFO,DRFAS
-Dhdfs.audit.logger=INFO,DRFAAUDIT $HADOOP_SECONDARYNAMENODE_OPTS"
export HADOOP_CLIENT_OPTS="-Xmx2048m $HADOOP_CLIENT_OPTS"
export HADOOP_SECURE_DN_USER=
export HADOOP_LOG_DIR=/var/log/hadoop/$USER
export HADOOP_SECURE_DN_LOG_DIR=/var/log/hadoop/
export HADOOP_PID_DIR=/var/run/hadoop
export HADOOP_SECURE_DN_PID_DIR=/var/run/hadoop
export HADOOP_IDENT_STRING=$USER
[root@pw4302-l2 ~]# cat /etc/hadoop/core-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
      <name>fs.default.name</name>
      <value>gpfs:///</value>
    </property>
    <property>
      <name>fs.gpfs.impl</name>
      <value>org.apache.hadoop.fs.gpfs.GlobalParallelFileSystem</value>
    </property>
    <property>
      <name>gpfs.mount.dir</name>
      <value>/gpfs_fpo</value>
    </property>

<property>
<name>hadoop.tmp.dir</name>
<value>/var/gpfs/tmp_link</value>
</property>

</configuration>
[root@pw4302-l2 ~]# cat /etc/hadoop/mapred-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
```

```
<property>
<name>mapred.local.dir</name>
<value>/var/gpfs/tmp_link</value>
</property>


</configuration>
[root@pw4302-l2 ~]# cat /etc/hadoop/hdfs-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
<name>dfs.block.size</name>
<value>134217728</value><!--128MB-->
</property>

<property>
<name>dfs.replication</name>
<value>3</value>
</property>


</configuration>
```

Note: Even though HDFS is not used in a Spectrum Scale deployment, some values in hdfs-site.xml file are still used by the Spectrum Scale connector, if set. As a best practice, set the value of `dfs.block.size` to `blockSize * blockGroupFactor` for the data pool.

The value of `dfs.block.size` is in bytes, so convert from MB by dividing by 1024^2. This should be 128 MB = 134,217,728 bytes.

During our test, there was a symlink pointing to an incorrect library file, and it caused the error shown in Example 4-41 during our test.

Example 4-41 Error message during a Hadoop test

```
mrsh.printAPIVersion.1_1_1
java.lang.UnsatisfiedLinkError:
org.apache.hadoop.fs.gpfs.GlobalParallelFileSystem.nGetInitInfo()Ljava/lang/String
;
at org.apache.hadoop.fs.gpfs.GlobalParallelFileSystem.nGetInitInfo(Native Method)
at
org.apache.hadoop.fs.gpfs.GlobalParallelFileSystem.initialize(GlobalParallelFileSy
stem.java:256)
at
org.apache.hadoop.fs.gpfs.GlobalParallelFileSystem.initialize(GlobalParallelFileSy
stem.java:206)
at org.apache.hadoop.fs.FileSystem.createFileSystem(FileSystem.java:1549)
at org.apache.hadoop.fs.FileSystem.access$200(FileSystem.java:70)
at org.apache.hadoop.fs.FileSystem$Cache.get(FileSystem.java:1567)
```

```
at org.apache.hadoop.fs.FileSystem.get(FileSystem.java:360)
at org.apache.hadoop.fs.Path.getFileSystem(Path.java:187)
at
org.apache.hadoop.mapreduce.lib.input.FileInputFormat.addInputPath(FileInputFormat
.java:372)
at org.apache.hadoop.examples.WordCount.main(WordCount.java:65)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
at java.lang.reflect.Method.invoke(Unknown Source)
at
org.apache.hadoop.util.ProgramDriver$ProgramDescription.invoke(ProgramDriver.java:
68)
at org.apache.hadoop.util.ProgramDriver.driver(ProgramDriver.java:139)
at org.apache.hadoop.examples.ExampleDriver.main(ExampleDriver.java:64)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
at java.lang.reflect.Method.invoke(Unknown Source)
at org.apache.hadoop.util.RunJar.main(RunJar.java:156)
```

The issue is resolved with the following symlink command, which must be run in all nodes, as shown in Example 4-42.

```
ln -fs /usr/lpp/mmfs/lib/libgpfs.so /lib/libgpfs.so
```

Example 4-42 Issue correction

```
[root@pw4302-l2 ~]# ls -la /lib/libgpfs.so
lrwxrwxrwx 1 root root 28 Dec  3 16:03 /lib/libgpfs.so ->
/usr/lpp/mmfs/lib/libgpfs.so
[root@pw4302-l2 ~]# ls -la /usr/lpp/mmfs/lib/libgpfs.so
-r-xr-xr-x 1 root root 69659 Jun  5  2014 /usr/lpp/mmfs/lib/libgpfs.so
[root@pw4302-l2 ~]#
```

In the bare installation, it was linked as shown in Example 4-43.

Example 4-43 Bare install config file

```
lrwxrwxrwx  1 root root 28 Dec  3 16:03 /lib/libgpfs.so ->
/usr/lpp/mmfs/lib/i386/libgpfs.so
```

Example 4-44 shows our `/usr/lib64` libraries that are related to Hadoop.

Example 4-44 Libraries related to Hadoop

```
[root@pw4302-l2 ~]# ls -la /usr/lib64/libhadoop*
-rw-r--r--. 1 root root  61694 Nov 19  2012 /usr/lib64/libhadoop.a
-rw-r--r--. 1 root root    881 Nov 19  2012 /usr/lib64/libhadoop.la
-rw-r--r--. 1 root root 394212 Nov 19  2012 /usr/lib64/libhadooppipes.a
-rw-r--r--. 1 root root 223277 Nov 19  2012 /usr/lib64/libhadoop.so
lrwxrwxrwx. 1 root root     27 Dec  3 13:19 /usr/lib64/libhadoop.so.1 ->
libhadoop.so.1.0.0-original
-rw-r--r--. 1 root root 223277 Nov 19  2012 /usr/lib64/libhadoop.so.1.0.0
-rw-r--r--. 1 root root 223277 Nov 19  2012 /usr/lib64/libhadoop.so.1.0.0-original
-rw-r--r--. 1 root root 104488 Nov 19  2012 /usr/lib64/libhadooputils.a
```

```
[root@pw4302-12 ~]#
```

We created symbolic links on all nodes, as shown in Example 4-45.

Example 4-45 Symbolic links for IBM Platform Symphony

```
mmdsh "ln -s /usr/lpp/mmfs/fpo/hadoop-1.1.1/*.jar $HADOOP_HOME/lib"
mmdsh "ln -s /usr/lpp/mmfs/fpo/hadoop-1.1.1/libgpfshadoop.64.so
$HADOOP_HOME/lib/libgpfshadoop.so"
mmdsh "ln -s /usr/lpp/mmfs/lib/libgpfs.so $HADOOP_HOME/lib/libgpfs.so"
```

Example 4-47 shows the results after linking the `$HADOOP_HOME/lib` directory.

Example 4-46 Hadoop libs

```
[root@pw4302-12 ~]# ssh symphony1 ls -la /usr/share/hadoop/lib/|grep -i ^l
lrwxrwxrwx 1 root root      52 Dec  2 14:42 hadoop-1.1.1-gpfs.jar ->
/usr/lpp/mmfs/fpo/hadoop-1.1.1/hadoop-1.1.1-gpfs.jar
lrwxrwxrwx 1 root root      50 Dec  2 14:43 libgpfshadoop.so ->
/usr/lpp/mmfs/fpo/hadoop-1.1.1/libgpfshadoop.64.so
lrwxrwxrwx 1 root root      28 Dec  2 14:43 libgpfs.so ->
/usr/lpp/mmfs/lib/libgpfs.so
[root@pw4302-12 ~]#
```

Create the link on all nodes as `egoadmin`, as shown in Example 4-47.

Example 4-47 Link creation as egoadmin

```
[root@pw4302-12 ~]# su - egoadmin
[egoadmin@pw4302-12 ~]$ . /opt/ibm/platformsymphony/profile.platform
[egoadmin@pw4302-12 ~]$ ln -s /usr/lpp/mmfs/fpo/hadoop-1.1.1/*.jar
$PMR_SERVERDIR/../lib
[egoadmin@pw4302-12 ~]$
```

Use symbolic links with the same path on each node, pointing to that node's unique directories. Example 4-48 assumes that the symbolic link paths are `/var/gpfs/tmp_link` and `/var/gpfs/mapred/local/`*`<hostnames>`*. Check that the **`hostname`** command outputs the same value as the directory being linked to. We run it on all nodes with the **`mmdsh`** command, as shown on Example 4-48.

Example 4-48 Directory structure

```
[root@pw4302-12 tmp]# mmdsh mkdir -p /var/gpfs
[root@pw4302-12 tmp]#
[root@pw4302-12 tmp]# mmdsh ln -s /gpfs_fpo/tmp/$(hostname) /var/gpfs/tmp_link
[root@pw4302-12 tmp]# mmdsh ln -s /gpfs_fpo/mapred/local/$(hostname)
/var/gpfs/mapred_local_link
[root@pw4302-12 tmp]#
```

4.7 MapReduce test with Hadoop Wordcount in IBM Platform Symphony 7.1

For testing, we create a huge file and multiply it with the following command:

```
while [ $count -lt 51 ]; do cat sg247975.txt>>newbook; let count=$count+1; done
```

The command creates a book of approximately 45000 pages.

You can submit MapReduce jobs on the Platform Symphony webpage. For this example:

http://172.16.20.161:8080/platform

After you log in to the console, from the menus, select **Workload** → **Mapreduce** → **Jobs**, as shown in Figure 4-5.

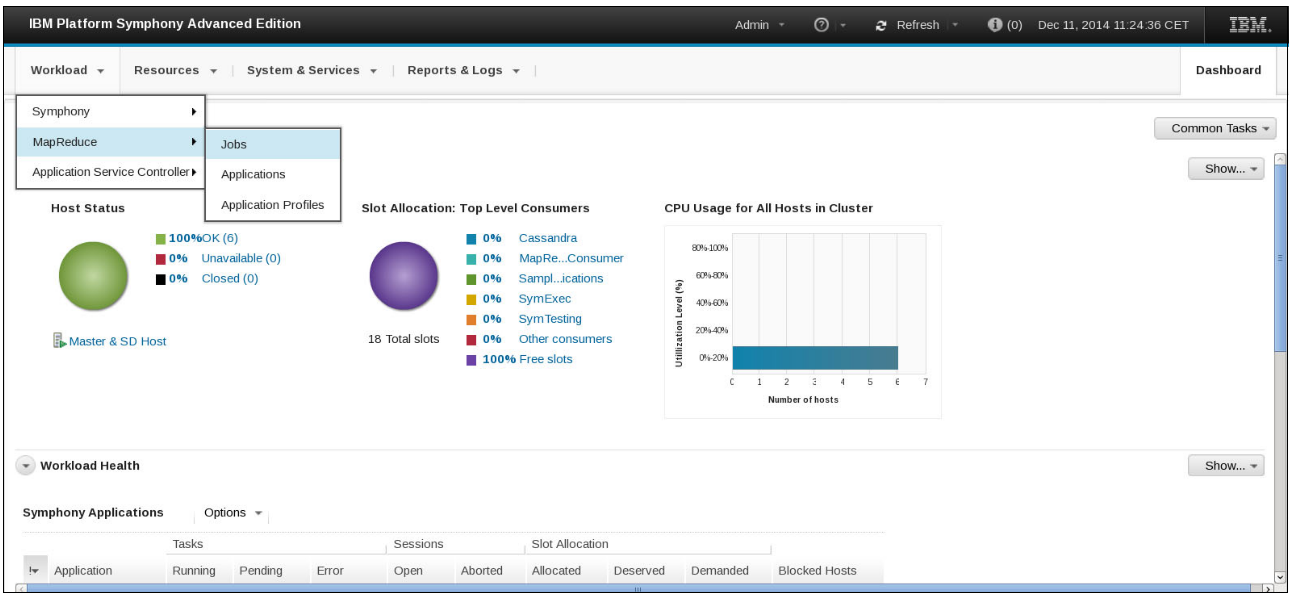


Figure 4-5 MapReduce jobs menu

Select **New** to open the Submit Job window shown in Figure 4-6

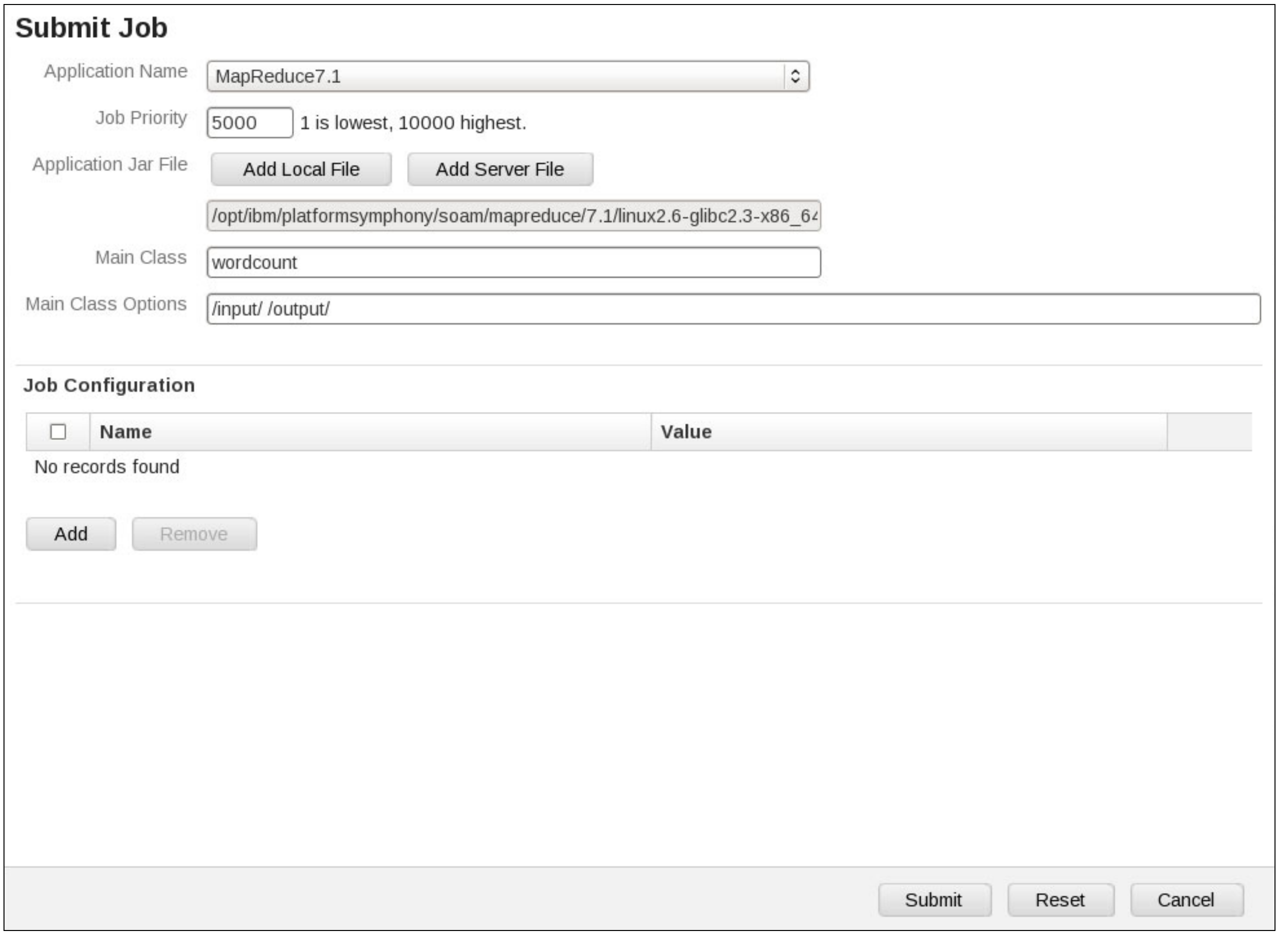


Figure 4-6 MapReduce job selection page

We test it with the Wordcount example file, as shown in Figure 4-7.

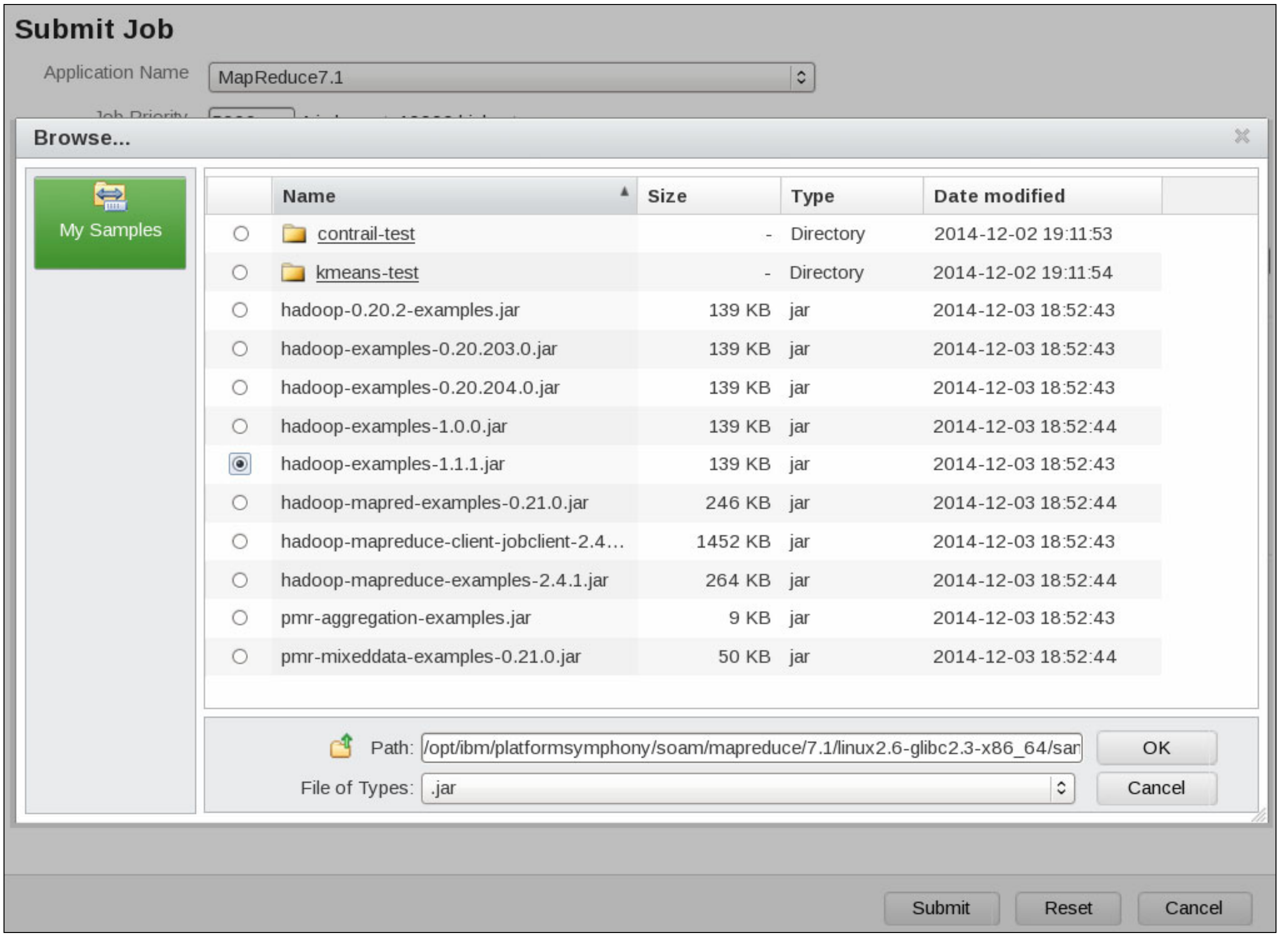


Figure 4-7 Hadoop example files

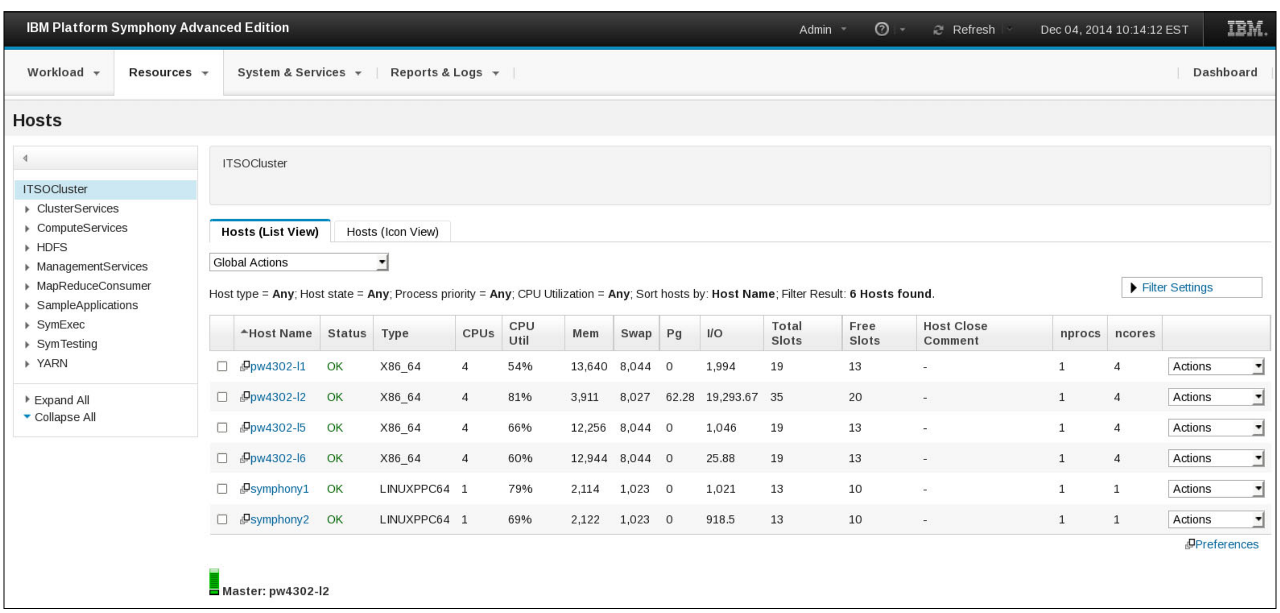
Select **Main Class Options**, and your directories in the Spectrum Scale cluster are as shown in Figure 4-6. Now, you can submit the job.

After it finishes, you see the “Job finished” screen output shown in Figure 4-8 on page 112.

