

Implementing IBM InfoSphere BigInsights on IBM System x

Introducing big data and IBM
InfoSphere BigInsights

Installing an InfoSphere
BigInsights environment

Monitoring and securing
InfoSphere BigInsights



Mike Ebbers
Renata Ghislotti de Souza
Marcelo Correia Lima
Peter McCullagh
Michael Nobles
Dustin VanStee
Brandon Waters

Redbooks



International Technical Support Organization

**Implementing IBM InfoSphere BigInsights on IBM
System x**

June 2013

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

Second Edition (2013)

This edition applies to IBM InfoSphere BigInsights Enterprise Edition Version 1.4.0.0.

© Copyright International Business Machines Corporation 2013. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	ix
Trademarks	x
Preface	xi
Authors	xi
Now you can become a published author, too!	xii
Comments welcome	xiii
Stay connected to IBM Redbooks	xiii
Summary of changes	xv
June 2013, Second Edition	xv
Chapter 1. A whole new world of big data	1
1.1 What is big data	2
1.2 The big data challenge	4
1.2.1 The traditional data warehouse in relation to big data	4
1.2.2 How continual data growth affects data warehouse storage	5
1.3 How IBM is answering the big data challenge	7
1.3.1 Big data platform	7
1.3.2 Big data Enterprise Engines	9
1.4 Why you should care	10
Chapter 2. Why choose BigInsights	13
2.1 BigInsights introduction	14
2.2 What is Hadoop?	14
2.2.1 Hadoop Distributed File System in more detail	15
2.2.2 MapReduce in more detail	16
2.3 What is BigInsights?	18
2.3.1 All-in-one installation	18
2.3.2 Integration with existing information architectures	19
2.3.3 Enterprise class support	19
2.3.4 Enterprise class functionality	19
2.3.5 BigSheets	20
2.3.6 BigInsights scheduler	20
2.3.7 Text analytics	23
2.4 BigInsights and the traditional data warehouse	23
2.4.1 How can BigInsights complement my data warehouse?	24
2.5 Use cases for BigInsights	24
2.5.1 Industry-based use cases	25
2.5.2 Social Media use case	25
Chapter 3. BigInsights network architecture	29
3.1 Network design overview	30
3.2 Logical network planning	30
3.2.1 Deciding between 1 Gbps and 10 Gbps	31
3.2.2 Switch and Node Adapter redundancy: costs and trade-offs	32
3.3 Networking zones	32
3.3.1 Corporate Management network	33
3.3.2 Corporate Administration network	33

3.3.3 Private Data network	34
3.3.4 Optional Head Node configuration considerations	34
3.4 Network configuration options	35
3.4.1 Value configuration	35
3.4.2 Performance configuration	36
3.4.3 Enterprise option	37
3.5 Suggested IBM system networking switches	39
3.5.1 Value configuration switches	40
3.5.2 Performance configuration switch	40
3.6 How to work with multiple racks	40
3.6.1 Value configuration	41
3.6.2 Performance configuration	42
3.6.3 Enterprise option	44
3.7 How to improve performance	46
3.7.1 Network port bonding	46
3.7.2 Extra capacity through more hardware provided for redundancy	46
3.7.3 Virtual Link Aggregation Groups for greater multi-rack throughput	46
3.8 Physical network planning	47
3.8.1 IP address quantities and networking into existing corporate networks	47
3.8.2 Power and cooling	47
Chapter 4. BigInsights hardware architecture	49
4.1 Roles of the management and data nodes	50
4.1.1 The management node	50
4.1.2 The data node	50
4.2 Using multiple management nodes	50
4.3 Storage and adapters used in the hardware architecture	51
4.3.1 RAID versus JBOD	51
4.3.2 Disk virtualization	51
4.3.3 Compression	51
4.4 The IBM hardware portfolio	52
4.4.1 The IBM System x3550 M4 as a management node	52
4.4.2 The IBM System x3630 M4 as a data node	53
4.5 Lead configuration for the BigInsights management node	55
4.5.1 Use two E5-2650, 2.0 GHz, 8-core processors in your management node	55
4.5.2 Memory for your management node	56
4.5.3 Dual power cables per management node	56
4.5.4 Two network adapters per management node	57
4.5.5 Storage controllers on the management node	57
4.5.6 Hard disk drives in the management node	57
4.6 Lead configuration for the BigInsights data node	57
4.6.1 Processor options for the data node	57
4.6.2 Memory considerations for the data node	58
4.6.3 Other considerations for the data node	59
4.6.4 Data node configuration options	60
4.6.5 Pre-defined rack configurations	60
4.6.6 Storage considerations	61
4.6.7 Basic input/output system tool	62
Chapter 5. Operating system prerequisites for BigInsights	65
5.1 Prerequisite software	66
5.1.1 Operating provisioning software	66
5.1.2 Yellowdog Updater Modified repository	66

5.1.3	Operating system packages	66
5.2	Operating system settings related to software	67
5.2.1	System clock synchronization	67
5.2.2	Services to disable for improved performance	67
5.2.3	Raising the ulimits setting to accommodate Hadoop's data processing within BigInsights	67
5.2.4	Optional: set up password-less Secure Shell	68
5.3	Optionally configure /etc/hosts	68
5.4	Operating system settings related to hardware	69
5.4.1	Operating system level settings if optional network cards were added	69
5.4.2	Storage configuration	71
Chapter 6. BigInsights installation		73
6.1	Preparing the environment for installation	74
6.2	Installing BigInsights using the graphical user interface	74
6.3	Silent installation of BigInsights	84
6.3.1	Installing BigInsights using the silent installation option	84
6.4	How to install the Eclipse plug-in	88
6.5	Common installation pitfalls	90
Chapter 7. Cluster validation		97
7.1	Cluster validation	98
7.1.1	Initial validation	98
7.1.2	Running the built-in health check utility	98
7.1.3	Simple applications to run	100
7.2	Performance considerations	102
7.3	TeraSort scalability and performance test example	105
7.4	Other useful scripts	107
7.4.1	addnode.sh	108
7.4.2	credstore.sh	108
7.4.3	synconf.sh	108
7.4.4	start.sh, stop.sh, start-all.sh, and stop-all.sh	108
7.4.5	status.sh	108
Chapter 8. BigInsights capabilities		109
8.1	Data ingestion	110
8.1.1	Loading data from files using the web console	110
8.1.2	Loading files from the command line	112
8.1.3	Loading data from a data warehouse	112
8.1.4	Loading frequently updated files	113
8.2	BigSheets	115
8.3	Web console	115
8.4	Text Analytics	115
8.4.1	Text analytics architecture	116
8.4.2	Log file processing example	116
Chapter 9. BigInsights hardware monitoring and alerting		125
9.1	BigInsights monitoring	126
9.1.1	Workflows and scheduled workflows	126
9.1.2	MapReduce jobs	128
9.1.3	Job and task counters	129
9.2	Nigel's monitor	131
9.2.1	nmon within a shell terminal	131
9.2.2	Saving nmon output to a file	134

9.3	Ganglia	134
9.3.1	Ganglia installation (optional)	135
9.3.2	Ganglia configuration (if installed)	136
9.3.3	Multicast versus unicast	138
9.3.4	Large cluster considerations	139
9.3.5	BigInsights 1.4 configuration to enable Hadoop metrics with Ganglia	139
9.4	Nagios	140
9.5	IBM Tivoli OMNibus and Network Manager	141
9.5.1	Tivoli Netcool Configuration Manager	141
9.5.2	Highlights of Tivoli Netcool Configuration Manager	142
9.5.3	IBM Tivoli Netcool/OMNibus	142
9.5.4	IBM Tivoli Network Manager IP	142
9.6	IBM System Networking Element Manager	143
9.6.1	Product features	143
9.6.2	Software summary	143
	Chapter 10. BigInsights security design	145
10.1	BigInsights security overview	146
10.2	Authorization	147
10.2.1	Roles	147
10.3	Authentication	149
10.3.1	Flat file	149
10.3.2	Lightweight Directory Access Protocol	151
10.3.3	Pluggable Authentication Module	154
10.4	Secure browser support	155
	Chapter 11. IBM Platform Symphony	159
11.1	Overview	160
11.2	The changing nature of distributed computing	160
11.3	About IBM Platform Symphony	161
11.4	IBM Platform Symphony architecture	161
11.5	Platform Symphony MapReduce framework	162
11.6	Multi-tenancy built in	164
11.7	How InfoSphere BigInsights works with Platform Symphony	166
11.8	Understanding the Platform Symphony performance benefit	170
11.9	Supported applications	171
11.10	BigInsights versions supported	171
11.11	Summary	172
	Appendix A. M4 reference architecture	173
	The M4 series of servers: Bill of materials	174
	IBM x3630 M4: The data node	174
	IBM System x3550 M4: The management node	175
	Recommended features for BigInsights x3550 M4 management node	177
	RAID adapter	177
	Appendix B. Installation values	179
	BigInsights default installation values	180
	Open source technologies and version numbers	181
	Ganglia monitoring options	182
	Appendix C. Checklist	189
	BIOS settings to check	190
	Networking settings to verify operating system	190

Operating system settings to check	191
Non-package-related items	191
BigInsights configuration changes to consider	192
Installed Red Hat package items	194
Related publications	201
IBM Redbooks	201
Other publications	201
Online resources	201
Help from IBM	202

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](http://www.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®	IBM®	Symphony®
BigInsights™	Information Agenda®	System p®
BladeCenter®	InfoSphere®	System x®
Cognos®	Netcool®	System z®
DataStage®	RackSwitch™	Tivoli®
DB2®	Redbooks®	WebSphere®
developerWorks®	Redbooks (logo)  ®	
GPFS™	Smarter Planet®	

The following terms are trademarks of other companies:

Netezza, and N logo are trademarks or registered trademarks of IBM International Group B.V., 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.