```
Submit Job
Job finished.
mrsh.printAPIVersion.1_1_1
14/12/04 14:40:48 GMT INFO input.FileInputFormat: Total input paths to process : 6
14/12/04 14:40:49 GMT WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java clas
applicable
14/12/04 14:40:49 GMT WARN snappy.LoadSnappy: Snappy native library not loaded
14/12/04 14:40:49 GMT INFO internal.MRJobSubmitter: Connected to JobTracker(SSM)
14/12/04 14:40:49 GMT INFO internal.MRJobSubmitter: Job <word count> submitted, job id <17>
14/12/04 14:40:49 GMT INFO internal.MRJobSubmitter: Job will not verify intermediate data integrity using checksum.
14/12/04 14:40:49 GMT INFO mapred.JobClient: Running job: job_ssm_0017
14/12/04 14:40:50 GMT INFO mapred.JobClient: map 0% reduce 0%
14/12/04 14:40:56 GMT INFO mapred.JobClient: map 16% reduce 0%
14/12/04 14:41:01 GMT INFO mapred.JobClient: map 33% reduce 0%
14/12/04 14:41:02 GMT INFO mapred.JobClient: map 100% reduce 100%
14/12/04 14:41:02 GMT INFO mapred.JobClient: Job complete: job_ssm_0017
14/12/04 14:41:03 GMT INFO mapred.JobClient: Counters: 19
14/12/04 14:41:03 GMT INFO mapred.JobClient: Shuffle Errors
14/12/04 14:41:03 GMT INFO mapred.JobClient: WRONG_PATH=0
14/12/04 14:41:03 GMT INFO mapred.JobClient: CONNECTION=0
14/12/04 14:41:03 GMT INFO mapred.JobClient: IO_ERROR=0
14/12/04 14:41:03 GMT INFO mapred.JobClient: FileSystemCounters
14/12/04 14:41:03 GMT INFO mapred.JobClient: FILE_BYTES_WRITTEN=1911522
14/12/04 14:41:03 GMT INFO mapred.JobClient: Map-Reduce Framework
14/12/04 14:41:03 GMT INFO mapred.JobClient: Reduce input groups=20914
14/12/04 14:41:03 GMT INFO mapred.JobClient: Combine output records=146398
14/12/04 14:41:03 GMT INFO mapred.JobClient: Map output records=1187856
14/12/04 14:41:03 GMT INFO mapred.JobClient: Map input records=173424
14/12/04 14:41:03 GMT INFO mapred.JobClient: SHUFFLED_MAPS=6
14/12/04 14:41:03 GMT INFO mapred.JobClient: Reduce shuffle bytes=1911354
14/12/04 14:41:03 GMT INFO mapred.JobClient: Combine input records=1313340
14/12/04 14:41:03 GMT INFO mapred.JobClient: Spilled Records=250968
14/12/04 14:41:03 GMT INFO mapred.JobClient: SPLIT_RAW_BYTES=228
14/12/04 14:41:03 GMT INFO mapred.JobClient: Map output bytes=12309216
14/12/04 14:41:03 GMT INFO mapred.JobClient: Reduce input records=20914
14/12/04 14:41:03 GMT INFO mapred.JobClient: GC_TIME_MILLIS=536
14/12/04 14:41:03 GMT INFO mapred.JobClient: FAILED_SHUFFLE=0
14/12/04 14:41:03 GMT INFO mapred.JobClient: MERGED_MAP_OUTPUTS=0
14/12/04 14:41:03 GMT INFO mapred.JobClient: Reduce output records=20914
```

Figure 4-8 MapReduce job finished output

During the MapReduce job, you can monitor how the cluster uses the resources, as shown in Figure 4-9 and in Figure 4-10.



Host Name	Status	Type	CPUs	CPU Util	Mem	Swap	Pg	I/O	Total Slots	Free Slots	Host Close Comment	nprocs	ncores	
pw4302-l1	OK	X86_64	4	54%	13,640	8,044	0	1,994	19	13	-	1	4	Actions
pw4302-l2	OK	X86_64	4	81%	3,911	8,027	62.28	19,293.67	35	20	-	1	4	Actions
pw4302-l5	OK	X86_64	4	66%	12,256	8,044	0	1,046	19	13	-	1	4	Actions
pw4302-l6	OK	X86_64	4	60%	12,944	8,044	0	25.88	19	13	-	1	4	Actions
symphony1	OK	LINUXPPC64	1	79%	2,114	1,023	0	1,021	13	10	-	1	1	Actions
symphony2	OK	LINUXPPC64	1	69%	2,122	1,023	0	918.5	13	10	-	1	1	Actions

Figure 4-9 CPU use during a MapReduce job

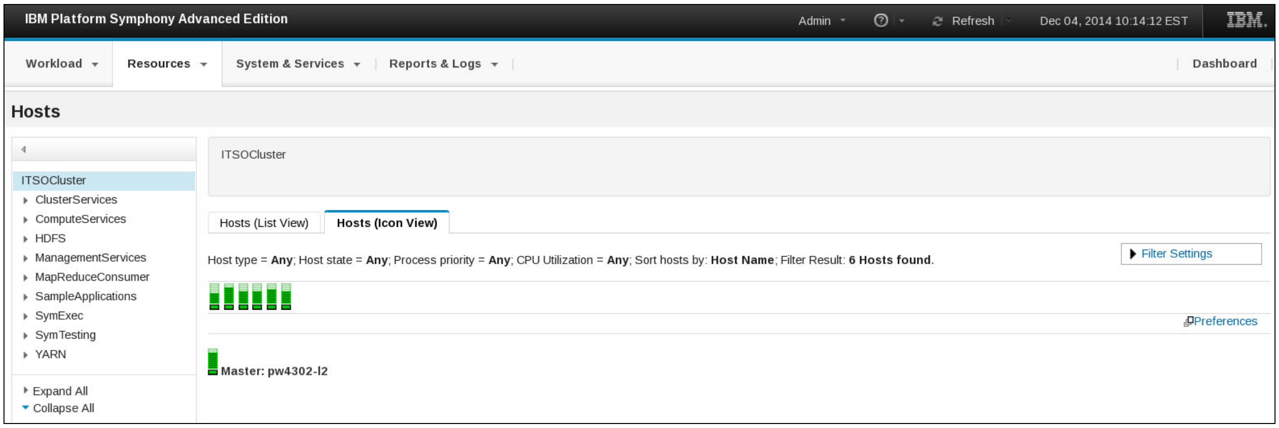


Figure 4-10 Cluster use during a MapReduce job

When the job is complete, the main MapReduce page shows the time elapsed during the MapReduce job, as shown in Figure 4-11.

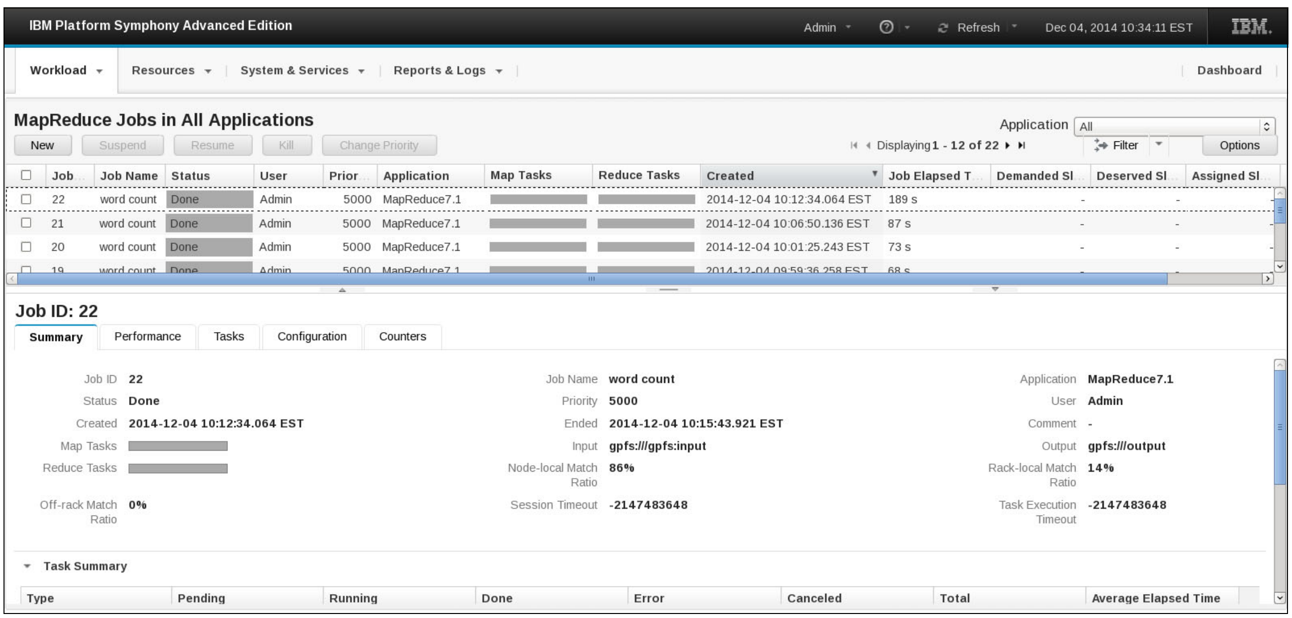


Figure 4-11 Elapsed time until job finish

The results file is created. In our example, it is in the `/gpfs_fpo/output/` directory as `part-r-00000`. In the file, you can see the occurrence of each word, as shown in Figure 4-12.