Preface

As world activities become more integrated, the rate of data growth has been increasing exponentially. And as a result of this data explosion, current data management methods can become inadequate. People are using the term *big data* (sometimes referred to as *Big Data*) to describe this latest industry trend. IBM® is preparing the next generation of technology to meet these data management challenges.

To provide the capability of incorporating big data sources and analytics of these sources, IBM developed a stream-computing product that is based on the open source computing framework Apache Hadoop. Each product in the framework provides unique capabilities to the data management environment, and further enhances the value of your data warehouse investment.

In this IBM Redbooks® publication, we describe the need for big data in an organization. We then introduce IBM InfoSphere® BigInsights™ and explain how it differs from standard Hadoop. BigInsights provides a packaged Hadoop distribution, a greatly simplified installation of Hadoop and corresponding open source tools for application development, data movement, and cluster management. BigInsights also brings more options for data security, and as a component of the IBM big data platform, provides potential integration points with the other components of the platform.

A new chapter has been added to this edition. Chapter 11 describes IBM Platform Symphony®, which is a new scheduling product that works with IBM Insights, bringing low-latency scheduling and multi-tenancy to IBM InfoSphere BigInsights.

The book is designed for clients, consultants, and other technical professionals.

Authors

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

Mike Ebbers is a Project Leader and Consulting IT Specialist at the IBM ITSO, Poughkeepsie Center. He has worked for IBM since 1974 in the field, in education, and as a manager. He has been with the ITSO since 1994.

Renata Ghislotti de Souza is a Software Engineer for IBM Brazil. She has over six years of experience in the Linux/Open Source field and holds a degree in Computer Science from UNICAMP. Her areas of expertise include Data Mining, Apache Hadoop, and open source software.

Marcelo Correia Lima is a Business Intelligence Architect at IBM. He has 15 years of experience in leading development and integration of Enterprise Applications. His current area of expertise is Business Analytics Optimization (BAO) Solutions. He has been planning and managing full lifecycle implementation of many projects, involving Multidimensional Modeling, Multidimensional Clustering and Partitioning, IBM InfoSphere Data Architect, IBM InfoSphere DataStage®, IBM Cognos® Business Intelligence and IBM DB2®. Recently, Marcelo has added Hadoop, Big Data, and IBM InfoSphere BigInsights to his background. 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.

Peter McCullagh is a Technology Consultant who works in the UK. He holds a degree in Chemistry from the University of Bristol. Peter is a member of the IBM Software Group (SWG) Services Big Data team, and works with several IBM products including InfoSphere BigInsights, InfoSphere Streams, DB2, and IBM Smart Analytics System.

Michael Nobles is a Big Data, Consulting Solution Specialist who works for IBM in the US. He has been involved in various aspects of software development since 1992. In 2001, he started working in the area of Business Intelligence (BI) covering many industries, and joined IBM in 2004. More recently, Michael has added Hadoop, Big Data, and real-time Business Intelligence to his areas of expertise. Specializing in the IBM products of InfoSphere BigInsights and InfoSphere Streams, he is working to help the world, one company at a time, with their *data in motion* and *data at rest* challenges. As a Technical Pre-sales Professional, covering North America on the IBM Big Data, Advanced Data Processing Team, Michael looks forward to helping his clients grow and succeed in this new world of Big Data.

Dustin VanStee is a Big Data Benchmarking Specialist for IBM Systems and Technology Group in the US. He has 13 years of experience with IBM in various fields including hardware design for IBM System p® and IBM System z® servers, SSD technology evaluation, and currently Big Data benchmarking. Dustin holds a Masters degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute. His areas of expertise include solid-state drive technology and Apache Hadoop.

Brandon Waters is a Big Data Client Technical Professional in the US for the IBM Federal Software Group. He has been with IBM since 2006 after obtaining a Master's degree in Electrical Engineering from the Virginia Polytechnic and State University. Beginning his career with a focus in database administration and tooling, Brandon's area of expertise has now shifted to software offerings within the IBM Big Data Platform and IBM Netezza®.

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

Gord Sissons, IBM Toronto, is the Product Marketing Manager for IBM Platform Symphony. He contributed the chapter on Platform Symphony. Gord has more than 20 years of experience in product management, distributed computing, Internet technologies and consulting services. Formerly Director of Technology at Sun Microsystems in Canada, and founder of NeatWorx Web Solutions Inc., he has held several senior roles in services and technology consulting throughout his career. Gord is a graduate of Carleton University in Ottawa, Ontario, with a degree in Systems and Computer Engineering.

For their valuable reviews of various chapters, thanks to:

Bob Loudon, IBM Raleigh
Jean-Francois J Rivard, IBM Atlanta
James Wang, IBM Poughkeepsie

Now you can become a published author, too!

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

length, and you can participate either in person or as a remote resident working from your home base.

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

ibm.com/redbooks

- ▶ Send your comments in an email to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

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

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- ▶ Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- ▶ Look for us on LinkedIn:

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

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

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

- ▶ Stay current on recent Redbooks publications with RSS Feeds:

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

Summary of changes

This section describes the technical changes made in this edition of the book and in previous editions. This edition might also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-8077-01
for Implementing IBM InfoSphere BigInsights on IBM System x
as created or updated on June 12, 2013.

June 2013, Second Edition

This revision reflects the addition, deletion, or modification of new or changed information described below.

New information

- ▶ Chapter 11 has been added, describing IBM Platform Symphony. IBM Platform Symphony is a new scheduling product that works with IBM Insights, bringing low-latency scheduling and multi-tenancy to IBM InfoSphere BigInsights.

The chapter explores the changing nature of distributed computing, and then explains how Platform Symphony operates and how InfoSphere BigInsights works with Platform Symphony. The performance benefit of Platform Symphony is also addressed. Sections highlighting the applications supported and the BigInsights versions supported are provided.



A whole new world of big data

As the planet becomes more integrated, the rate of data growth is increasing exponentially. This data explosion is rendering commonly accepted practices of data management to be inadequate. As a result, this growth has given birth to a new wave of business challenges around data management and analytics. Many people are using the term *big data* (sometimes referred to as *Big Data*) to describe this latest industry trend. To help you to understand it better, this chapter provides a foundational understanding of big data, what it is, and why you should care about it. In addition, it describes how IBM is poised to lead the next generation of technology to meet and conquer the data management challenges that it presents.

1.1 What is big data

For those individuals whose professions are 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 year or so. It is becoming increasingly popular to incorporate big data in data management discussions. In a similar way, it was previously popular to bring the advent of service-oriented architecture (more commonly known as SOA) and Web 2.0, just to give a few examples. The term big data is a trendy talking point at many companies, but few people understand what exactly is meant by it. Instead of 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 not only what big data is, but also why you should care.

Beginning in 2008 during a speech to the Council of Foreign Relations in New York, IBM began its Smarter Planet® initiative. *Smarter Planet* is focused on the development of leading-edge technologies that are aimed at 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 becoming increasingly difficult as the number and variety of sources continues to grow. The planet is exponentially more instrumented, intelligent, and integrated and it will only continue to expand with better and faster capabilities. The World Wide Web is truly living up to its name and through its continued expansion, the web is 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.

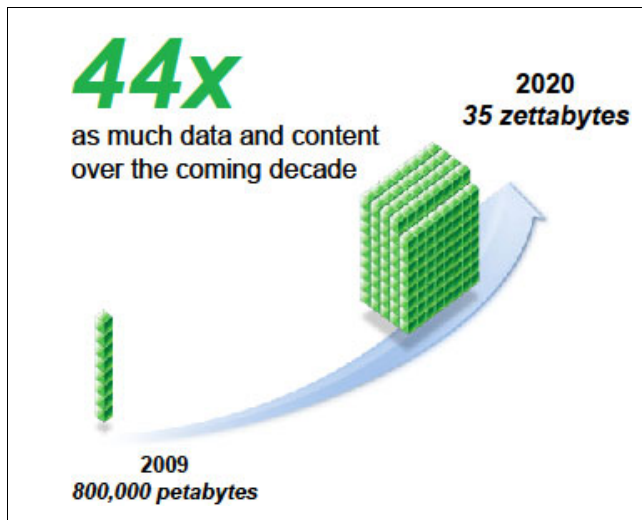


Figure 1-1 Predicted worldwide data growth

There was a point earlier in history, where 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 chatrooms, 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 or tablet while you are sending an 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 ten 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? In Figure 1-2, we show a few examples of the items and sources of this data explosion.