```
!       36372
!=      11258
"       16188
"'      1166
"'";    866
"*      5196
"*****  866
"**************************************************\n"    3464
"-//Sun 866
"."     1732
"/"     4330
"//W3//Comment//EN">    866
"00"    4330
"02"    2598
"2"     866
"23"    866
"4A"    866
"6"     866
"70"    866
"70,00" 866
"</b>   866
"</b><br>");    1732
"<address>"     2334
"<cluster>"     1167
"<port>"        2334
"="     866
"ALLOCATE       1166
"ALTER  1166
"AS     2033
"Access 1166
"CANCEL"        866
"CEESDLG1"      866
"COMMON"        866
"Daily  1167
"EXECIO 2332
"FREE   2332
"FROM   866
"GMULLER";      866
"GMULLER.SERV"; 866
"GMULLER\0"     1732
"Hourly 2334
```

Figure 4-12 Result file /gpfs_fpo/output/part-r-00000

This chapter has shown how it is possible to run IBM Platform Symphony V7.1 with Hadoop 1.1.1 in a mixed cluster environment for big data.

5

IBM Spectrum Scale for big data environments

IBM Spectrum Scale provides global file-based storage for large scale, high-performance cluster applications and analytics, such as scale-out databases and Hadoop-based analytics.

This chapter presents an overview of the components of Spectrum Scale, how it compares with Hadoop Distributed File System (HDFS), and covers the new features in version 4.1. It includes the following topics:

5.1 Spectrum Scale functions

IBM Spectrum Scale is a proven, scalable, high-performance data and file management solution (based upon IBM General Parallel File System or GPFS technology) which has being used extensively across multiple industries worldwide. Spectrum Scale provides simplified data management and integrated information lifecycle tools capable of managing petabytes of data and billions of files to help the growing cost of managing growing amounts of data.

Spectrum Scale enables virtualization, analytics, file, and object use cases to be unified into a single scale-out data plane for the entire data center. It can provide a single namespace for all of this data for a single point of management. Data can then be tiered in differentiated classes of storage and accessed around the globe. This ensures that data is always available in the right place at the right time. See Figure 5-1.

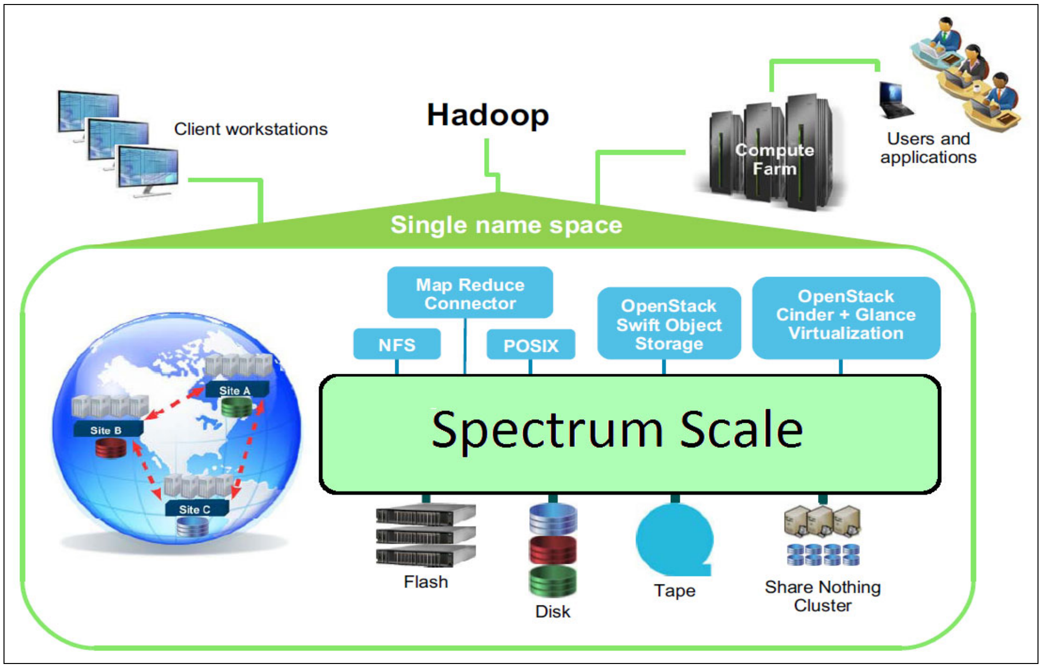


Figure 5-1 Overview of Spectrum Scale capabilities

Spectrum Scale provides world-class storage management delivered in software and used on any hardware, with extreme scalability, flash accelerated performance, and automatic policy-based storage tiering from flash through disk to tape. This can reduce storage costs 90% while improving security and management efficiency in cloud, big data, and analytics environments.

5.2 Spectrum Scale benefits

Spectrum Scale is full-featured, software-defined storage with management tools for advanced storage virtualization, integrated high availability, automated tiered storage, and the performance to effectively manage very large quantities of file data. It provides storage management, information lifecycle management tools, centralized administration and shared access to file systems from remote Spectrum Scale clusters providing a global namespace.

Spectrum Scale provides fast data access and simple, cost-effective data management. Implementing it helps in several ways:

- Enhanced security

  Protect data on disk from security breaches, unauthorized access, or being lost, stolen, or improperly discarded with encryption of data at rest. Enable Health Insurance Portability and Accountability Act (HIPAA), Sarbanes-Oxley, European Union Standards (EU), and various national data privacy compliance laws.
- Extreme performance

  Server-side Spectrum Scale flash caches speed I/O performance up to 6x, which improves application performance, while still providing all of the manageability benefits of shared storage.
- Reduced acquisition costs

  By using standard servers and storage rather than expensive, special purpose hardware.
- Limitless elastic data scaling

  Scale out with relatively inexpensive standard hardware, while maintaining world-class storage management.
- Increased resource and operational efficiency

  Pooling redundant isolated resources optimizes use.
- Greater IT agility

  Quickly react, provision, and redeploy resources in response to new requirements.
- Intelligent resource use and automated management

  Automated, policy-driven management of storage reduces storage costs 90% and improves operational efficiencies.
- Geographically distributed workflows

  Putting critical data close to everyone and the resources that need it helps accelerate schedules and time to market.

Use on some of the world's most powerful supercomputers has proven that Spectrum Scale provides highly reliable and efficient use of infrastructure bandwidth. For example, the Mira supercomputer at Argonne National Laboratory—the fifth-fastest supercomputer in the world—features a 768,000-core IBM Blue Gene cluster with Spectrum Scale that supports scientific research, including complex modeling in the fields of material science, climatology, seismology, and computational chemistry.

Spectrum Scale enables a group of computers concurrent access to share a common set of file data over a storage area network (SAN) infrastructure, a TCP/IP network, or a combination of connection types. The computers can run a mixture of IBM AIX, Linux, or Microsoft Windows operating systems.

A Spectrum Scale cluster can be a single node in a tiered storage solution, two nodes providing a high-availability platform supporting a database application, or thousands of nodes used for applications such as weather-pattern modeling. The largest existing configurations, such as Mira, can exceed hundreds of thousands of cores.

5.3 Comparison of HDFS and Spectrum Scale features

As explained in Chapter 1, “Introduction to big data” on page 1, Spectrum Scale is suitable as a substitute for HDFS as the file system for big data workloads. Spectrum Scale includes several enterprise features that provide distinct advantages, such as the capability to make a logical, read-only copy of the file system at any point in time.

You can use this snapshot feature to back up your system and recover from errors also. The snapshot file does not occupy disk space until it is modified or deleted, so it provides an efficient backup and recovery method.

Spectrum Scale includes Active File Management (AFM) support to enable asynchronous access and control of local and remote files. You can use this capability to create associations from a local Spectrum Scale cluster to a remote cluster or storage and to define the location and flow of file data to automate data management. The AFM implementation uses the inherent scalability of Spectrum Scale to provide a multi-node, consistent cache of data in a home cluster.

Table 5-1 provides a comparison of Spectrum Scale and the HDFS features.

Table 5-1 IBM Spectrum Scale versus HDFS features

Features	Spectrum Scale	HDFS
Hierarchical storage management	Allows sufficient use of disk drives with different performance characteristics	
High-performance support for MapReduce applications	Stripes data across disks by using metablocks, which allows a MapReduce split to be spread over local disks	Places a MapReduce split on one local disk
High performance support for traditional applications	► Manages metadata by using the local node when possible, rather than reading metadata into memory unnecessarily ► Caches data on the client side to increase throughput of random reads ► Supports concurrent reads and writes by multiple programs ► Provides sequential access that enables fast sorts, improving performance for query languages, such as Pig and Jaql	

Features	Spectrum Scale	HDFS
High availability	Has no single point of failure because the architecture supports the following attributes: ► Distributed metadata ► Replication of both metadata and data ► Node quorums ► Automatic distributed node failure recovery and reassignment	High availability failover would cause service interruption on the namenode
POSIX compliance	Fully POSIX-compliant, which provides the following benefits: ► Support for a wide range of traditional applications ► Support for UNIX utilities that enable file copying by using FTP or SCP ► Updating and deleting data ► No limitations or performance issues when using a Lucene text index	Not POSIX-compliant, which creates the following limitations: ► Limited support of traditional applications ► No support of UNIX utilities, which requires using the `Hadoop DFS get` command or the put command to copy data ► After the initial load, data is read-only ► Lucene text indexes must be built on local file system or NFS because updating, inserting, or deleting entries is not supported
Ease of deployment	Supports a single cluster for analytics and databases	Requires separate clusters for analytics and databases
Other enterprise level file system features	► Workload isolation ► Security	
Data replication	Provides cluster-to-cluster replication over a wide area network	

5.4 What's new in Spectrum Scale 4.1

The subsections that follow describe the new features in IBM Spectrum Scale 4.1.

5.4.1 File encryption and secure erase

Spectrum Scale version 4.1 introduces file-level encryption for the protection of data at rest and secure data deletion. File encryption can help protect data from security breaches or unauthorized access and from being stolen or improperly discarded. Encryption is controlled by policy, and data is not decrypted until it reaches the application node. Each application node can have the same encryption key or a different key, which allows for secure scale-out processing or multitenancy on shared storage hardware.

Without encryption, when a file is deleted, the space is marked as free, but the data is not overwritten until a new file occupies the same physical storage location. Secure erase provides a fast, simple way to delete a file and also make the free space unreadable and, therefore, securely erased. Spectrum Scale 4.1 complies with publications NIST SP 800-131A, "Recommended Security Controls for Federal information Systems and Organizations," and FIPS 140-2, "Security Requirements for Cryptographic Modules."

5.4.2 Transparent flash cache

Many applications can benefit from a large amount of local file cache capacity. A local data cache can give applications low-latency access to file data and reduce the load on shared network and back-end storage. Using system memory for file cache is expensive and has limited capacity, compared to persistent data storage devices. Solid-state disks provide an economical way to expand the capacity of local data cache, but using them for file caching requires intelligent software.

Spectrum Scale flash cache addresses the need for an expanded local file cache by taking advantage of solid-state disks placed directly in client nodes. The solid-state disks are seamlessly integrated as an extension of the Spectrum Scale file cache. Flash cache transparently acts as an extension of the Spectrum Scale file cache, called the *pagepool*. In this way, Spectrum Scale flash cache can cost-effectively accelerate applications and deliver more than a terabyte of local data cache capacity.

5.4.3 Network performance monitoring

Because software-defined storage relies on the underlying server, network, and storage hardware, it is critical to keep this infrastructure running at peak performance. To help monitor network performance, Spectrum Scale 4.1 provides Remote Procedure Call (RPC) latency statistics to help detect and troubleshoot networking issues that might affect operation.

5.4.4 AFM enhancements

Version 4.1 introduces several features that optimize AFM operation and make it easier to move data. Although data transfers into an AFM cache are triggered on demand, it is possible for AFM to prefetch (prepopulate) data, so the necessary file data is waiting in the cache before it is needed.

Spectrum Scale 4.1 speeds data prefetching, which makes it easier to prepopulate more files in a shorter period of time. In this way, prefetch improves the performance of large file transfers.

To move more data quickly between the cache and home, Spectrum Scale 4.1 also adds parallel data movement within a single file set. Now, you can move data in parallel between cache and home for a single large file or many smaller files.

To further improve data transfer speed across high-performance networks, AFM now supports the use of the NSD protocol in addition to NFS. This makes it easier to use AFM to use flash devices to accelerate applications and move data between Spectrum Scale clusters.

5.4.5 NFS data migration

With Spectrum Scale 4.1, you can use AFM to migrate data from one cluster to another when upgrading hardware or buying a new system. Because AFM uses the NFS protocol, any NFS data source might serve as home, and it does not have to be Spectrum Scale.

Spectrum Scale data migration can minimize downtime for applications, move file data along with permissions associated with files, and consolidate data from multiple legacy systems into a single, more powerful system.

5.4.6 Backup and restore improvements

File set snapshots can now be restored to the active file system. The tool restores the active file set data and attributes to the point in time when the snapshot was taken.

The **`mmbackup`** utility has been enhanced to automatically adjust its own work distribution and parallel access to IBM Spectrum Protect (formerly IBM Tivoli® Storage Manager), based on resource availability and user-provided input parameters.

5.4.7 FPO enhancements

File Placement Optimizer (FPO) enhancements include improved data recovery with the addition of advanced data locality and aware file system recovery. To better support MapReduce and other big data workloads, Spectrum Scale 4.1 also boosts the performance of concurrent directory changes.

5.5 For more information

For learn more about Spectrum Scale, see this IBM Redbooks publication:

IBM Spectrum Scale (formerly GPFS), SG24-8254

`http://www.redbooks.ibm.com/redpieces/abstracts/sg248254.html`

6

IBM Application Service Controller in a mixed environment

This chapter introduces IBM Application Service Controller, a technology to control long-running applications in an IBM Platform Symphony.

The following sections are included in this chapter:

6.1 Enabling the Application Service Controller

To enable the Application Service Controller, you need the IBM Platform Symphony Advanced Edition version 7.1 already installed. Follow the instructions in 4.3, "IBM Platform Symphony installation steps" on page 72.

6.1.1 Steps to enable the Application Service Controller

First, log in as the Platform Symphony cluster administrator (in this example, `egoadmin`), source the profile, and log in to EGO as shown in Example 6-1.

Example 6-1 Log in as egoadmin

```
[root@pw4302-l2 symphony71]# su - egoadmin
[egoadmin@pw4302-l2 ~]$ . /opt/ibm/platformsymphony/profile.platform
[egoadmin@pw4302-l2 ~]$ egosh user logon -u Admin -x Admin
Logged on successfully
[egoadmin@pw4302-l2 ~]$
```

Then, disable all service-oriented application modernization (SOAM) applications, stop EGO services, and shut down EGO daemons on all nodes, as shown in Example 6-2.

Example 6-2 Disabling services

```
[egoadmin@pw4302-l2 ~]$ soamcontrol app disable all
Are you sure that you want to disable all applications and terminate workload?
(Y/N) Y
All applications disabled.
[egoadmin@pw4302-l2 ~]$
[egoadmin@pw4302-l2 ~]$ egosh service stop all
<all> will be considered as the option to stop all the services,
and all the existing services will be stopped
Service <plc> has been stopped successfully
Service <WEBGUI> has been stopped successfully
Service <GPFSmonitor> has been stopped successfully
Service <derbydb> has been stopped successfully
Service <purger> has been stopped successfully
Service <SMCP> has been stopped successfully
Service <NameNode> has been stopped successfully
Service <USSD> has been stopped successfully
Service <RSA> has been stopped successfully
Service <SD> has been stopped successfully
Service <WebServiceGateway> has been stopped successfully
Service <MRSS> has been stopped successfully
Service <SymphonyYARN> has been stopped successfully
Service <ServiceDirector> has been stopped successfully
Service <SecondaryNode> has been stopped successfully
Service <DataNode> has been stopped successfully
Service <RS> has been stopped successfully
[egoadmin@pw4302-l2 ~]$ egosh ego shutdown all
Do you really want to shut down LIMs on all hosts? [y/n] y
Shut down LIM on <pw4302-l1> ...... done
Shut down LIM on <pw4302-l5> ...... done
Shut down LIM on <pw4302-l6> ...... done
Shut down LIM on <symphony2> ...... done
```

```
Shut down LIM on <symphony1> ...... done
Shut down LIM on <pw4302-l2> ...... done
[egoadmin@pw4302-l2 ~]$
```

Apply the Application Service Controller entitlement by using the **egoconfig setentitlement** command, as shown in Example 6-3.

Example 6-3 Apply the Application Service Controller entitlement

```
[egoadmin@pw4302-l2 symphony71]$ egoconfig setentitlement
platform_asc_sym_entitlement.dat
To enable the new entitlement, restart the cluster.
Successfully set entitlement.
[egoadmin@pw4302-l2 symphony71]$
```

Now issue the command **egosh ego start all** to restart the cluster services on all hosts, as shown in Example 6-4.

Example 6-4 Starting the cluster on all

```
[egoadmin@pw4302-l2 ~]$ egosh ego start all
Do you really want to start up LIM on all hosts ? [y/n]y
Start up LIM on <pw4302-l2> ...... done
Start up LIM on <pw4302-l1> ...... done
Start up LIM on <pw4302-l5> ...... done
Start up LIM on <pw4302-l6> ...... done
Start up LIM on <symphony2> ...... done
Start up LIM on <symphony1> ...... done
[egoadmin@pw4302-l2 ~]$
```

Note: By default, to start EGO on all hosts, EGO uses rsh to connect to the hosts in the cluster. The connectivity does not work well without first configuring rsh and its authentication. However, IBM Redbooks residency team that wrote this book recommends that you use SSH keys exchange for the egoadmin user and add the following line to the `$EGO_CONFDIR/ego.conf` file:

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

6.1.2 Verifying the Application Service Controller in the GUI

Log in to the Platform Symphony graphical user interface (GUI) shown in Figure 6-1 on page 126 by using the following address:

`http://<IP Address or hostname>:8080/platform/`

Figure 6-1 Symphony login window

Click **Workload**, and verify that the Application Service Controller item appears in the GUI, as shown in Figure 6-2.

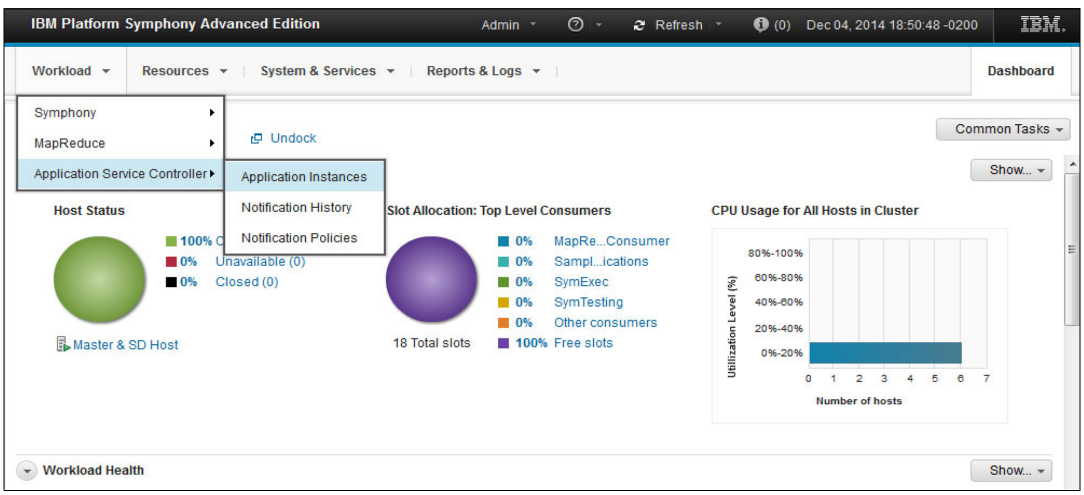


Figure 6-2 Verify that the Application Service Controller has been enabled

6.2 Sample application instance templates

To create an application instance template, look at some of the existing templates in the $EGO_TOP/asc/conf/ directory where there are two YAML files, as shown in Example 6-5.

Example 6-5 Simple application Instance templates

```
[root@pw4302-12 platformsymphony]# ls $EGO_TOP/asc/conf/*.yaml
/opt/ibm/platformsymphony/asc/conf/asc_sample_minimal.yaml
/opt/ibm/platformsymphony/asc/conf/asc_sample.yaml
[root@pw4302-12 platformsymphony]#
```

In the first file (Example 6-6), there is an example on how a daemon could be available and controlled by the Application Service Controller.

Example 6-6 Sample application instance template

```
[root@pw4302-l2 platformsymphony]# cat
/opt/ibm/platformsymphony/asc/conf/asc_sample_minimal.yaml
asc_template_version: 2014-09-20

resources:
    service_activity:
        type: IBM::ASC::Activity
        properties:
            commands:
                start:
                    command: sleep 5000

    service:
        type: IBM::ASC::Service
        properties:
            consumer:
                name: MyConsumer
                executionuser: root
            resourcegroup: ComputeHosts
            activity:
                - { get_resource: service_activity }
[root@pw4302-l2 platformsymphony]#
```

The second, more detailed file, shown in Example 6-5 on page 126, provides a sample of how a high available service is created using the Application Service Controller, with a virtual IP and two services floating through all ComputeHosts.

> **Note:** In Example 6-6, we did not paste the entire contents of the file, because it has too many lines. But by analyzing the configuration file, you can see the complexity of the high-available resource groups, with many services and many nodes, which can be handled by the Application Service Controller.

Another template sample is the one for big data applications shown in Example 6-7, which is available in the $EGO_TOP/asc/conf/samples/ directory.

Example 6-7 Sample application instance template for big data applications

```
[root@pw4302-l2 logs]# ls $EGO_TOP/asc/conf/samples/
ambari asc_sample.yaml  clouderamanager  egodocker.py  hadoopDocker  sparkHDFS
asc_sample_minimal.yaml  cassandra      deployment.xml   hadoop      mongodb
zookeeper
[root@pw4302-l2 logs]#
```

In the $EGO_TOP/asc/conf/samples/ directory, you can see a list of supported applications (since the last GA) as follows:

- Ambari
- Cassandra
- Cloudera manager
- Hadoop 2.4.1
- Hadoop 2.4.1 with Docker support
- MongoDB
- Spark and HDFS with Docker support
- ZooKeeper

The list of applications' templates available increases after each GA.

> **Note:** You can download more templates from the IBM Platform Application Service Controller page on the IBM Bluemix DevOps Services web page:
>
> `https://hub.jazz.net/user/ibmasc`

6.3 Registering an application template for Cassandra

This section shows how to register an application template with the Application Service Controller. The intention is not to build a full working database, but to demonstrate, step by step, how to make a template ready by using the controller.

From the IBM Platform Symphony Advance Edition GUI, click **Workload** → **Application Service Controller** → **Application Instances** to open the initial applications window shown in Figure 6-3. Because this is the first application registered in the cluster, the ‘‘No application instances found” message is displayed.

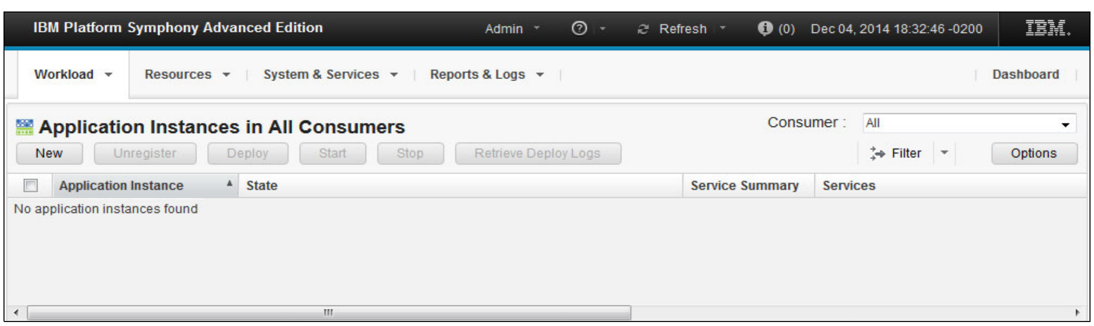


Figure 6-3 Registered application instances

6.3.1 Preparing the application package

Before you can register an application instance, you must create the required packages that contain all of the third-party binaries, Application Service Controller scripts, the `deployment.xml` file, and any extra files for the services. The application template defines the packages that need to be created before you can register application instances. Packaging an Application Service Controller service for deployment involves adding all service files and any dependent files that are associated with the service in a package.

All packages have dependencies. Cassandra needs Java installed (it is already installed in the cluster, as described in Chapter 4, “Mixed IBM Power Systems and Intel environment for big data” on page 69).

Note: Verify that all dependencies are either preinstalled or deployed with the service. For example, if your application is Microsoft .NET dependant, ensure that the .NET framework is installed and that it is the correct version for your application.

The package can be up to 2 GB in size, maximum.

You can use any of the following supported formats for the package:

- `.zip`
- `.tar`
- `.taz`
- `.tar.zip`
- `.tar.Z`
- `.tar.gz`
- `.tgz`
- `.jar`
- `.gz`

Note: If you use a utility other than `.gzip`, ensure that the compression and uncompression utility is in the path environment variable when deploying the application.

Start building the package for Cassandra, as shown in Example 6-8, by following these steps:

1. Copy the deployment.xml that is provided in the samples root path to the folder **`$EGO_CONFDIR/../../asc/conf/samples/deployment.xml`**
2. Create an `ascscripts` subfolder and copy all of the scripts to that folder. Ensure that all `.sh` files have execution permission. In this example, the scripts are ready in the $EGO_TOP/asc/conf/samples/cassandra/Cassandra_2.1_RHEL65_scripts/ folder.
3. Create a package subfolder.

Example 6-8 Create the package structure

```
[root@pw4302-l2 samples]# cd /tmp
[root@pw4302-l2 tmp]# mkdir cassandrapkg
[root@pw4302-l2 tmp]# cd cassandrapkg/
[root@pw4302-l2 cassandrapkg]# cp $EGO_TOP/asc/conf/samples/deployment.xml .
[root@pw4302-l2 cassandrapkg]# mkdir ascscripts
[root@pw4302-l2 cassandrapkg]# cp
$EGO_TOP/asc/conf/samples/cassandra/Cassandra_2.1_RHEL65_scripts/* ./ascscripts/
[root@pw4302-l2 cassandrapkg]# chmod +x ./ascscripts/*
[root@pw4302-l2 cassandrapkg]# ls -l ./ascscripts/*
-rwxr-xr-x. 1 root root 172 May 16 10:33 ./ascscripts/checkdecomm.sh
-rwxr-xr-x. 1 root root 150 May 16 10:33 ./ascscripts/common.inc
-rwxr-xr-x. 1 root root 201 May 16 10:33 ./ascscripts/deploy.sh
-rwxr-xr-x. 1 root root 391 May 16 10:33 ./ascscripts/prestartcass.sh
-rwxr-xr-x. 1 root root  67 May 16 10:33 ./ascscripts/startcass.sh
-rwxr-xr-x. 1 root root  74 May 16 10:33 ./ascscripts/startdecomm.sh
-rwxr-xr-x. 1 root root  59 May 16 10:33 ./ascscripts/stopcass.sh
-rwxr-xr-x. 1 root root 201 May 16 10:33 ./ascscripts/undeploy.sh
[root@pw4302-l2 cassandrapkg]# mkdir package
[root@pw4302-l2 cassandrapkg]# ls
ascscripts  deployment.xml  package
```

Note: The deployment.xml file ($EGO_TOP/asc/conf/samples/deployment.xml) is a sample file that must be customized for each package.

To test the deploying of Cassandra with the sample scripts, no changes need to be made in the deployment.xml file.

Collect all of the service binaries, scripts, and extra files that are required for the service to work. For this environment, because Cassandra is used, you need to download the binary files from the Cassandra downloads page of the Apache website:

http://bit.ly/1HZU4d6

That page suggests using a download mirror location. Download the binaries to the packages subfolder, as shown in Example 6-9.

Example 6-9   Download the Cassandra packages

```
[root@pw4302-l2 cassandrapkg]# cd package/
[root@pw4302-l2 package]# wget
http://www.us.apache.org/dist/cassandra/2.1.5/apache-cassandra-2.1.5-bin.tar.gz
--2015-05-16 10:43:13--
http://www.us.apache.org/dist/cassandra/2.1.5/apache-cassandra-2.1.5-bin.tar.gz
Resolving www.us.apache.org... 104.130.219.184,
2001:4801:7828:101:be76:4eff:fe11:7b9d
Connecting to www.us.apache.org|104.130.219.184|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24282772 (23M) [application/x-gzip]
Saving to: âapache-cassandra-2.1.5-bin.tar.gzâ

100%[=========================================================>] 24,282,772
1.25M/s   in 19s

2015-05-16 10:43:33 (1.21 MB/s) - âapache-cassandra-2.1.5-bin.tar.gzâ
```

Note: The sample scripts are configured to work with Cassandra v2.1.0, which is not available anymore, Therefore, you must change the scripts to work with the new version.

We looked into the scripts to find the references to the earlier version and correct them, as shown in Example 6-10.

Example 6-10   Change the scripts to work with Cassandra v2.1.5

```
[root@pw4302-l2 ascscripts]# grep 2.1.0 *
common.inc:export INSTALL_PATH=$DEPLOY_HOME/apache-cassandra-2.1.0
deploy.sh:tar xzvf ./package/apache-cassandra-2.1.0-bin.tar.gz
[root@pw4302-l2 ascscripts]# vi common.inc
[root@pw4302-l2 ascscripts]# vi deploy.sh
[root@pw4302-l2 ascscripts]# grep 2.1.0 *
[root@pw4302-l2 ascscripts]# grep 2.1.5 *
common.inc:export INSTALL_PATH=$DEPLOY_HOME/apache-cassandra-2.1.5
deploy.sh:tar xzvf ./package/apache-cassandra-2.1.5-bin.tar.gz
[root@pw4302-l2 ascscripts]#
```

Note: To use the application template sample, the Application Service Controller must be started as root user. Changes in the scripts are required if another user is going to start the controller.

We installed the cluster as `egoadmin`, so we need change the install path for the package in common.inc, as shown in Example 6-11. The default value in /var does not provide access to `egoadmin`.

Example 6-11 Change DEPLOY_HOME in ascscripts/common.inc

```
#!/bin/bash

export DEPLOY_HOME=/home/egoadmin/asc/auto/cassandra

export INSTALL_PATH=$DEPLOY_HOME/apache-cassandra-2.1.5

export PATH=$PATH:/usr/sbin/:/sbin/
```

Now, create the Cassandra package as Example 6-12 shows.

Example 6-12 Create the Cassandra package