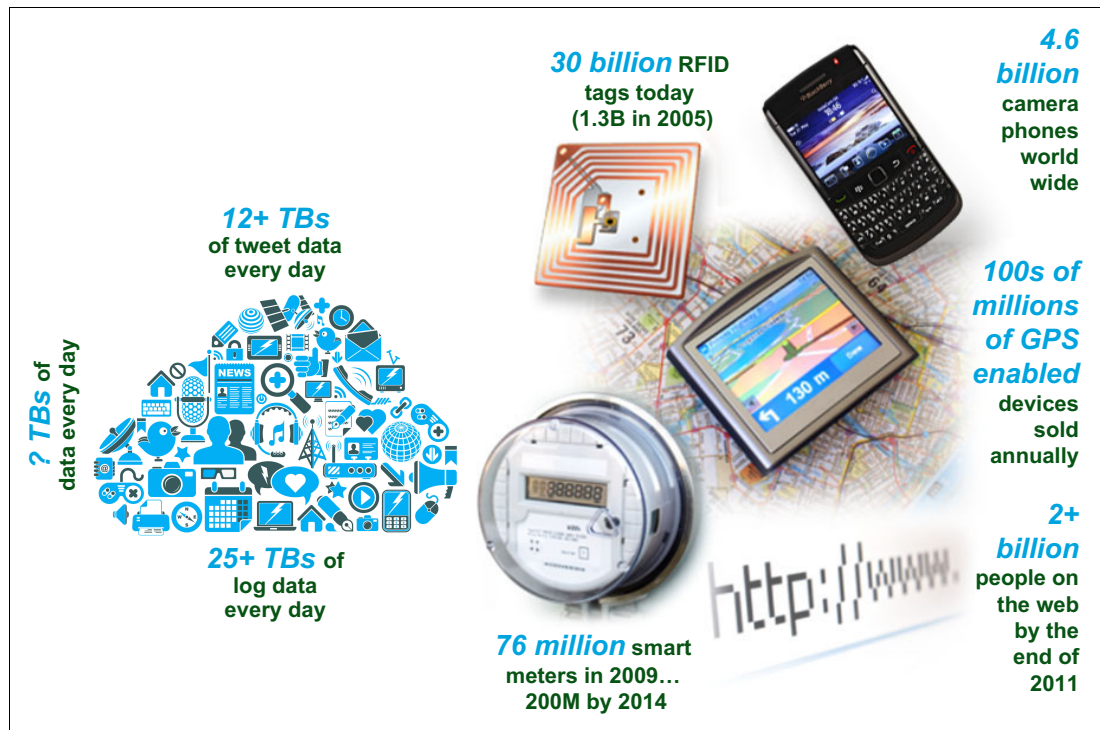


Figure 1-2 Big data explosion

Web-based applications, including social media sites, now exceed standard e-commerce websites in terms of user traffic. Facebook roughly produces 25+ TBs of log data on a daily basis. Twitter creates 12+ TBs of tweet data (made up mostly of text, despite the 140-character tweet limit), even more if there is a major global event (#IBMBigDataRedbook...are we trending yet?). Most everyone has an email address (often multiple), a smartphone (sometimes multiple as well), usually a cache of photo images and video (whether they choose to share with the social network or not), and can voice their opinion globally with their own 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*). Also, these software applications are all 24x 7x365 operational and continuously generating 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, you still must decipher the veracity of this new information as you do with structured data.

Here 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. To provide added insight into what is going on within your particular business arena, you must address the three Vs that define big data. A visual representation of the three Vs can be seen in Figure 1-3.

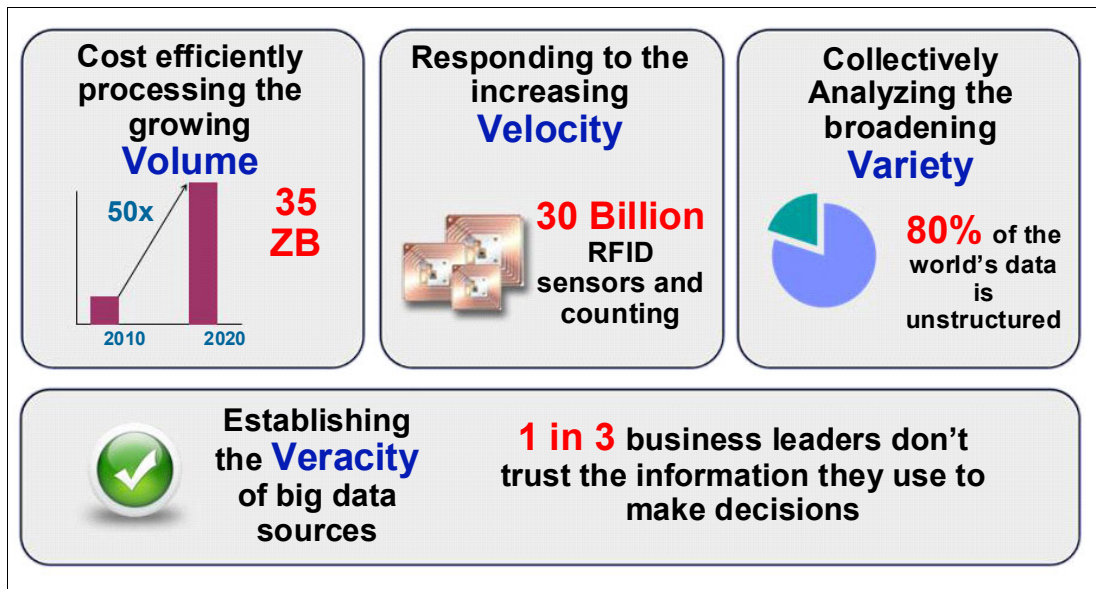


Figure 1-3 Three Vs of big data

Veracity: *Veracity* is not another official “V” for big data, but it holds true that the veracity (or *validity*, yet another V) of data is just as important to big data solutions as any prior data management solutions.

Figure 1-3 provides a picture of what big data is at its core. Harvesting and analyzing all of this data to provide competitive advantage is the challenge that faces businesses both today and moving forward into the future. How does an organization extract insight from the immense *Volume*, *Variety*, *Velocity*, and *Veracity* of data in a timely and cost-effective manner? This is the question and challenge that is posed by big data.

1.2 The big data challenge

Armed with an understanding of big data, we now explore the challenge that it presents to the data management world. In 1.1, “What is big data” on page 2, we posed the question, “How does an organization extract insight from the immense *Volume*, *Variety*, and *Velocity* of data in a timely and cost-effective manner?” The presence of big data means that we must harness its additional wealth of data and merge it with our existing data to perform meaningful analytics.

1.2.1 The traditional data warehouse in relation to big data

Some people might have the opinion that big data presents nothing new. They might say that it is already addressed by the *data warehouse (DW)*. Some might suggest that their DW works fine for the collection and analysis of structured data, and that their *Enterprise Content Management (ECM)* solution works well for their unstructured data needs. DW design is a mature practice in the data management arena and affords those who implemented a DW a

significant value by enabling deeper analytics of the stored data. We are not saying that traditional DWs do not have a role in the big data solution space. We are saying that DWs are now a foundational piece of a larger solution.

Typically, DWs are built on some enterprise-level *relational database management systems (RDBMSs)*. Regardless of the vendor, at their core these platforms are designed to store and query structured data. This approach was solid until the desire to do the same thing with unstructured data began to rise. As the need for this functionality became more prevalent, many vendors began to include unstructured data storage and query capabilities in their RDBMS offerings. The most recent example is the ability to handle XML data. Although IBM might believe that they did it better than anyone else in the market, IBM was no different as it introduced the basic XML data type in its 2006 release of DB2 V9.1. Furthering this capability to enforce structure on unstructured data, text search and analysis tools were developed to enable the extraction and reformatting of data. This data was able to then be loaded into the structured DW for query and analysis.

In 2012, we saw a high velocity data source, such as streaming video or sensor data, continuously sending data 24x7x365. Specifically, we assume that the central supply warehouse of a company does not have on-site security. Instead, they might choose to use several high definition (HD) video cameras for monitoring key locations at the facility. We also assume that the cameras are streaming this data to another location for monitoring and storage in a DW where data for the company's day-to-day transactions is also stored and analyzed.

The person in charge of overnight monitoring of this video is not necessarily a security professional. This person might be a college student, working to earn extra money and is working on their project deadline instead of being intensely focused on the security camera monitor. If someone breaches the warehouse and makes off with valuable company assets, there is a possibility that the security staff might miss the opportunity to alert the appropriate authorities in time to take appropriate action. Because that data is captured in the DW for later analysis, the assets of the company are already compromised and there is the strong possibility that they might be unable to recover them. In instances where you have real-time events that take place and a need to process data as it arrives, a DW lacks the capability to provide much value.

In-memory solutions: There are in-memory solutions that are aimed at faster analysis and processing of large data sets. However, these solutions still have the limitation that data must be primarily structured. Thus, in-memory solutions are subject to the same pitfalls as traditional DWs as it pertains to management of big data.

Big data can be subcategorized as *data at rest* and *data in motion*. The following section (and this book in general) addresses data at rest.

1.2.2 How continual data growth affects data warehouse storage

Big data is not just about the sheer volume of data that is available. However, data volume is a key factor in the architecture of DW and analytics-intensive solutions. When discussing DW architecture, the user service level agreements (SLAs) are key in constructing an efficient data model, schema, hardware, tuning of the database, and so on. Because we are describing DWs, we can assume that we are working with 10s to 100s of TBs (and in many cases, petabytes). This data must be located somewhere and is typically placed on a storage array of *network-attached storage (NAS)*.

A common performance bottleneck in DW environments is the *I/O* that is required for reading massive amounts of data from storage for processing within the *DW* database server. The server ability to process this data is usually a non-factor because they typically have significant amounts of RAM and processor power, parallelizing tasks across the computing resources of the servers. Many vendors have developed *DW* appliances and appliance-like platforms (which we call *DW platforms*) that are designed for the analytics intensive workload of large *DWs*. *IBM Netezza* and *Smart Analytics Systems* are examples of these types of platforms.

Imagine that traditional *DW* environments are able to capture and analyze all of the necessary data instead of operating under the 80/20 principle.

80/20 principle: The *80/20 principle* is a willingness to analyze only 20% of all data and disregard the remaining 80% for no reason other than its format does not fit the incumbent model for data analysis.