```
[root@pw4302-l2 cassandrapkg]# tar czvf apache-cassandra-2.1.5-pkg.tar.gz
deployment.xml ascscripts package
deployment.xml
ascscripts/
ascscripts/stopcass.sh
ascscripts/prestartcass.sh
ascscripts/startdecomm.sh
ascscripts/undeploy.sh
ascscripts/deploy.sh
ascscripts/checkdecomm.sh
ascscripts/startcass.sh
ascscripts/common.inc
package/
package/apache-cassandra-2.1.5-bin.tar.gz
[root@pw4302-l2 cassandrapkg]#
```

6.3.2 Registering the application instance

To create a new instance, simply click the **New** button, which opens a window to select the YAML template to be loaded, as shown in Figure 6-4.

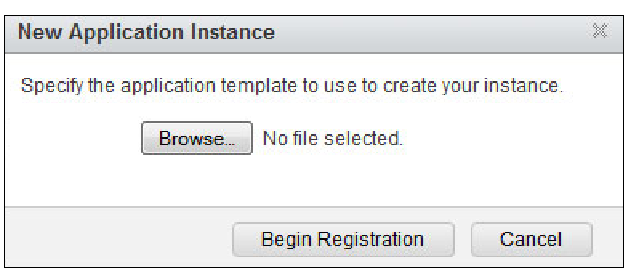


Figure 6-4 New application window

Click **Browse** and select the correct YAML to be loaded. For this example, we test the Cassandra 2.1 template shown in Figure 6-5. To do this, download the Cassandra YAML file from the Platform Symphony management controller server. The file is located at `$EGO_TOP/asc/conf/samples/cassandra/Cassandra_2.1_RHEL65.yaml`. Download the package that is generated in Example 6-12 on page 131.

> **Note:** Remember that we did not install the cluster as `root`, so we changed the scripts to install the cluster in an `egoadmin` location. Therefore, before we use the YAML to create the instance, we change all occurrences of these paths in the `Cassandra_2.1_RHEL65.yaml` file:
>
> `/var/asc/auto/cassandra to /home/egoadmin/asc/auto/cassandra/log`
> `/var/log/asc to /home/egoadmin/asc/auto/cassandra/log`

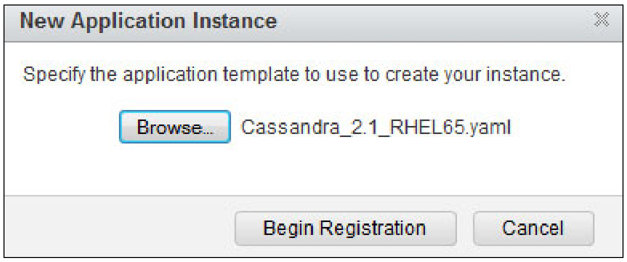


Figure 6-5 Cassandra YAML template loaded

Then, specify a name that will appear in Application Service Controller. For this example, we enter `Cassandra`, as shown in Figure 6-6.

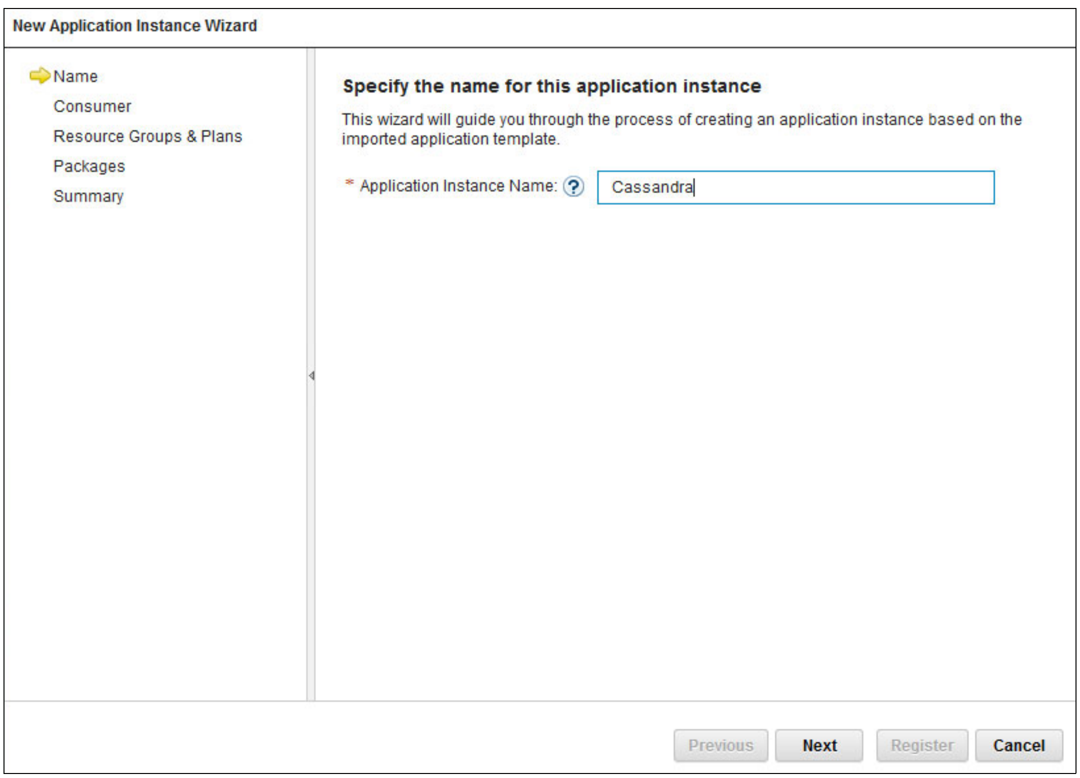


Figure 6-6 Specify the application name

Next, specify a top-level consumer for the application. For this application, we create a new consumer named `Cassandra`, as shown in Figure 6-7.

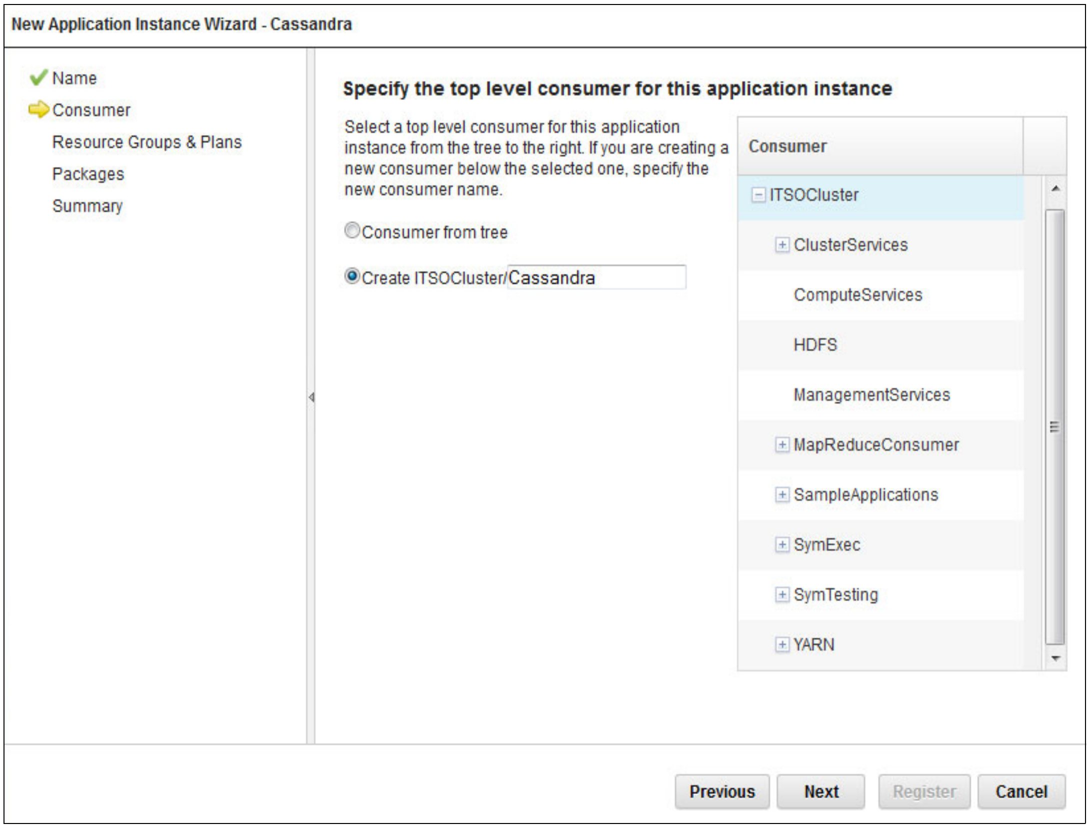


Figure 6-7 Selecting the consumer for the new application to be registered

Then, select the resource groups that will be used for Cassandra's seed nodes and for regular compute nodes. For this example, we select the ManagementHosts Resource group to be used for seed nodes and ComputeHosts for regular nodes (see Figure 6-8).

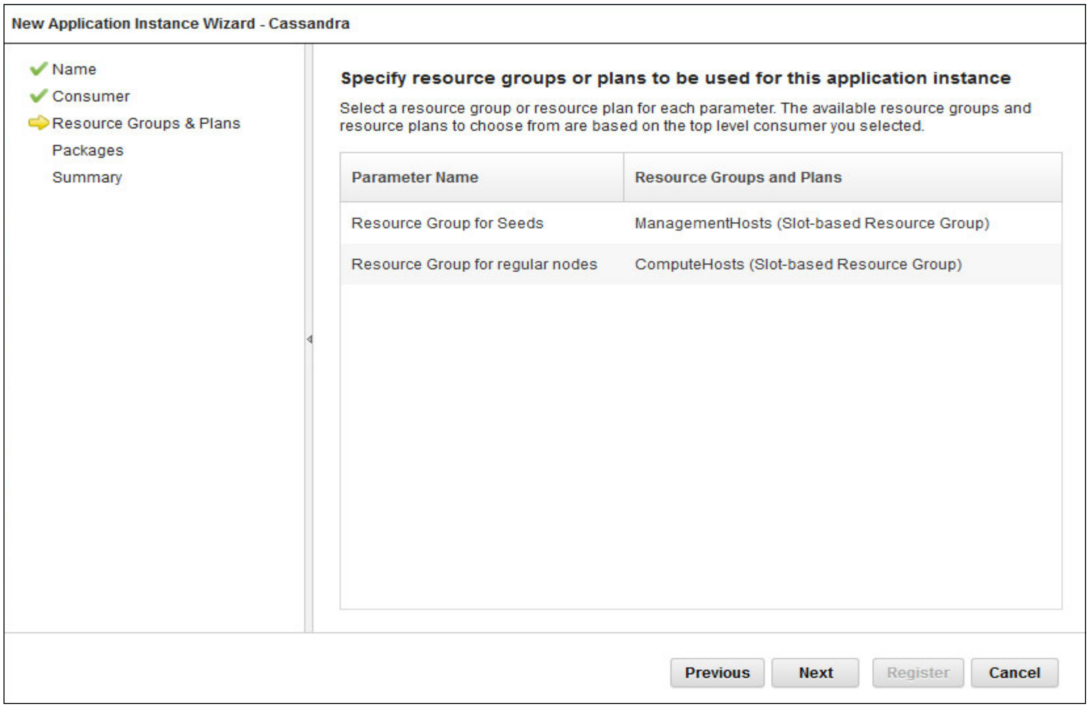


Figure 6-8 Choose the correct Resource Groups

Now, select the package generated in Example 6-12 on page 131, and download the package to your workstation, as shown in Figure 6-9.