Because these *DW* platforms are optimized for analytics intensive workloads, they are highly specialized systems and are not cheap. At the rate that data continues to grow, it is feasible to speculate that many organizations will need *petabyte (PB)* scale *DW* systems in the next 2 - 5 years. Continuing with the security example from 1.2.1, “The traditional data warehouse in relation to big data” on page 4, HD video generates about 1 GB of data per minute of video, which translates to 1.5 TB of data generated daily per camera. If we assume that five cameras are in use, that is roughly 7.5 TB per day that is being generated which extrapolates to the following data amounts:

- ▶ 52.5 TB a week
- ▶ 210 TB a month
- ▶ 2.52 PB a year

This amount is over 2 PB annually of more data coming into a warehouse that is completely separate from typical day-to-day, business-centric data systems for which you might already be capturing and performing some form of analytics. This is assuming that your data warehouse has that level of storage available (which is a *big* assumption).

Perhaps instead of 52.5 TB a week of additional data, you can realistically see 5 TB a week being captured and incorporated into your normal data analytics business processes. That still adds up to 20 TB/month of extra data that you did not account for within your enterprise data warehouse architecture. That is 20 TB for each month that you have to plan for in terms of improving your *DW* data model to ensure user SLAs are still met, more storage, and potentially more hardware that you have to purchase. You also have to consider added power that is needed in your data center and the need to potentially hire more personnel for *DW* administration, and so on.

As you can see, the costs of capturing and analyzing data swell quickly. Instead of incurring the added cost to collect and analyze all this additional data, what if you could use commodity hardware as a foundation for storing data? What if you could use the resources of this hardware to filter data, and use your existing *DW* to process the remaining data that is determined to hold business value? That might be significantly more cost effective than expanding your *DW* platform to a size large enough to perform analytics on all of the data.

Video: Typically, *DWs* are not able to use video because video is not considered structured data and does not lend itself to any data types that are common to *relational database management system (RDBMS)* technology. In this example, you would more than likely have to purchase a separate video storage solution versus being able to add to your existing *DW*.

1.3 How IBM is answering the big data challenge

In answering this new challenge, the IBM approach has been multi-faceted. IBM is incorporating new capabilities into existing infrastructure to enable the enterprise to efficiently store and analyze virtually any variety, volume, or velocity of data. As mentioned consistently throughout this chapter, this added functionality and additional tools enhance current DW investments rather than replace them. What exactly does this mean? Contrary to what some might have you believe, no single infrastructure can solve all big data problems. Rather than create only a single product, IBM assessed where the addition of new technology can complement the existing DW architecture and thus provide added value to the enterprise. At IBM, we answer the challenge with our big data platform.

1.3.1 Big data platform

In Figure 1-4, notice how the big data platform is not just one product recommendation that is aimed at replacing your current DW infrastructure. In fact, the Data Warehouse is specified as a foundational component of the overall architecture.

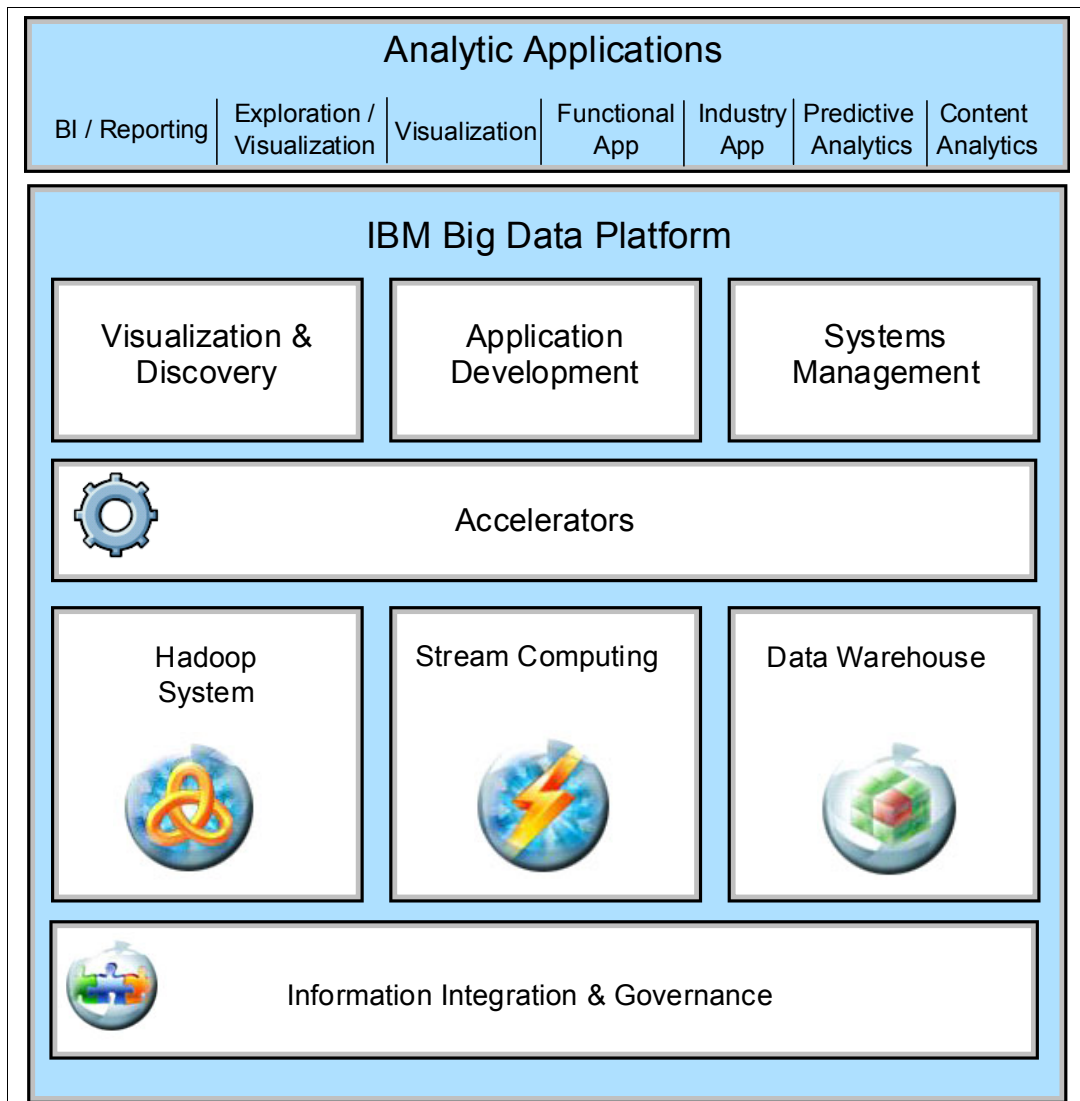


Figure 1-4 IBM big data platform

At the foundation of the platform, which is shown toward the bottom of Figure 1-4, is *Information Integration and Governance*. A key facet of any data management solution, this foundation should not change in the big data realm. This layer encompasses core capabilities of any trusted data environment, enabling organizations to develop an IBM Information Agenda® to understand, cleanse, transform, and deliver trusted information to the enterprise.

In Figure 1-5, we show more capabilities that you might consider when you take a platform approach to big data.



Figure 1-5 Other platform capabilities

These components help to promote the efficiency of several key areas within a data management ecosystem:

► **Visualization and Discovery**

Visualization makes data more digestible and easier to understand, and helps users to discover previously unrecognized trends and data relationships. The IBM Velocity platform engine provides these capabilities which enables data discovery, understanding, and navigation of federated big data sources while leaving the data in place and intact.

► **Application Development**

Common development environments promote collaboration and simplify problem triage during quality assurance (QA) testing in the event of a failure. IBM includes support and tooling for the open source JSON query language to help organizations standardize on a platform and accelerate the development of applications that can use Hadoop's distributed architecture.

► **Systems Management**

DW-based platforms store and manage very large volumes of data. They serve as a key piece of the day-to-day operations, helping decision makers steer the enterprise in a positive direction. It is important to have tools to enable administrators to manage these systems and ensure that they are working correctly and performing to agreed-upon SLAs.

Ultimately, all of these things transform and feed data into user-based analytic applications.

Attention: Included in the big data platform are accelerators that are built into the BigInsights product to speed up data processing for specific applications, industries, and business processes. These accelerators are mentioned in later chapters but are not covered extensively within this book.

1.3.2 Big data Enterprise Engines

Big data is sometimes divided into *data at rest* and *data in motion*. BigInsights analyzes data at rest. InfoSphere Streams analyze data in motion. That is, IBM incorporates big data in the Hadoop System and Stream Computing components of the platform, which we refer to as our big data Enterprise Engines as shown in Figure 1-6.