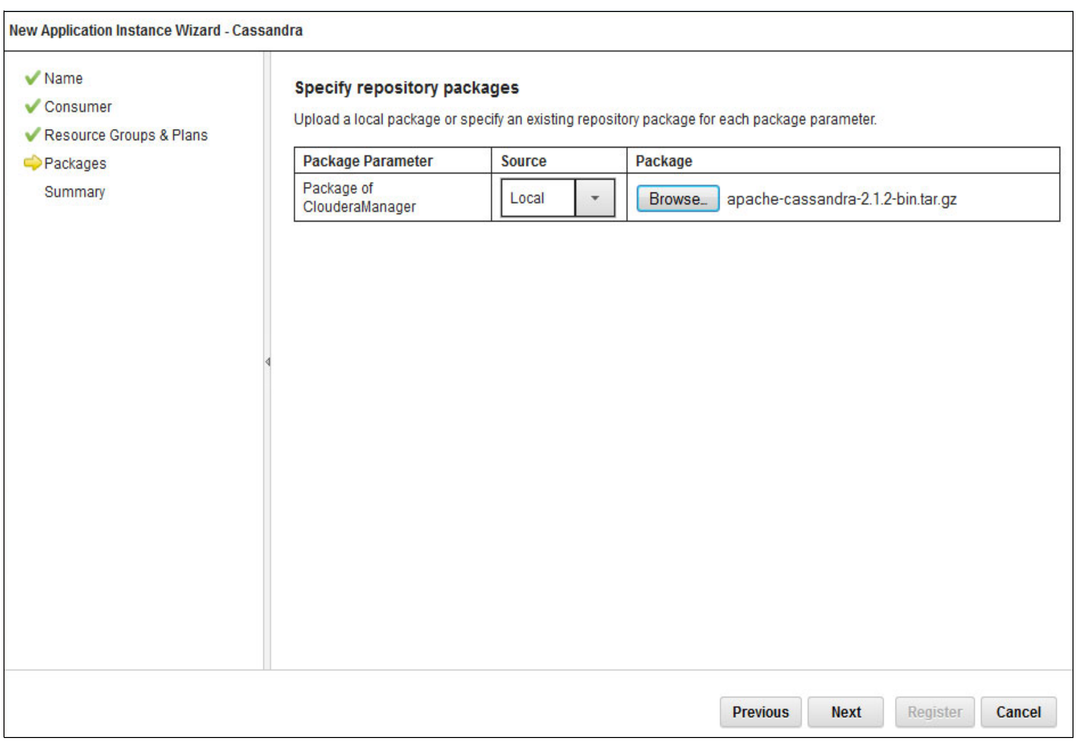


Figure 6-9 Select the package

Click the **Register** button in the Summary window shown in Figure 6-10.

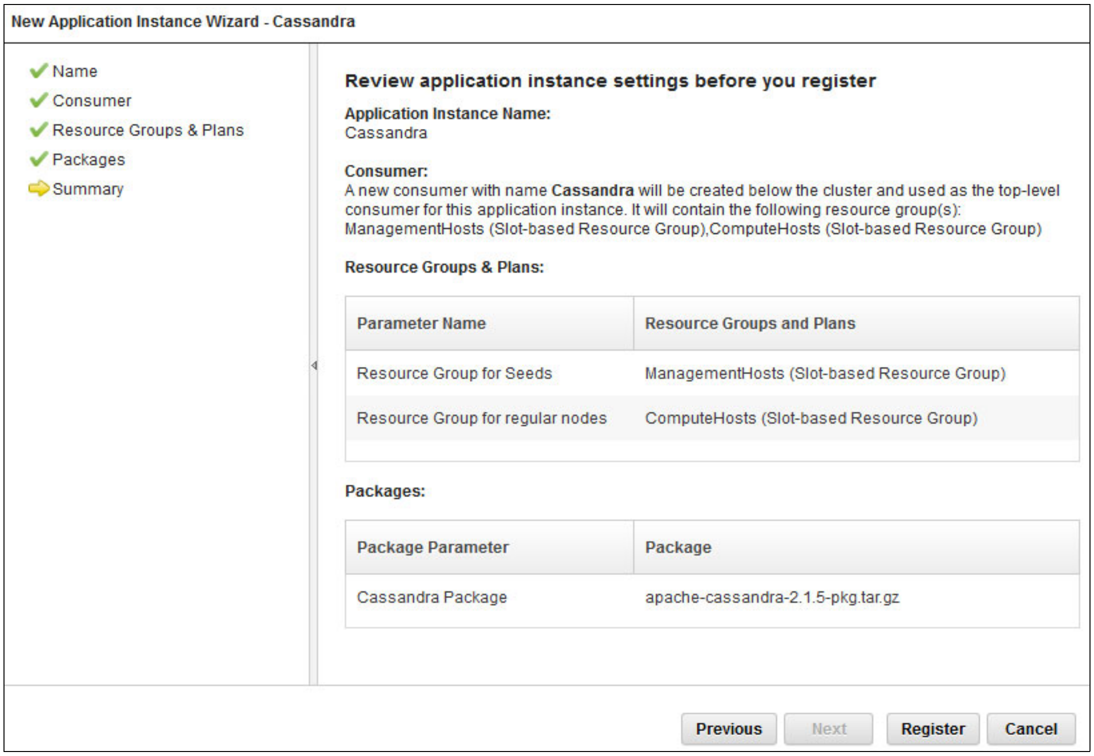


Figure 6-10 Register the application

It takes a few minutes to transfer the package and register the application in the grid, so be patient. After the package is registered, you see the completion message shown in Figure 6-11.

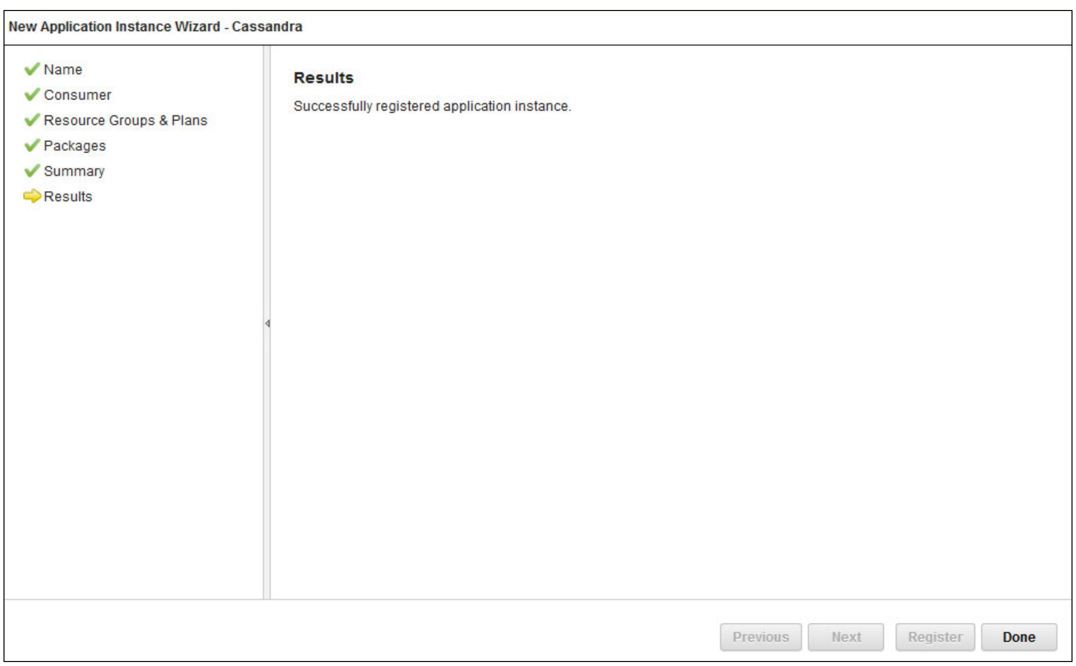


Figure 6-11 Registration completed

The application will appear as registered at the Application Instances panel shown in Figure 6-12.

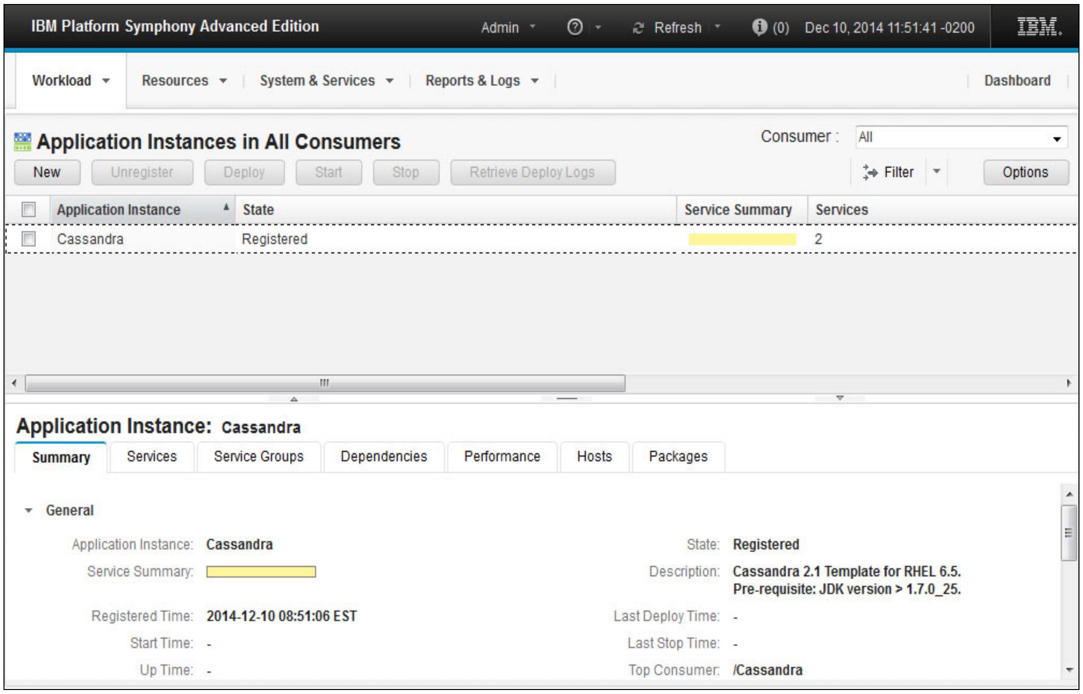


Figure 6-12 Cassandra application registered in the Application instance panel

Now, select the **Cassandra** application and click **Deploy**, as shown in Figure 6-13.

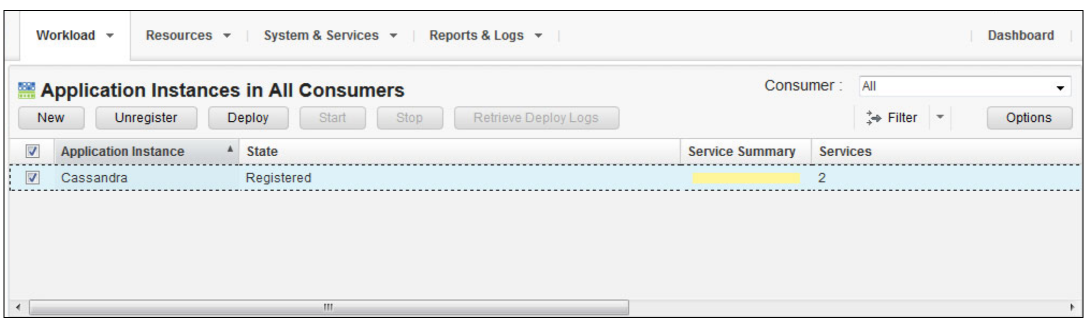


Figure 6-13 Select Cassandra and click Deploy

A dialog window opens to deploy the application, as shown in Figure 6-14. There is no need to include a timeout, because that is optional. In our cluster, we simply click **Deploy**.

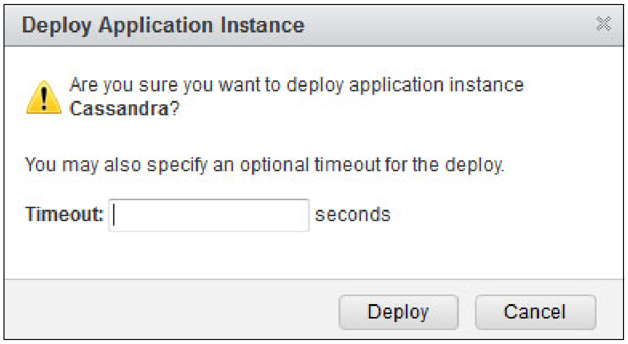


Figure 6-14 Deploy the application instance

In the Application Instance panel shown in Figure 6-15, notice that the Application Service Controller is deploying Cassandra to the grid.

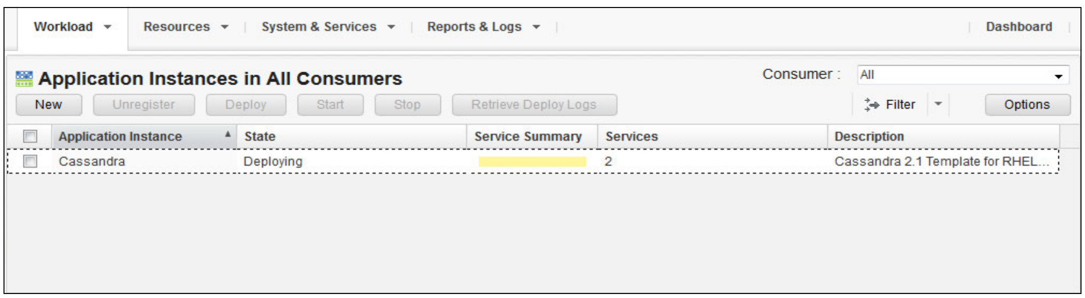


Figure 6-15 Application Service Controller deploying Cassandra

After a few minutes, the Cassandra application is ready to be used in the cluster, as shown in Figure 6-16.

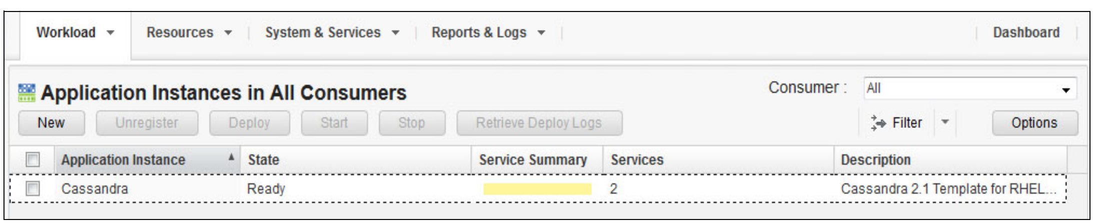


Figure 6-16 Cassandra application ready to be used in the cluster

Select the Cassandra application, and click the Start button, as shown in Figure 6-17.

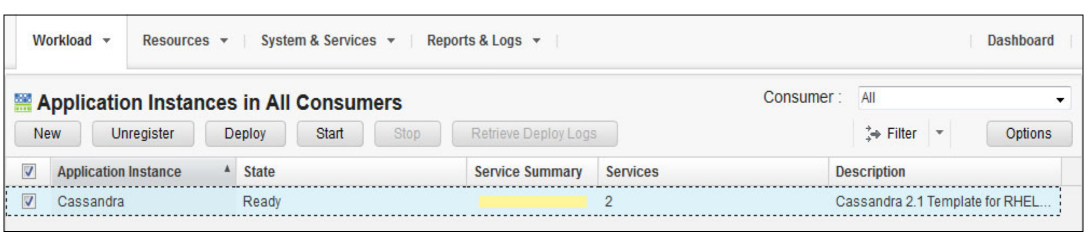


Figure 6-17 Select Cassandra and click Start

Confirm that the application can be started by clicking the Start button again, as shown in Figure 6-18.

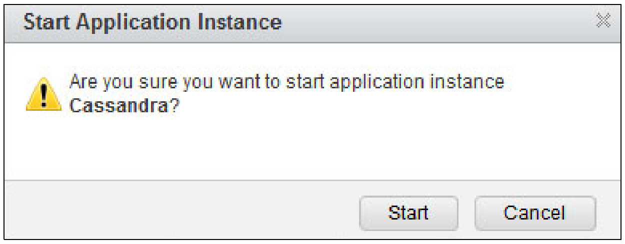


Figure 6-18 Start Application Instance dialog window

The Application Service Controller processes the start procedure as stated at the YAML file (see Figure 6-19).

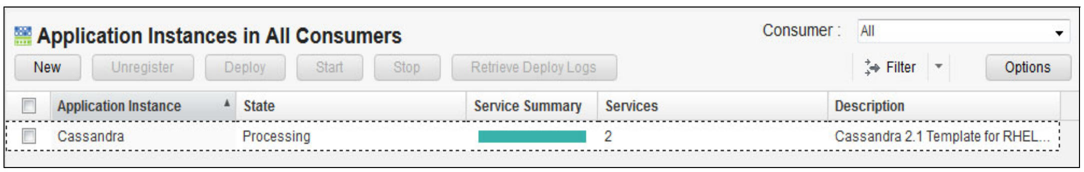


Figure 6-19 Start procedure being processed by Application Service Controller

As Figure 6-20 shows, Cassandra is deployed and started in the grid.

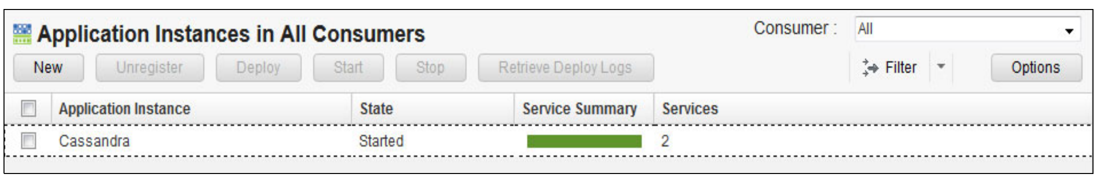


Figure 6-20 Cassandra application started

Now, verify that Cassandra is correctly running in your system, as shown in Example 6-13.

Example 6-13 Check Cassandra status