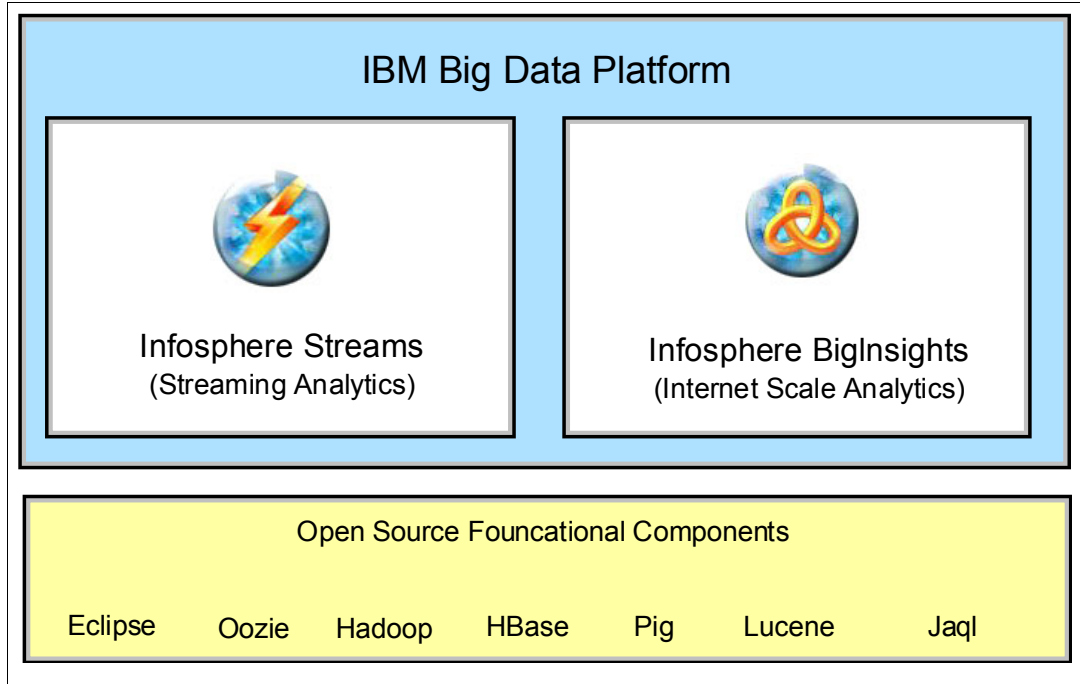


Figure 1-6 IBM big data enterprise engines

To provide the capability of incorporating big data sources and analytics of these sources, IBM developed a stream-computing product and used the open source computing framework that is known as *Apache Hadoop*. Each product provides unique capabilities to the data management environment to further enhance the value of your DW investment.

IBM InfoSphere BigInsights

IBM incorporated Apache Hadoop into its big data platform. For individuals who are unfamiliar with this architecture, Hadoop is a software framework that is typically implemented on a cluster of commodity-based hardware servers to perform distributed computational operations across the hardware in the cluster. Unlike traditional DWs, Hadoop does not require a physical data model and schema to be defined before ingesting data into the Hadoop cluster. Therefore, it is able to store virtually any data format within its file system, known as *Hadoop Distributed File System (HDFS)*. To make this a feasible option for the enterprise, IBM developed a product called *InfoSphere BigInsights*.

This offering provides a packaged Hadoop distribution, a greatly simplified installation of Hadoop and corresponding open source tools for application development, data movement, and cluster management. BigInsights also provides more options for data security which is frequently a point of concern for anyone contemplating the incorporation of new technology into their data management environment. BigInsights is a component of the IBM big data platform and as such, provides potential integration points with the other components of the platform including the DW, data integration and governance engines, and third-party data analytics tools. The stack includes tools for built-in analytics of text, natural language processing, and spreadsheet-like data discovery and exploration.

IBM InfoSphere Streams

The second engine is a streams-computing engine called *InfoSphere Streams* (or *Streams* for short). InfoSphere Streams can analyze continuously streaming data before it lands inside the DW. In addition to volume, our definition of big data includes the *velocity* of data as well. Streams computing is ideal for high-velocity data where the ability to recognize and react to events in real time is a necessary capability.

Although there are other applications aimed at providing stream-computing capability, the Streams architecture takes a fundamentally different approach to continuous processing, differentiating it from other platforms. Its distributed runtime platform, programming model, and tools for developing continuous processing applications promote flexibility, development for reuse, and unparalleled performance. A picture of these areas that is provided by Streams can be seen in Figure 1-7.

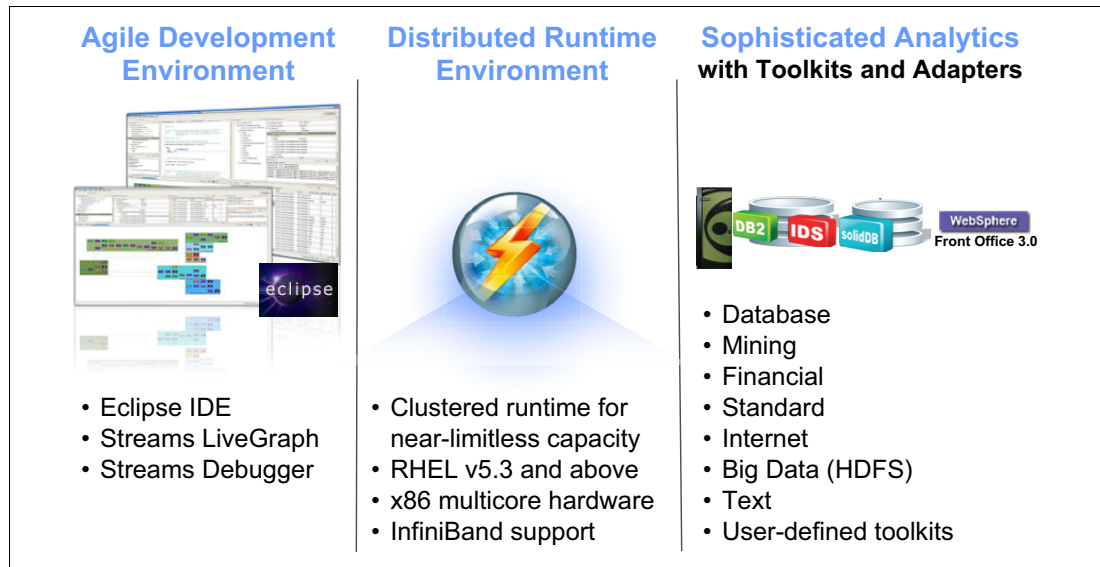


Figure 1-7 InfoSphere Streams capabilities and components

When stream processing is described, it is typically associated with *complex event processing (CEP)*. However, *Streams* and *CEP* are truly different. Aside from fact that both operate on *real-time* data, have ultra-low latency, and provide *event-based*, stream processing, InfoSphere Streams potentially outperforms CEP in other aspects.

CEP provides analysis on discrete business events, is rule-based with correlation only across certain event types, has modest data rates, and operates only on structured data. Alternatively, InfoSphere Streams provides simple and complex analytics on continuous data streams, is able to scale for computational complexity, and supports a wide range of relational and non-relational data types. When discussing similarity to *CEP*, *InfoSphere Streams* supports higher data rates and a much broader range of data types. For example, there are data sources consumable by Streams including but not limited to sensors, cameras, video, audio, sonar or radar inputs, news feeds, stock tickers, and relational databases.

1.4 Why you should care

Although the data explosion that is described presents a new challenge, it also presents a great opportunity to capture and extract insights from this ocean of information.

Social networks are the most recognizable and a sparkling example of what big data is all about. These networks serve as a consolidation of various data formats being generated and enhanced at various speeds to provide the user with significant insight on a particular individual, company, hobby, and so on. There is an exponentially increasing quantity of applications that are connecting users. These applications are generating more data to be analyzed, understood, and used for the enhancement of the user's experience. The underlying technology which enables this marriage of massive volumes and variety of data can potentially be used within your organization's enterprise data management environment as an additional data source for potential competitive advantage.

Ashish Thusoo, the former Head of Big Data at Facebook, recently shared some insights in Forbes magazine around the implications of developing and using applications that can take advantage of big data. One strong yet intuitive point that he mentioned, was that as modern technology becomes more cost effective, it shifts the conversation from *what data to store* to *what can we do with more data*.

Thusoo's point is ultimately what we are getting at in this chapter. We know that there is more data, we know that there are now various formats it can be in, and that this *new* data is being generated at faster rates than before, but what can we do with it? The answer most people want to be able to say is store it, query it, and perform analytics on it to yield a better understanding which leads to improvement in whatever key performance indicators are important to their business. A key driver for corporations storing data, architecting DWs, and so on, is to be able to query and perform analytics on the data. This function is used to not only understand their clients, but to also be better, faster, and smarter than their competitors by including big data within their data centers. Smart people use complex algorithms to understand, model, and predict behavior that is based on data. The general questions that the enterprise wants to have answered have changed little over time. *Who, what, where, when, why, and how much*, are still the basic questions. DW solutions enable the enterprise to answer those questions with an acceptable degree of certainty. Through the incorporation of big data, the enterprise is potentially able to answer these questions to a higher level of confidence than ever before.



Why choose BigInsights

Knowing that the *big data* challenge is very real, Chapter 2 describes how the *Hadoop* architecture lends itself to addressing these new challenges in data management. Additionally, we explore more about what is included in *IBM BigInsights*. Also explained is why you should consider selecting this offering to construct your Hadoop implementation versus other distributions or doing a piece-by-piece installation of individual open source components.

2.1 BigInsights introduction

BigInsights is the IBM product that is built on top of *Apache Hadoop*, which is designed to make distributed processing accessible to all users. This product helps enterprises manipulate massive amounts of data by optionally mining that data for insights in an efficient, optimized, and scalable way.

2.2 What is Hadoop?

Fundamentally, *Hadoop* is two components: a *Hadoop Distributed File System (HDFS)*, and *MapReduce*. HDFS provides a way to store data and MapReduce is a way of processing data in a distributed manner. These components were developed by the open source community that are based on documents that were published by Google in an attempt to overcome the problems that are faced in trying to deal with an overwhelming volume of data. Google published papers on their approach to resolve these issues and then Yahoo started work 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 system 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 datablocks 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 an "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. The JobTracker will also, where possible, ensure 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 2-1 shows an access plan for a MapReduce that was submitted by a client.