```
[root@pw4302-12 ~]#
/home/egoadmin/asc/auto/cassandra/apache-cassandra-2.1.5/bin/nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address      Load      Tokens  Owns (effective)  Host ID
Rack
UN  192.168.56.4  65.72 KB   256     100.0%
52e55759-5f66-4056-932d-d64ddc649547  rack1
UN  192.168.56.2  133.5 KB   256     100.0%
ff6fced0-a311-4f29-a4ba-532b3419fe1f  rack1

[root@pw4302-12 ~]#
```

7

IBM Platform Computing cloud services

This chapter describes the IBM Platform Computing cloud services. It includes a scenario of how IBM Platform LSF multicluster and IBM Spectrum Scale Active File Manager can help you efficiently manage the use of the cloud services.

This chapter provides factual and comparative costs of deploying a solution on-premises versus in the cloud. It also includes information about the benefits of on-premises versus in cloud solution deployments.

The following topics are covered:

7.1 IBM Platform Computing cloud services

Engineering, scientific, financial, or research workloads are not the only demanding workloads for technical and high-performance computing (HPC) infrastructures. Big data problems are solved by using the same method: Distributing the workload across multiple machines within a technical computing cluster.

Meeting all of these demands can be especially challenging for organizations that have seasonal or unpredictable demand spikes, need access to additional compute or storage resources to support a growing business, or are starting to use these technologies. The time that it takes to respond to a critical market analysis, a product release, or a research study can be affected by resource availability, which affects competitiveness and profitability.

Organizations can quickly and efficiently overcome these challenges by combining workload management from IBM Platform Computing with the efficiency and cost benefits of cloud computing.

IBM Platform Computing Cloud Service running on the IBM SoftLayer® cloud delivers a versatile, high-performing cloud-based environment to fuel your organization's growth, when you are engaged in the following activities:

- Seeking ways to meet variable workload demands
- In need of clustered resources but do not have the budget or in-house skills to deploy and use a technical computing infrastructure
- Running out of data center space and need to continue to increase compute and storage capacity
- Considering providing applications on a pay-per-use basis but do not have the infrastructure or time to create a service

If any of these activities are important to you, you can count on the IBM Cloud Service benefits to meet your needs:

- Ready-to-use IBM Platform LSF and IBM Platform Symphony clusters in the cloud reduce time to results and accelerate time to market.
- The IBM Spectrum Scale high-performance file system with that is delivered as a service improves data management and provides seamless transfer between on-premises and cloud infrastructures.
- Non-shared physical machines, InfiniBand interconnect, the latest processor technology, and your choice among SoftLayer data centers leads to optimal application performance and security.
- Integrated workload management with both on-premises and on-cloud infrastructures simplifies management and the user experience, and full support from IBM technical computing experts reduces administrative impact.

7.2 Cloud services architecture

The IBM Platform Computing Cloud Service is built on top of SoftLayer.

SoftLayer deploys the infrastructure in its data centers in the form of *Points of Presence (PoPs)*, which are groups of thousands of machines, petabytes of storage, and all of the networking, firewalls, power distribution, internet connectivity that is needed to support this infrastructure. Theoretically, an IBM client might use an entire PoP, which is more than 60,000 processor cores on a single cluster. A client can request other PoPs to meet demands. Usually, a cluster starts much smaller than at the PoP scale and flexes up or down as their needs dictate. All Platform Computing Cloud Service configurations provide exclusive, non-shared server use for the client. The Cloud Service offers a true cloud-based use model: Pay by the hour or by the month for all elements of the service.

The service is operated by the IBM Platform Computing team, which ensures that HPC experts are available to manage and support your chosen environment. The service uses Platform Computing HPC management and scheduling tools, Platform LSF and Platform Symphony, which provide optimum performance and user experience.

The SaaS architecture counts on the two Platform Computing schedulers, Platform LSF and Platform Symphony, for either traditional HPC clusters or service-oriented architectures (SOA). The service is provided with or without IBM Spectrum Scale (formerly GPFS), which can ease and reduce the data transfer that is needed to and from the cloud by using the Spectrum Scale Active File Management (AFM) facility for caching only the files needed to run the workload on the remote site, in the case of a hybrid cloud. Figure 7-1 shows the high-level architecture.

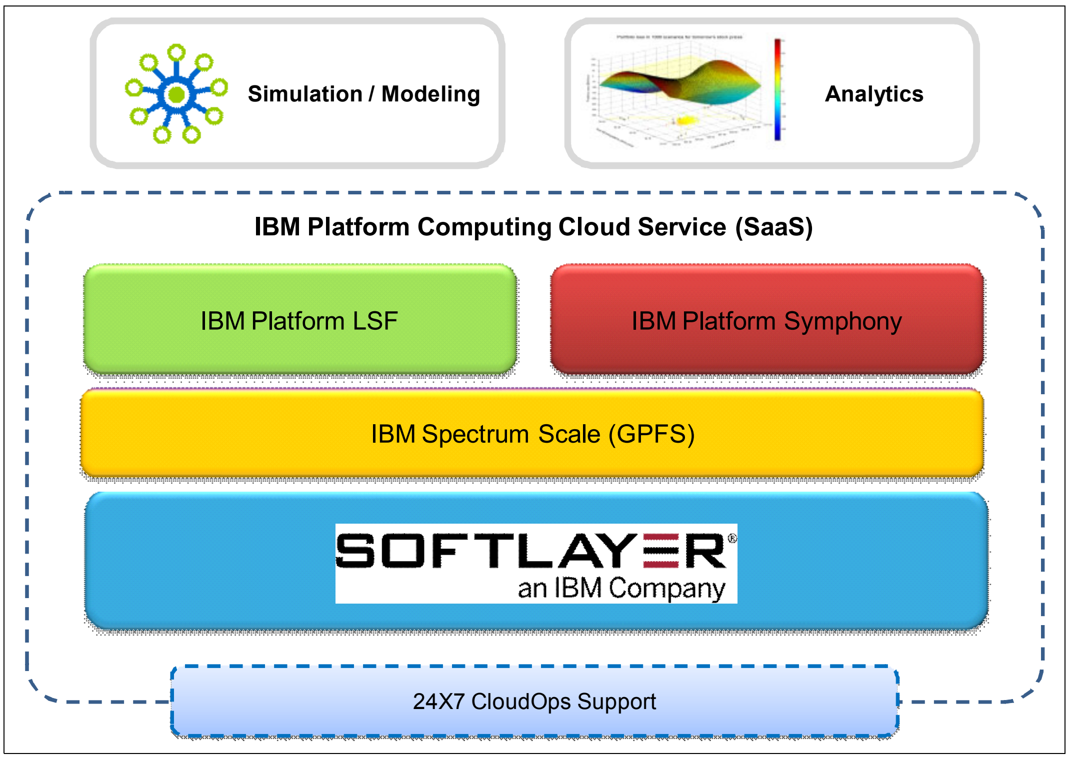


Figure 7-1 IBM Platform Computing Cloud Service high-level architecture

7.3 IBM Spectrum Scale high-performance services

For clients who are considering adding storage capacity or who require more performance and scalability than a network file system (NFS) can provide, Spectrum Scale is now available as a service on the SoftLayer cloud as part of the Platform Computing Cloud Service. With the addition of Spectrum Scale in the cloud, the Platform Computing Cloud Service enables speedy deployment of fully supported, ready-to-run technical computing or analytic environments in the cloud.

Organizations that use the Cloud Service can easily meet additional resource demands without the cost of purchasing or managing an in-house infrastructure. This minimizes the administrative burden and quickly addresses evolving business needs.

7.4 IBM Platform Symphony services

Although a benefit of Platform Symphony is its ability to support diverse applications in a multitenant environment while ensuring service levels, performance tests show that it also helps to provide better performance and efficiency and superior management and monitoring.

If you do not have a specific application to run on Platform Symphony but you need a service environment for your Hadoop workload, see 7.5, “IBM High Performance Services for Hadoop” on page 142.

For more information about how IBM Platform Symphony can help improve your Hadoop workload, see the following website:

http://www.ibm.com/systems/platformcomputing/products/symphony/highperfhadoop.html

7.5 IBM High Performance Services for Hadoop

IBM High Performance Services for Hadoop is a good choice if your organization is looking for a fully supported, ready-to-run Hadoop environment for production use or as a development and testing environment. This service enables you to quickly and easily deploy Hadoop workloads on ready-to-run clusters on the SoftLayer cloud, complete with bare metal SoftLayer infrastructure, a private network, and your choice of a data center to help you achieve optimal performance and security.

An experienced and dedicated cloud operations team configures, deploys, and supports the cloud-based infrastructure and the software. This helps minimize the administrative burden on your organization and the need to develop the skills to design and manage a Hadoop environment.

High Performance Services for Hadoop delivers a Hadoop-ready cluster as a service on SoftLayer, which offers following benefits:

- Rapid access to Hadoop clusters in the cloud for both product use and development testing
- Optimal performance with bare metal resources
- Security through physical isolation and choice of data center location
- Reduced capital expenditure
- Minimal user and administrator impact
- Easy adoption of public cloud technology and resources

IBM High Performance Services for Hadoop also offers the following benefits:

- More capability and lower costs: Easily meet demand without the up-front costs of purchasing in-house infrastructure or the ongoing cost of infrastructure management.
- Match resources to demand, which helps reduce capital expenditures: Minimize administrative costs by using a skilled cloud operations team with deep Hadoop expertise.

- Security:
  - Improve security through physical isolation with a dedicated virtual local area network (VLAN)
  - Upload data securely via virtual private network (VPN) or Multi-Protocol Label Switching (MPLS) to gateway servers
  - Use your SoftLayer data center of choice for regulatory compliance
- Faster time to results:
  - Accelerate Hadoop MapReduce workloads with dedicated bare metal servers.
  - Optimize I/O performance with 10-gigabit Ethernet networking.

7.6 IBM Platform LSF services

IBM Platform LSF is a powerful workload management platform for demanding, distributed HPC environments. It provides a comprehensive set of intelligent, policy-driven scheduling features that enable you to use all of your compute infrastructure resources and ensure optimal application performance.

Platform LSF helps to ensure that all available resources are fully used by enabling you to take full advantage of all technical computing resources in the cloud. Platform LSF helps to ensure that the computing power in the cloud is fully used, and it helps to manage application software licenses use, which is usually expensive for demanding workloads.

The IBM Platform Computing LSF Cloud Service provides the following features:

- A single source for end-to-end cluster support with access to technical computing experts to eliminate the skills barrier for using clustered resources
- Dedicated bare-metal servers and InfiniBand interconnect for applications that require the full capacity of a non-virtualized and parallel computing environment
- Control of data center location, which enables organizations to choose the location where workloads run to protect their information and meet data regulations
- Non-shared physical machines and dedicated network for workloads that require maximum security

7.7 Hybrid Platform LSF on-premises with a cloud service

This scenario describes a transparent user experience that manages workloads between an on-premises cluster and in the cloud can be achieved with IBM Platform LSF Multicluster and IBM Spectrum Scale AFM.

> **Note:** If you already have Platform LSF Standard Edition, skip section 7.7.1, "Upgrading IBM Platform HPC to enable multicluster functions" on page 144. Otherwise, see the IBM Redbooks publications titled *IBM Platform Computing Integration Solutions*, SG24-8081, and the *IBM Platform Computing Solutions Reference Architectures and Best Practices*, SG24-8169 for details about how to implement IBM Platform HPC.

7.7.1 Upgrading IBM Platform HPC to enable multicluster functions

To start the upgrade, you need the name of your Platform LSF installation directory (LSF_TOP in this example), the Platform LSF administrators (LSF_ADMINS), and the cluster name (LSF_CLUSTER_NAME) available. If you do not have this information, run the commands shown in Example 7-1 to gather the information.

Example 7-1 Gathering information for Platform LSF Standard Edition upgrade

```
[root@homecluster etc]# grep LSF_TOP $PCMD_TOP_LOCAL/etc/lsf.install.config
LSF_TOP="/shared/ibm/platform_lsf"
[root@homecluster etc]# grep LSF_ADMINS $PCMD_TOP_LOCAL/etc/lsf.install.config
LSF_ADMINS="phpcadmin root"
[root@homecluster etc]#
[root@homecluster etc]# grep CLUSTER $PCMD_TOP_LOCAL/etc/lsf.install.config
LSF_CLUSTER_NAME="phpc_cluster"
[root@homecluster etc]#
```

The information that is gathered by the commands in Example 7-1 is necessary to upgrade IBM Platform HPC and install IBM Platform LSF Standard Edition in the cluster.

To start, add the parameters to the `install.config` file from your Platform LSF installation directory, as shown in Example 7-2. You must add the path for the Platform LSF Standard Edition entitlement file of the installation configuration file, for example:

```
LSF_ENTITLEMENT_FILE="/tmp/phpc/platform_lsf_std_entitlement.dat"
```

Example 7-2 Configuration file to install Platform LSF (install.config)

```
#**********************************************************
#           LSF 9.1.3 INSTALL.CONFIG FILE
#**********************************************************
#
# Name:     install.config
#
# Purpose:  LSF installation options
#
# $Id$
#
# File Format:
#    o  Options (without # sign) can only appear once in the file.
#    o  Blank lines and lines starting with a pound sign (#) are ignored.
#
# Option Format:
#    o  Each disabled example looks like this:
#       # -----------------
#       # LSF_OPTION_NAME="EXAMPLE_VALUE"
#       # -----------------
#
#    o  An enabled option looks like this:
#       # -----------------
#       LSF_OPTION_NAME="ACTUAL_VALUE"
#       # -----------------
#
# Instructions:
#    1. Edit install.config to specify the options for
#       your cluster. Uncomment the options you want and
```

```
#       replace the EXAMPLE values with your own settings.
#       Note that the sample values shown in this template
#       are EXAMPLES only. They are not always the default
#       installation values.
#
#    2. Run ./lsfinstall -f install.config
#
#
#**********************************************************
#           PART 1: REQUIRED PARAMETERS
#     (During an upgrade, specify the existing value.)
#**********************************************************
# -----------------
LSF_TOP="/shared/ibm/platform_lsf"
# -----------------
# Full path to the top-level installation directory {REQUIRED}
#
# The path to LSF_TOP must be shared and accessible to all hosts
# in the cluster. It cannot be the root directory (/).
# The file system containing LSF_TOP must have enough disk space for
# all host types (approximately 300 MB per host type).
#
# -----------------
LSF_ADMINS="phpcadmin root"
# -----------------
# List of LSF administrators {REQUIRED}
#
# The first user account name in the list is the primary LSF
# administrator. It cannot be the root user account.
# Typically, this account is named lsfadmin.
# It owns the LSF configuration files and log files for job events.
# It also has permission to reconfigure LSF and to control batch
# jobs submitted by other users. It typically does not have
# authority to start LSF daemons. Usually, only root has
# permission to start LSF daemons.
# All the LSF administrator accounts must exist on all hosts in the
# cluster before you install LSF.
# Secondary LSF administrators are optional.
#
# -----------------
LSF_CLUSTER_NAME="phpc_cluster"
# -----------------
# Name of the LSF cluster {REQUIRED}
#
# It must be 39 characters or less, and cannot contain any
# white spaces. Do not use the name of any host, user, or user group
# as the name of your cluster.
#
#
#**********************************************************
#          PART 2: PRIMARY PARAMETERS FOR NEW INSTALL
#  (These parameters are ignored if they are already defined in the cluster.)
#**********************************************************
#
# -----------------
```

```
# LSF_MASTER_LIST="hostm hosta hostc"
# -----------------
# List of LSF server hosts to be master or master candidate in the
# cluster {REQUIRED when you install for the first time or during
# upgrade if the parameter does not already exist.}
#
# You must specify at least one valid server host to start the
# cluster. The first host listed is the LSF master host.
#
# -----------------
LSF_ENTITLEMENT_FILE="/tmp/phpc/platform_lsf_std_entitlement.dat"
# -----------------
# You must specify a full path to the LSF entitlement file.
#
...
```

To perform the update after you enter all of the environment variables in the configuration file (see Example 7-2 on page 144), run **lsfinstall**, as shown in Example 7-3.

Example 7-3 Running the lsfinstall command

```
[root@homecluster lsf9.1.3_lsfinstall]# ./lsfinstall -f install.config

Logging installation sequence in /tmp/phpc/lsf9.1.3_lsfinstall/Install.log

International Program License Agreement

.
.
.

Press Enter to continue viewing the license agreement, or
enter "1" to accept the agreement, "2" to decline it, "3"
to print it, "4" to read non-IBM terms, or "99" to go back
to the previous screen.
```

Read and accept the license agreement to proceed with the installation. After you finish reading and agreeing to the terms, when you press the number 1 key, Platform LSF checks for the prerequisites.

If the prerequisites are met, the installer prompts for the distribution TAR file to be used, as shown in Example 7-4.

Example 7-4 Platform LSF preinstall check and distribution selection

```
LSF pre-installation check ...

Checking the LSF TOP directory /shared/ibm/platform_lsf ...
... Done checking the LSF TOP directory /shared/ibm/platform_lsf ...

You are installing IBM Platform LSF - 9.1.3 Standard Edition.

Checking LSF Administrators ...
   LSF administrator(s):       " phpcadmin root"
   Primary LSF administrator:  "phpcadmin"
Checking the configuration template  ...
```

```
CONFIGURATION_TEMPLATE not defined. Using DEFAULT template.
    Done checking configuration template ...
    Done checking ENABLE_STREAM ...

Checking the patch history directory  ...
... Done checking the patch history directory /shared/ibm/platform_lsf/patch ...

Checking the patch backup directory ...
... Done checking the patch backup directory /shared/ibm/platform_lsf/patch/backup
...


Searching LSF 9.1.3 distribution tar files in /tmp/phpc Please wait ...

  1) linux2.6-glibc2.3-x86_64

Press 1 or Enter to install this host type: 1
```

The installation proceeds without further prompts until a message is displayed similar to the one that is shown in Example 7-5.

Example 7-5 Installation completed successfully

```
You have chosen the following tar file(s):
    lsf9.1.3_linux2.6-glibc2.3-x86_64

Checking selected tar file(s) ...
... Done checking selected tar file(s).


Pre-installation check report saved as text file:
/tmp/phpc/lsf9.1.3_lsfinstall/prechk.rpt.

... Done LSF pre-installation check.

.
.
.

Creating lsf_quick_admin.html ...
... Done creating lsf_quick_admin.html

lsfinstall is done.

To complete your LSF installation and get your
cluster "phpc_cluster" up and running, follow the steps in
"/tmp/phpc/lsf9.1.3_lsfinstall/lsf_getting_started.html".

After setting up your LSF server hosts and verifying
your cluster "phpc_cluster" is running correctly,
see "/shared/ibm/platform_lsf/9.1/lsf_quick_admin.html"
to learn more about your new LSF cluster.