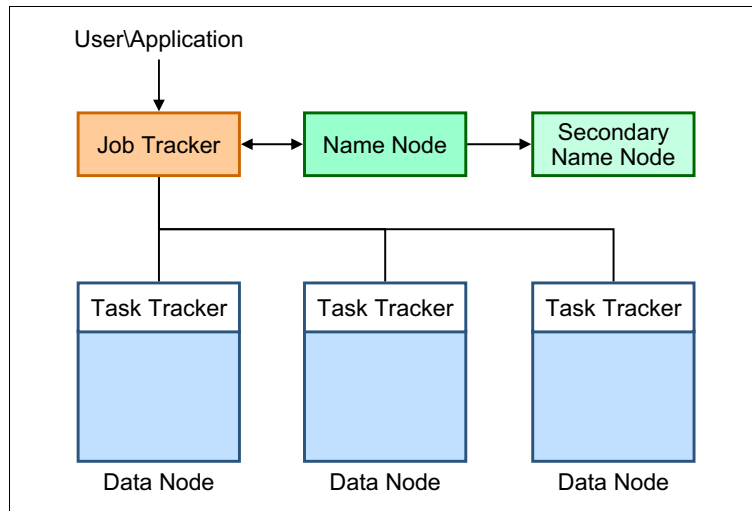


Figure 2-1 Access plan for a MapReduce submitted by a client

2.2.1 Hadoop Distributed File System in more detail

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

When a file is saved in HDFS, it is first broken down into blocks with any remainder data that is occupying the final block. The size of the block depends on the way that HDFS is configured. At the time of writing, the default block size for Hadoop is 64 megabytes (MB). To improve performance for larger files, BigInsights changes this setting at the time of installation to 128 MB per block. Then, each block is sent to a different data node and written to the hard disk drive (HDD). When the data node writes the file to disk, it then 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 writeback to the second, 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 down 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 2-2 shows an example of HDFS.

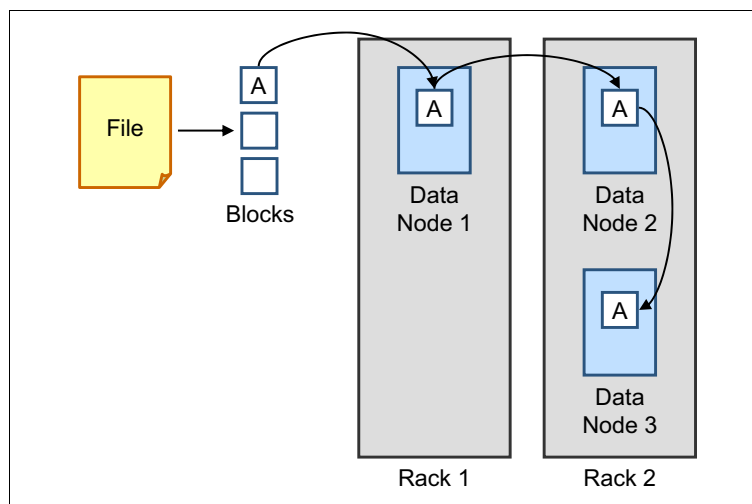


Figure 2-2 A visual example of HDFS

Default settings and behavior

By default, BigInsights sets the block size to *128 MB* and the replication factor to 3. The location of each data node is recognized by Hadoop and nodes are grouped by rack. Hadoop can be instructed as to which data nodes are in the same rack and recognizes that nodes on the same network switch are in the same rack by default. When a file is copied from the first data node to the second, Hadoop makes the second copy on a second rack if it can. The third copy is saved to a different node, also on the second rack.

Scalability

Hadoop is designed to run on many commodity servers. The Hadoop software architecture also lends itself to be scalable within each server. HDFS can deal with individual files that are terabytes in size and Hadoop clusters can be petabytes in size if required. Individual nodes can be added to Hadoop at any time. The only cost to the system is the input/output (I/O) of redistributing the data across all of the available nodes, which ultimately might speed up access. The upper limit of how large you can make your cluster is likely to depend on the hardware that you have assembled. For example, the NameNode stores metadata in random access memory (RAM) that is roughly equivalent to a GB for every TB of data in the cluster.

Failover

The *TaskTracker* on each data node stays in contact with *JobTracker* with a heartbeat signal that is sent across the network periodically. This signal also functions as an indicator of how busy that data node is so that the JobTracker does not over-subscribe one node, while it neglects others. If a data node fails while performing a job, the JobTracker notices when the TaskTracker fails to send a heartbeat. After a few cycles of no heartbeat, the JobTracker reassigns the job to a different node. After a specified maximum number of failures on data nodes, the JobTracker cancels the job on the Management Node.

Unstructured data

Hadoop breaks down the files into blocks and does not try to assign the data to a schema or enforce any type of structure. Therefore, the files that can be stored can take almost any format, including unstructured or semi-structured. The only formats that might cause an issue are certain forms of compression, as described in 4.3.3, “Compression” on page 51.

2.2.2 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 multi-node cluster is what allows this level of data processing to take place.

MapReduce applications can process vast amounts (multiple terabytes) of data 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 re-executes 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 is composed of the following phases:

- ▶ Map
- ▶ Reduce

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).

Important: Hadoop is a great engine for batch processing. However, if the data volume is small, the processor usage that is incurred by using the MapReduce framework might negate the benefits of using this approach.

Based on the data set that one is 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 can be (and often are) different from each other.

The reduce phase

As with the map function, developers also must create a *reduce* function. The key/value pairs from map outputs must correspond to the appropriate reducer partition such 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 together to maintain their sort ordering that is established during the map phase. When the final merge is complete (because this process is done in rounds for performance optimization purposes), 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.

Development languages: *Java* is a common language that is used to develop these functions. However, there is support for a host of other development languages and frameworks, which include Ruby, Python, and C++.

2.3 What is BigInsights?

BigInsights is a software platform that is developed by IBM that uses the Hadoop architecture that is combined with IBM-developed technologies to produce an enterprise-ready software solution. Details that are provided in the following list are particular to *BigInsights Version 1.4*, the latest version available at the time of writing.

IBM InfoSphere BigInsights Enterprise Edition contains the following components:

- ▶ The IBM Distribution of Apache Hadoop. Contains Apache Hadoop (1.0.0), a 64-bit Linux version of the IBM SDK for Java 6
- ▶ IBM InfoSphere BigInsights Jaql. A language that is designed for JavaScript Object Notation (JSON), is primarily used to analyze large-scale, semi-structured data
- ▶ IBM InfoSphere BigInsights Jaqlserver. A Jaql UDF access gateway to Jaql
- ▶ BigInsights console. Provides central management of the cluster from a web-based interface to maintain the servers in the cluster to manage jobs, and to browse the distributed file system
- ▶ InfoSphere BigInsights workflow scheduler. Ensures that all jobs get an appropriate share of resources over time
- ▶ BigSheets. A browser-based analytics tool to extend the scope of your business intelligence data. With BigSheets, you can easily view and interact with massive amounts of data into consumable, situation-specific business contexts.
- ▶ Avro (1.5.1). A data serialization system
- ▶ Derby (10.5.3.1). A Java relational database management system that can be embedded in Java programs and used for online transaction processing
- ▶ Flume (0.9.4). A distributed, reliable, and highly available service for efficiently moving large amounts of data around a cluster
- ▶ HBase (0.90.5). A non-relational distributed database that is written in Java
- ▶ Hive (0.8.0). A data warehouse infrastructure that facilitates both data extraction, transformation, and loading (ETL), and the analysis of large data sets that are stored in HDFS
- ▶ Lucene (3.3.0). A high-performance, full-featured text search engine library that is written entirely in Java
- ▶ Oozie (2.3.1). A workflow management system to coordinate Apache Hadoop Jobs
- ▶ Pig (0.9.1). A platform for analyzing large data sets, consists of a high-level language for expressing data analysis programs and an infrastructure for evaluating those programs
- ▶ ZooKeeper (3.3.4). A centralized service for maintaining configuration information, providing distributed synchronization, and providing group services

Some of these products are open source projects that were tested by IBM to ensure that these versions work together seamlessly. Some, like BigSheets, are extra-value components that IBM has added on top of what is available from open sources.

2.3.1 All-in-one installation

The first value added in a BigInsights deployment is the benefit of an all-in-one installation that uses a *graphical user interface (GUI)*. The included components were tested with each other and integrated with other IBM-provided BigInsights components. This configuration offers the following benefits:

- ▶ Simple installation as demonstrated in Chapter 6, “BigInsights installation” on page 73. No special skills are required.
- ▶ Fast installation with a built-in, installation health checker requires only a short amount of time to perform and initially verify the installation.
- ▶ Pretested components provide versions that were tested together and can work seamlessly together.
- ▶ A developer-focused, single-node installation option through a GUI. This option is useful for developers if they do not need to change or customize settings during a multi-node installation process.
- ▶ A multiple-node installation option, either through a GUI or a command-line based, silent installation process for easily installing the components across multiple nodes and even multiple racks of hardware.

2.3.2 Integration with existing information architectures

To extract the most value possible from a Hadoop-based system, integrate it with existing systems. BigInsights contains high-speed connectors for the IBM Netezza family of data warehouse appliances, IBM DB2, IBM InfoSphere Warehouse, and IBM Smart Analytics Systems. These high speed connectors help to simplify and accelerate data manipulation tasks.

BigInsights also contains a tool for connecting to IBM InfoSphere DataStage for ETL jobs and a standard Java Database Connectivity (JDBC) connector for connecting to a wide variety of data and information systems. Examples of applicable systems include: Oracle, Microsoft SQL Server, MySQL, and Teradata.

2.3.3 Enterprise class support

Open source software comes with no guarantees or support if something goes wrong, or even if assistance is needed. However, *IBM InfoSphere BigInsights Enterprise Edition* is delivered with standard licensing and support agreements. This support means that businesses can deploy this console, and at the same time, potentially minimize uncertainty and risk. Clients can be confident that if anything goes wrong they can call upon 24-hour support and a worldwide professional services organization.

If you choose to get started with the *BigInsights Basic Edition*, which is a download that is available at no charge, you can manage up to 10 TB of data and receive online support at no charge. If you choose to stay with the Basic Edition and want to add telephone support, this option is available at the time of writing this Redbooks publication. Software and online support that are available at no charge from IBM is a great way to get started.

2.3.4 Enterprise class functionality

BigInsights was designed with business needs in mind. For this reason, in addition to the built-in redundancy that is supplied by Hadoop, there are hardware options to ensure continuity.

BigInsights also has a web console that allows easy submission of MapReduce jobs. It also allows you to view submitted jobs, facilitate workload management, and provide options for system administration. The automated job status area shows, in real time, how jobs are progressing. The console also allows you to view HDFS content and provides role-specific views. Depending on the user’s access-group assignments, the console can be used to start

and stop services within the cluster, monitor jobs that are running within the cluster, deploy applications for users to be able to run, and provide access to the BigSheets, spreadsheet-style interface for users to interact with the data stored in Hadoop.

Figure 2-3 shows an example of the BigInsights web console.

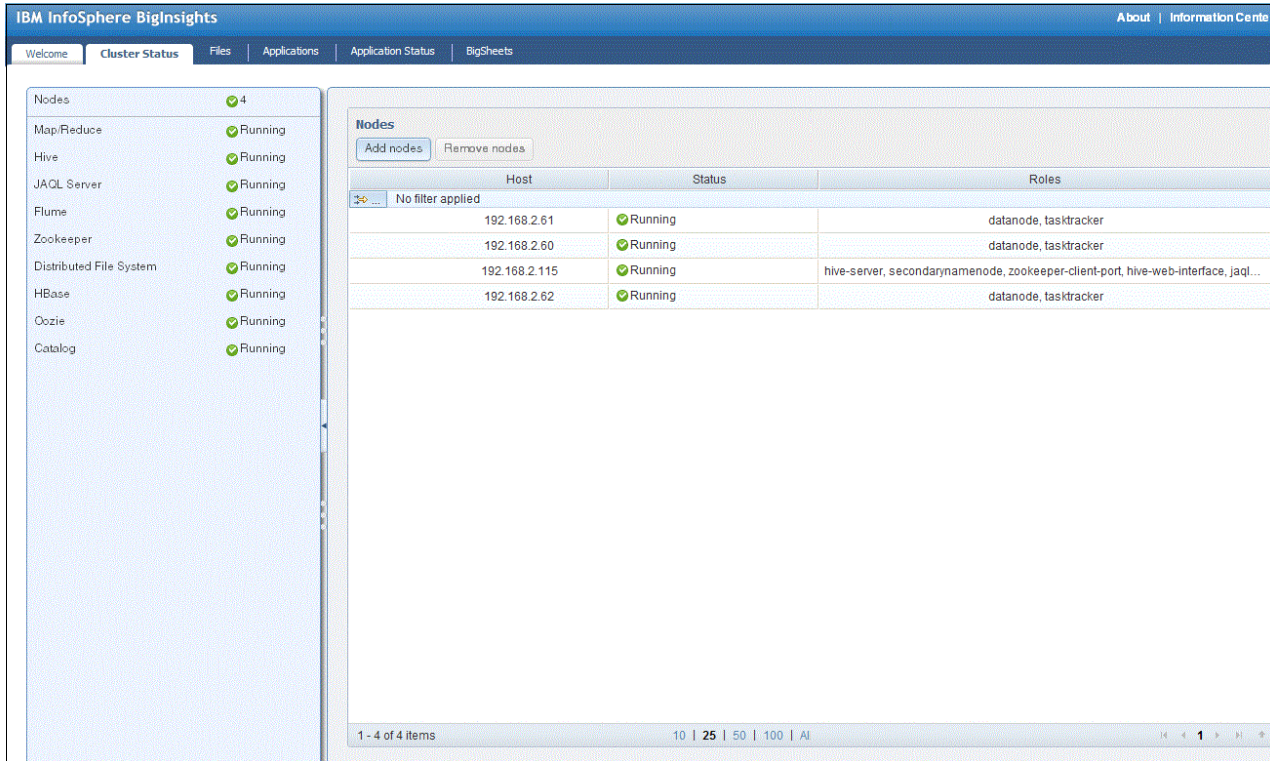


Figure 2-3 BigInsights web console

2.3.5 BigSheets

BigSheets is a browser-based analytic tool that was initially developed by the IBM Emerging Technologies group. Today, BigSheets is included with the BigInsights Enterprise Edition to enable business users and non-programmers to explore and analyze data in distributed file systems. BigSheets presents a spreadsheet-like interface so that users can model, filter, combine, explore, and chart data that is collected from various sources. The BigInsights web console includes a tab to access BigSheets.

Typically, users filter, explore, and enrich the contents of their online spreadsheets, called *collections*, by using built-in functions and macros. Furthermore, some users combine data that is in different collections, creating new sheets (collections), and rendering charts to visualize their data. Finally, users can export the results of their BigSheets processing into various common formats for use by downstream applications. BigSheets provides export facilities for collections to be saved into HTML, JSON, CSV, RSS, and Atom data formats.

2.3.6 BigInsights scheduler

The *InfoSphere BigInsights Scheduler* provides a flexible workflow allocation scheme for MapReduce jobs.

The InfoSphere BigInsights Scheduler is the default scheduler of InfoSphere BigInsights. The default metric for the scheduler is based on average response time metrics, although the *Hadoop Fair* scheduler console remains operational with the InfoSphere BigInsights scheduler. The Hadoop Fair scheduler guarantees that all jobs get an equitable share of cluster resources over time. However, using the InfoSphere BigInsights scheduler with the average response time metric, allocates maximum resources to small jobs, guaranteeing that these jobs are completed quickly.

Specifying the scheduler, the metric (algorithm), and configuring priority classes are administrative tasks and they are applicable across the entire cluster. To change the scheduler or metric, the JobTracker must be restarted. To enable the InfoSphere BigInsights scheduler, add or change the property in the `mapred-site.xml` file, as shown in Example 2-1.

Example 2-1 Setting the taskScheduler in the mapred-site.xml file

```
...
<property>
  <name>jobtracker.taskScheduler</name>
  <value>com.ibm.biginsights.scheduler.WorkflowScheduler</value>
</property>
...
```

The InfoSphere BigInsights scheduler also supports the Fair metric and *Max Stretch* metric. When you use the Fair metric, the scheduler mimics the behavior of the Hadoop Fair scheduler. The maximum stretch metric allocates resources in proportion to the amount of work that the jobs require. The scheduling metric, and the corresponding scheduling algorithm, is controlled by setting the property in the `mapred-site.xml` file, as shown in Example 2-2.

Example 2-2 Setting the scheduler algorithm in the mapred-site.xml file

```
...
<property>
  <name>mapred.workflowscheduler.algorithm</name>
  <value>AVERAGE_RESPONSE_TIME</value>
<!-- Possible values are:
  <value>AVERAGE_RESPONSE_TIME</value>
  <value>MAXIMUM_STRETCH</value>
  <value>FAIR</value>
  The default is AVERAGE_RESPONSE_TIME
-->
</property>
...
```

The InfoSphere BigInsights scheduler allocates resources to jobs according to job priority. You can specify priorities either per job or per Jaql query by setting the *flex.priority* property in *Hadoop JobConf*. The value of this property is the number of the priority class that the jobs belong to. *Priority class 0* always takes precedence over the other classes. By default, there are three priority classes, so *flex.priority* can be specified as values 0, 1, or 2. If *flex.priority* is not specified, the job runs at the default priority. In the default configuration, the default priority class is 2.

For example, in the following Jaql query, the `SetOptions()` function call sets *flex.priority* to 1 for all the Hadoop jobs in the query:

```
setOptions( { conf: { "flex.priority": 1 } } );
```

The priority can also be specified from the command line that starts a Hadoop job. See the following example:

```
$hadoop jar $BIGINSIGHTS_HOME/IHC/hadoop*examples.jar -Dflex.priority=1 <input>  
<output>
```

By default, the scheduler uses the three priority classes that are shown in Example 2-3. Their order is shown from the highest to lowest priority.

Example 2-3 Scheduler priority classes settings in the mapred-site.xml file

```
...  
<priorityClasses>  
  <class name="Urgent" share="0.3" />  
  <class name="Production" share="0.3" />  
  <class name="Normal" share="0" default=true" />  
</priorityClasses>
```

The administrator can change the priority classes by editing the configuration file, specified by the following property to the mapred-site.xml file:

```
<property>  
  <name>mapred.workflow.scheduler.config.file</name>  
  <value>conf/flex-scheduler.xml</value>  
</property>If the configuration file is not specified or cannot be read or parsed, the  
default classes are used in the configuration. The configuration file has the following  
structure:  
<priorityClasses>  
  <class name="Urgent" share="0.3" />    <!--this is priority class # 0  
  -->  
  <class name="Production" share="0.2" />  
  <class name="Adhoc" share="0.2" default="true" />  
  <class name="Low" share="0" />  
</priorityClasses>  
...
```

The order of priority classes in the configuration file is important because it implies the ordinal number of the class. The first class in the list is always priority 0. So, when the example specifies *flex.priority=1*, it essentially is requesting to run with **Production** priority.

The **share** attribute specifies the minimum share of all cluster slots that are guaranteed to this priority class. For example, even if many *Urgent* tasks are waiting in the system, 20% of the cluster's slots are still given to **ad hoc** priority tasks. If a priority class lacks runnable tasks to fill the minimum share of the slots, the unneeded slots are given to the highest priority class in need. In addition, the slack slots (slots outside the minimum allocations of the classes) are also given to the highest class that needs slots. For example, in the previous configuration file, the total of all minimum allocations is 0.7. Thus, the slack is 0.3 of all slots.