After installation, remember to bring your cluster up to date
by applying the latest updates and bug fixes.
```

Note: For the latest release information about IBM Platform LSF Version 9.1.3, see IBM Platform LSF in IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSETD4_9.1.3/lsf_welcome.html

For the latest information about IBM Platform HPC, see “Release notes for IBM Platform HPC” in IBM Knowledge Center:

http://bit.ly/1R1KwmM

Now, restart Platform HPC services to enable the new Platform LSF entitlement, as shown in Example 7-6.

Example 7-6 Restarting Platform HPC Services

```
[root@homecluster platform_lsf]# service phpc stop
Stopping Web Portal services                            [  OK  ]
Stopping PERF services                                  [  OK  ]
Stopping Rule Engine service                            [  OK  ]
Stopping PCMD service                                   [  OK  ]
Stopping Message broker                                 [  OK  ]
Stopping the LSF subsystem                              [  OK  ]

Stopping Platform HPC Services:                         [  OK  ]
[root@homecluster platform_lsf]# service phpc start
Checking for xcatd service started                      [  OK  ]

Starting the LSF subsystem                              [  OK  ]
- Waiting for EGO service started ..                    [  OK  ]

Cluster name : phpc_cluster EGO master host name : homecluster EGO master version
: 1.2.10
- Waiting for PCM master node online ..........         [  OK  ]

Starting PERF services                                  [  OK  ]
Starting Message broker                                 [  OK  ]
Starting PCMD service                                   [  OK  ]
Starting Rule Engine service                            [  OK  ]
Starting Web Portal services                            [  OK  ]

Starting Platform HPC Services:                         [  OK  ]
[root@homecluster platform_lsf]#
```

Your cluster is ready to be configured as a multicluster server.

7.7.2 Tasks to install IBM Platform LSF in the cloud

The IBM Cloud Services team installs and configures the Platform LSF cluster for you. You set up only your connection to the new cluster and configure your new multicluster feature.

Note: You can connect to the cloud network by using a VPN or MPLS. To connect, add only the name of the master and master candidates of the new cloud cluster to your Domain Name System (DNS) or hosts file, and exchange the SSH keys between the hosts.

7.7.3 Configuring the multicluster feature

After you have exchanged the SSH keys, enable the multicluster feature. Copy both of your cluster definitions files (on-premises and in the cloud) in both `$LSF_TOP/conf/` directories, as shown in Example 7-7.

Example 7-7 Copying cluster definition files between master nodes

```
[root@homecluster conf]# scp
softlayer:/usr/share/lsf/conf/lsf.cluster.HPC_Services
/shared/ibm/platform_lsf/conf/
lsf.cluster.HPC_Services
100% 1801     1.8KB/s   00:00
[root@homecluster conf]# scp
/shared/ibm/platform_lsf/conf/lsf.cluster.phpc_cluster
softlayer:/usr/share/lsf/conf/
lsf.cluster.phpc_cluster 100% 2897 2.8KB/s   00:00
```

Edit the `$LSF_TOP/conf/lsf.shared` file and check that all the clusters are defined in the cluster stanza file, as shown in Example 7-8.

Example 7-8 LSF shared configuration file containing both clusters

```
# $Revision$Date$
# ----------------------------------------------------------------------
# T H I S   F I L E:  Is shared by all clusters in the LSF system.
#
# This file contains all definitions referenced by individual
# lsf.cluster.<clustername> files. The definitions in this file can be
# a superset, i.e., not all definitions in this file need to be used in
# other files.
#
# See lsf.cluster(5) and "LSF User's and Administrator's Guide".
# ----------------------------------------------------------------------

Begin Cluster
ClusterName                   # Keyword
phpc_cluster
HPC_Services
End Cluster
```

Note: Make the `lsf.shared` file the same on both clusters.

Now, as shown in Example 7-9, add a module in the `lsb.modules` file in the local cluster to see resources in the remote cluster. In this example, the file is in the home cluster server:

`/install/shared/ibm/platform_lsf/conf/lsbatch/phpc_cluster/configdir/lsb.modules`

Example 7-9 Add schmod_mc on lsb.modules

```
# $Revision$Date$

# Define plugins for Scheduler and Resource Broker.
# SCH_PLUGIN coloum specifies the share module name for Scheduler, while
# RB_PLUGIN specifies the share module name for Resource Broker
# A Scheduler plugin can have one, multiple, or none RB plugins
# corresponding to it.
```

```
# SCH_DISABLE_PHASES specifies which phases of that scheduler plugin
# should be disabled, i.e., inactivated. A scheduler plugin has 4 phases:
# pre processing, match/limit, order/alloc, post processing. Scheduler
# won't invokes disabled phases over jobs
# Note all share modules should be put under LSF_LIBDIR

Begin PluginModule
SCH_PLUGIN                      RB_PLUGIN                    SCH_DISABLE_PHASES
schmod_default                  ()                              ()
schmod_fcfs                     ()                              ()
schmod_fairshare                ()                              ()
schmod_limit                    ()                              ()
schmod_mc                       ()                              ()
schmod_parallel                 ()                              ()
schmod_reserve                  ()                              ()
schmod_preemption               ()                              ()
schmod_advrsv                   ()                              ()
schmod_ps                       ()                              ()
#schmod_dc                      ()                              ()
End PluginModule
```

Restart the Platform LSF services on both the on-premises cluster and the cloud cluster, as shown in Example 7-10.

Example 7-10 Restart Platform LSF services

```
[root@homecluster conf]# lsadmin limrestart all

Checking configuration files ...
No errors found.

Do you really want to restart LIMs on all hosts? [y/n] y
Restart LIM on <homecluster> ...... done

[root@homecluster conf]# badmin mbdrestart

Checking configuration files ...

No errors found.
MBD restart initiated

[root@softlayer ~]# lsadmin limrestart all

Checking configuration files ...
No errors found.

Do you really want to restart LIMs on all hosts? [y/n] y
Restart LIM on <softlayer> ...... done
[root@softlayer ~]# badmin mbdrestart

Checking configuration files ...

No errors found.
MBD restart initiated
[root@softlayer ~]#
```

To check whether the multicluster feature is correctly configured and these clusters are enabled to access each other, run **lsclusters** and **bclusters** at the prompt to get OK status responses from both clusters, as shown in Example 7-11.

Example 7-11 Check the configuration

```
[root@homecluster ~]# lsclusters
CLUSTER_NAME   STATUS   MASTER_HOST             ADMIN    HOSTS  SERVERS
phpc_cluster   ok       homecluster          phpcadmin       1        1
HPC_Services   ok       softlayer             lsfadmin       1        1
[root@homecluster ~]# bclusters
[Job Forwarding Information ]
No local queue sending/receiving jobs from remote clusters

[Resource Lease Information ]
No resources have been exported or borrowed
[root@homecluster ~]#
```

7.7.4 Configuring job forwarding

This scenario shows how to change the high-priority queue to send jobs to the Platform Computing cloud services cluster. To do this, change the `high_priority` stanza in the `lsb.queues` file at the local cluster (`phpc_cluster`). In this scenario, this is the master node home cluster path:

`/install/shared/ibm/platform_lsf/conf/lsbatch/phpc_cluster/configdir/lsb.queues`

This scenario does not preempt running jobs, because the idea is to show how to send jobs to the cloud rather than interrupting a running job, so comment the `PREEMPTION` line and add the `SNDJOBS_TO` to point to the remote cluster.

Then, change the description to state the use of the queue. All changes are shown in bold in Example 7-12.

Example 7-12 File lsb.queues on the local cluster

```
Begin Queue
QUEUE_NAME    = high_priority
PRIORITY      = 43
NICE          = 10
SNDJOBS_TO    = receive@HPC_Services
#PREEMPTION = PREEMPTIVE
#RUN_WINDOW
#CPULIMIT      = 8:0/SunIPC     # 8 hours of host model SunIPC
#FILELIMIT     = 20000
#DATALIMIT     = 20000          # jobs data segment limit
#CORELIMIT     = 20000
#PROCLIMIT    = 5              # job processor limit
#USERS         = user1 user2 user3
#HOSTS         = all
#ADMINISTRATORS       = user1 user3
#EXCLUSIVE     = N
#PRE_EXEC     = /usr/local/lsf/misc/testq_pre >> /tmp/pre.out
#POST_EXEC    = /usr/local/lsf/misc/testq_post |grep -v "Hey"
#REQUEUE_EXIT_VALUES = 55 255 78
DESCRIPTION   =  Jobs submitted for this queue are scheduled as urgent\
```

```
jobs. Jobs in this queue can  jobs in this queue can be fowarded to the Cloud
Services Cluster.
End Queue
```

In a similar fashion, configure the receiving side to handle the jobs coming from the high-priority queue. For this scenario, the `lsb.queues` files in the remote cluster can be found in the SoftLayer host of the HPC_Services cluster at the following path:

`/usr/share/lsf/conf/lsbatch/HPC_Services/configdir/lsb.queues`

In `lsb.queues`, add a stanza at the end of the file as shown in Example 7-13.

Example 7-13 lsb.queues on the remote cluster

```
Begin Queue
QUEUE_NAME=receive
RCVJOBS_FROM=high_priority@phpc_cluster
PRIORITY=70
NICE=20
End Queue
```

Now, reconfigure the queues on both sides, as shown in Example 7-14.

Example 7-14 Reconfigure the queues

```
[root@softlayer ~]# badmin mbdrestart

Checking configuration files ...

No errors found.
MBD restart initiated
[root@softlayer ~]#

[root@homecluster ~]# badmin mbdrestart

Checking configuration files ...

No errors found.
MBD restart initiated
[root@homecluster ~]#
```

Example 7-15 shows how to check the job-forwarding status configuration for the local and remote queues.

Example 7-15 Check job forwarding

```
[root@homecluster ~]# bclusters
[Job Forwarding Information ]
LOCAL_QUEUE     JOB_FLOW   REMOTE     CLUSTER    STATUS
high_priority   send       receive    HPC_Servic ok

[Resource Lease Information ]
No resources have been exported or borrowed
[root@homecluster ~]#

[root@softlayer ~]# bclusters
[Job Forwarding Information ]
```

```
LOCAL_QUEUE     JOB_FLOW   REMOTE     CLUSTER    STATUS
receive         recv       -          phpc_clust ok

[Resource Lease Information ]
No resources have been exported or borrowed
[root@softlayer ~]#
```

7.7.5 Testing your configuration

Now, test the new configuration by using the command-line interface (CLI) or the graphical user interface (GUI) from Platform HPC. If you have Platform LSF with Platform Application Center, you can use this interface, too.

> **Note:** To submit the job to the cloud, the user must have authority to run jobs on the receiving queue.

This scenario uses the CLI to submit the jobs. Example 7-16 shows how to issue the **bsub** command to a dummy sleep job.

Example 7-16 Submit jobs to the respective queues

```
[root@homecluster ~]#  bsub -q high_priority sleep 50
Job <857> is submitted to queue <high_priority>.
[root@homecluster ~]#
[root@homecluster ~]#  bsub -q medium_priority sleep 50
Job <858> is submitted to queue <medium_priority>.
[root@homecluster ~]#
[root@homecluster ~]#  bsub -q medium_priority sleep 50
Job <859> is submitted to queue <medium_priority>.
[root@homecluster ~]#
[root@homecluster ~]#  bsub -q high_priority sleep 50
Job <860> is submitted to queue <high_priority>.
[root@homecluster ~]#
```

In this example, four consecutive jobs are submitted in a row, but the only queue that can forward jobs is the high-priority one. There are only two slots in the on-premises environment, so submit three jobs in the medium-priority queue and the last job in the high-priority queue. Only the job with the high priority runs in the cloud.

Example 7-17 shows the running jobs and the pending jobs in their respective queues.

Example 7-17 Jobs running in the cluster

```
[root@homecluster ~]# bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
857     root    RUN   high_priori homecluster homecluster *813545588 Mar 31 10:52
858     root    RUN   medium_prio homecluster homecluster *813554430 Mar 31 10:52
860     root    RUN   high_priori homecluster softlayer@H *813563747 Mar 31 10:53
859     root    PEND  medium_prio homecluster homecluster *813557373 Mar 31 10:52
```

As you can see, the first preference is to use available slots in the home cluster. After there is no resource available, only the high-priority job goes to the cloud, even after being submitted after the last medium-priority job.

Note: This is an example on how Platform LSF sends jobs only from a configured queue. Platform LSF is a powerful tool that helps you do advanced scheduling and provides the best policies to suit your business needs.

7.7.6 Hybrid cloud capacity increases and assistance

The previous sections describe how to configure a hybrid cloud in a few steps, although, with the help of IBM Platform Computing cloud services, you do not need to worry about configuring and managing a cloud infrastructure.

After following the five simple steps in previous sections, you can add extra capacity to receive jobs from your existing environment.

If you need assistance to configure a hybrid cloud environment, contact the IBM Platform Computing Services team.

7.8 Data management on hybrid clouds

Two easy ways to manage data across hybrid clouds are by implementing IBM Platform Data Manager for LSF and IBM Spectrum Scale Active File Management (AFM). Both technologies optimize data transfer needs to reduce costs and time to results because only the required data is moved at the correct time.

7.8.1 IBM Platform Data Manager for LSF

Platform Data Manager for LSF automates the transfer of data that is used by application workloads running on Platform LSF clusters and in the cloud. Frequently used data that is transferred between multiple data centers and the cloud can be stored in a smart, managed cache closer to compute resources. This smart data management helps improve data throughput and minimizes wasted compute cycles, which helps you lower storage costs in the cloud.

With Platform Data Manager, the following actions occur:

- Data is staged in and out independently of workloads, freeing compute resources while data is transferred behind the scenes.
- A smart, managed cache reuses transferred data and avoids duplication of data transfers, sharing cached copies with all workloads that need access to the data and among multiple users where appropriate.
- Data transfers are scheduled as jobs in Platform LSF and are subject to Platform LSF scheduling policies that are established by administrators, including priorities.

For more information about IBM Platform Data Manager for LSF, see the following website:

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

7.8.2 IBM Spectrum Scale Active File Management

AFM is a scalable, high-performance, file system caching layer that is integrated with Spectrum Scale. You can use AFM in two key ways:

- To create associations from a local cluster to a remote cluster or storage
- To define the location and flow of file data to automate data management to implement a single namespace view across sites around the world

AFM masks wide area network (WAN) latencies and outages by using Spectrum Scale to cache massive data sets. This allows data access and modifications even when a remote storage cluster is unavailable. AFM also updates the remote cluster asynchronously, which allows applications to continue operating while not being constrained by limited outgoing network bandwidth.

The AFM implementation uses the inherent scalability of Spectrum Scale to provide a multinode, consistent cache of data that is in a home cluster. By integrating it with the file system, AFM provides a Portable Operating System Interface (POSIX)-compliant interface, making the cache transparent to applications. AFM is easy to deploy, because it relies on open standards for high-performance file serving and does not require any proprietary hardware or software to be installed at the home cluster.

For a step-by-step configuration information, see “Using AFM” on the IBM developerWorks website:

http://ibm.co/1bPKBfY

Related publications

The publications listed in this section are useful for more information about the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional relevant information. Some publications referenced in this list might be available in softcopy only.

- *Implementing IBM InfoSphere BigInsights on IBM System x*, SG24-8077
- *IBM Platform Computing Solutions*, SG24-8073
- *IBM Platform Computing Solutions Reference Architectures and Best Practices*, SG24-8169
- *IBM Spectrum Scale (formerly GPFS)*, SG24-8254

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following web page:

ibm.com/redbooks

Other publications

This publication is also relevant for further information:

- *Implementing an Advanced Application Using Processes, Rules, Events, and Reports*, SG24-8065

Online resources

These websites are also relevant as further information sources:

- IBM Platform Computing

  http://www.ibm.com/systems/platformcomputing/products/index.html
- Big Data and the Speed of Business

  http://www.ibm.com/software/data/bigdata/industry.html
- Algorithmics Software

  http://www.ibm.com/software/analytics/algorithmics/
- IBM Watson: Ushering in a new era of computing

  http://www.ibm.com/innovation/ca/en/watson/watson_in_healthcare.shtml

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

IBM Software Defined Infrastructure for Big Data Analytics Workloads

(0.2”spine)
0.17”<->0.473”
90<->249 pages

SG24-8265-00

ISBN 0738440779

Printed in U.S.A.