The number of slots in the Hadoop cluster is the total of slots at every node in the cluster. The slots for a specific node are configured in the mapred-site.xml file, as shown in Example 2-4.

Example 2-4 Setting the maximum map task in the mapred-site.xml file

```
...  
<property>  
  <name>mapred.tasktracker.map.tasks.maximum</name>  
  <value>2</value>  
  <final>true</final>  
</property>  
...
```

2.3.7 Text analytics

The *text analytics* component extracts information from unstructured and semi-structured data. By using this component, enterprises can analyze large volumes of text and produce annotated documents that potentially provide valuable insights into unconventionally stored data.

The *text analytics* component includes the following features:

- ▶ A declarative language, Annotation Query Language (AQL), with familiar SQL-like syntax for specifying extraction programs (or extractors) with rich, clean rule-semantics. For more information about AQL, see this website:

http://pic.dhe.ibm.com/infocenter/bigins/v1r4/topic/com.ibm.swg.im.infosphere.biginsights.text.doc/doc/biginsights_aqlref_con_aql-overview.html

- ▶ An optimizer that generates an efficient execution plan for the extractor specification. For more information about the optimizer, see the following website:

http://pic.dhe.ibm.com/infocenter/bigins/v1r4/topic/com.ibm.swg.im.infosphere.biginsights.text.doc/doc/biginsights_aqlref_con_execution-model.html

- ▶ A runtime engine for issuing extractors in a highly efficient manner by using the parallelism that is provided by the IBM InfoSphere BigInsights engine. For more information about the runtime engine, see this website:

http://pic.dhe.ibm.com/infocenter/bigins/v1r4/topic/com.ibm.swg.im.infosphere.biginsights.text.doc/doc/biginsights_aqlref_con_execution-model.html

- ▶ For more information about running extractors on the cluster, see this website:

<http://pic.dhe.ibm.com/infocenter/bigins/v1r4/topic/com.ibm.swg.im.infosphere.biginsights.analyze.doc/doc/r0057884.html>

- ▶ Built in multilingual support for tokenization and part-of-speech analysis. For more information about multilingual support, see this website:

http://pic.dhe.ibm.com/infocenter/bigins/v1r4/topic/com.ibm.swg.im.infosphere.biginsights.text.doc/doc/text_analytics_languageaware_support.html

- ▶ A rich library of precompiled extractors for high-interest types of entities and relationships. For more information about these extractors, see this website:

http://pic.dhe.ibm.com/infocenter/bigins/v1r4/topic/com.ibm.swg.im.infosphere.biginsights.analyze.doc/doc/pre-compiled_extractor_libraries.html

- ▶ An Eclipse-based development environment for building and maintaining text analytics extractors.

- ▶ The Eclipse tools are a separate set of tools which must be installed separately. For more information about the Eclipse tools, see this website:

http://pic.dhe.ibm.com/infocenter/bigins/v1r4/topic/com.ibm.swg.im.infosphere.biginsights.analyze.doc/doc/developing_text_analytics_extractors.html

2.4 BigInsights and the traditional data warehouse

With its ability to process structured data and the inclusion of a data warehouse component, many people might look upon Hadoop and BigInsights as a direct competitor to the traditional data warehouse (DW). However, this analogy is not meant to be the case. BigInsights is designed to *complement* the data warehouse by providing unparalleled access to unstructured data while also providing a queryable archive for structured and semi-structured data.

2.4.1 How can BigInsights complement my data warehouse?

There are many different ways that BigInsights might complement your data warehouse. Here are just a few for your consideration.

Structure

To insert data into a DW, the data must first be structured in accordance with the data model. Many data sources like blogs, documents, or click streams do not usually have a fixed structure. Therefore, some transformation is typically performed on them as a first step to extract value. Then, this potentially valuable, structured information can be loaded into a DW for the user of a business insight tool.

Size

BigInsights can run on many commodity servers which provides a potentially less expensive way to use huge volumes of storage. This configuration leaves BigInsights perfectly placed to act as both an archive of “cold” data (data that is no longer needed by the DW) and as a holding area for raw data. This raw data refers to data that has not yet been brought in line with the data standards that are required. Archiving in BigInsights has the advantage over archiving to tape because the data can be more rapidly accessed. Additionally, hard disk drives do not require the maintenance or constant upgrades that tape drives usually require. Keeping the raw data in BigInsights as an archive saves the power of the data warehouse for use on business-critical data and other highly structured information that is used for business insight and reporting.

Access

BigInsights is not an ETL tool but it can act as a platform to use ETL tools in parallel over many nodes. This parallelism has the advantage of being faster, but also prevents a buildup of network traffic. This benefit is because, by using MapReduce, BigInsights directs the processing power to the data instead of transferring large volumes of data around the network before any transformation work can be performed. This process means that the traffic in the network as a result of BigInsights-based ETL processing is the refined results that you extract from your system.

Ad hoc analysis

BigInsights can access the unstructured data and provide an *ad hoc analysis*, even before a table is created using BigSheets or Hive. *BigSheets* allows exploration of the data before it is exported to the DW to generate insight and quick answers to questions.

The result

Combining these points together produces a clear case for where BigInsights fits in a big data strategy. BigInsights can be a repository for large volumes of various unstructured information. The data can be analyzed with BigSheets or text analytics to discover where the information that is important to the business is in the unstructured data. The distributed processing power of BigInsights is used with an ETL tool to bring out this information in a structured format that is fed into the DW. This process culminates with the DW being able to provide better, more accurate information to the decision makers in the business.

2.5 Use cases for BigInsights

As a final incentive when you consider BigInsights, a few “use cases” might be helpful. This section wraps up our discussion on “Why choose BigInsights” by covering industry-based use

cases and a specific-use case that takes advantage of many products in the IBM big data platform.

2.5.1 Industry-based use cases

If you were familiar with what IBM did with big data before reviewing this publication, you probably came across several discussions of how big data can be applied to social media. Specific instances of how data from various social media sources might help individuals in the retail and marketing fields include ways to discover customer *intent* to purchase something or to watch a new movie. Another topic that frequently comes up, is the ability to know what customers are really saying about your product or company. Companies no longer have to rely solely on surveys that might only allow four possible options for each question with room for comments at the end. Companies can now see what people are thinking and saying to their friends about a particular store, product, or even company through social media posts online. Although these examples are great use cases, social media is not the only value that big data applications can provide. The following examples provide possible big data use cases for different industries:

- ▶ Transportation
 - Incorporating traffic data, weather patterns, logistics, and fuel consumption to optimize travel logistics
- ▶ Utilities
 - Incorporating weather and energy usage patterns to continually update the efficiency of power generation
 - Monitoring energy usage patterns and weather data to decipher the best way to incorporate reusable and renewable energy sources (for example: wind and solar)
- ▶ Law Enforcement
 - Real-time, multimodal surveillance
 - Rapid detection of cyber security breaches (computer forensics, real-time monitoring of cameras, and situational awareness)
- ▶ Information Technology (IT)
 - Analysis of historical, log data to discover past data breaches and vulnerabilities
 - Analyze log data across the entire data center for indications of the overall health of the data center
- ▶ Financial Services
 - Analytics of customer data to determine behavior patterns
 - Detection of incidences of potential identity theft or fraud
 - Improvement of risk management models through the incorporation of more data, such as a change in a life situation

2.5.2 Social Media use case

Who is posting the most information in the social media realm about your latest product? What are they saying about your company? Your competitors? Your business partners? Maybe you do not know or care at this moment. But, when something happens that can affect your brand, you would probably like to know about it as soon as possible so that you can be *proactive* instead of reactive. In recent times, we have seen situations where a spike in social media activity can be a good or a bad thing, depending on the topic and your point of view.

Companies want to track what is being said and have a better understanding of the demographics of who is talking about their product or service. Companies also want to become more efficient with their marketing spend. To be effective, a comprehensive big data platform is needed, and not just a web-based sentiment service to provide surface-level information. To dig deeper, stay proactive, and retain efficiency, consider the IBM big data platform approach to Social Media Analytics.

The social media challenge

If someone says they like your product, that is a good thing. But in the grand scheme of things, what is that really worth? What actions would you take upon learning this information? And, how much would you pay to learn this? To really understand what actions you might consider taking, you need a deeper level of understanding, in context, and within a period where responding makes business sense.

The big data platform approach

Enter the IBM big data platform and the results of an effort with a company that makes movies. Using BigInsights as a starting point, a connection was made to multiple social media aggregators. Instead of collecting data manually from each website, the aggregators performed this action for us from thousands to millions of websites everyday. This collection method enables you to focus on the analysis instead of the collection of the data. Comments and details were pulled from the web through these aggregators to collect text-based comments for over 130 movie titles as a baseline.

When collected, they were processed by the BigInsights text analytics engine to not only determine sentiment (good or bad), but more levels of information (including demographics and more detailed topics about the movie trailer). For example, “The plot was confusing” or “I liked the music a lot” were discovered automatically in the data at rest. Additional details like “plot” and “music” are details that the marketing department can use to change their message by tweaking the movie trailer before the release of the movie.

When it comes to running an advertisement (movie trailer) during a large event, such as the Super Bowl, the stakes get even higher. Typically, the cost for running an ad is very high. And if the advertisement was not received well, the cost can go beyond just the expense of running the ad. If you knew in real time that the plot was confusing for ad number 1, you might choose to run tweaked ad number 2 later in the show to win your viewers back almost immediately.

The preceding approach is a high-level description of an IBM example solution for this use case. To better understand the data flow and the processing, review the description that is provided in Figure 2-4.

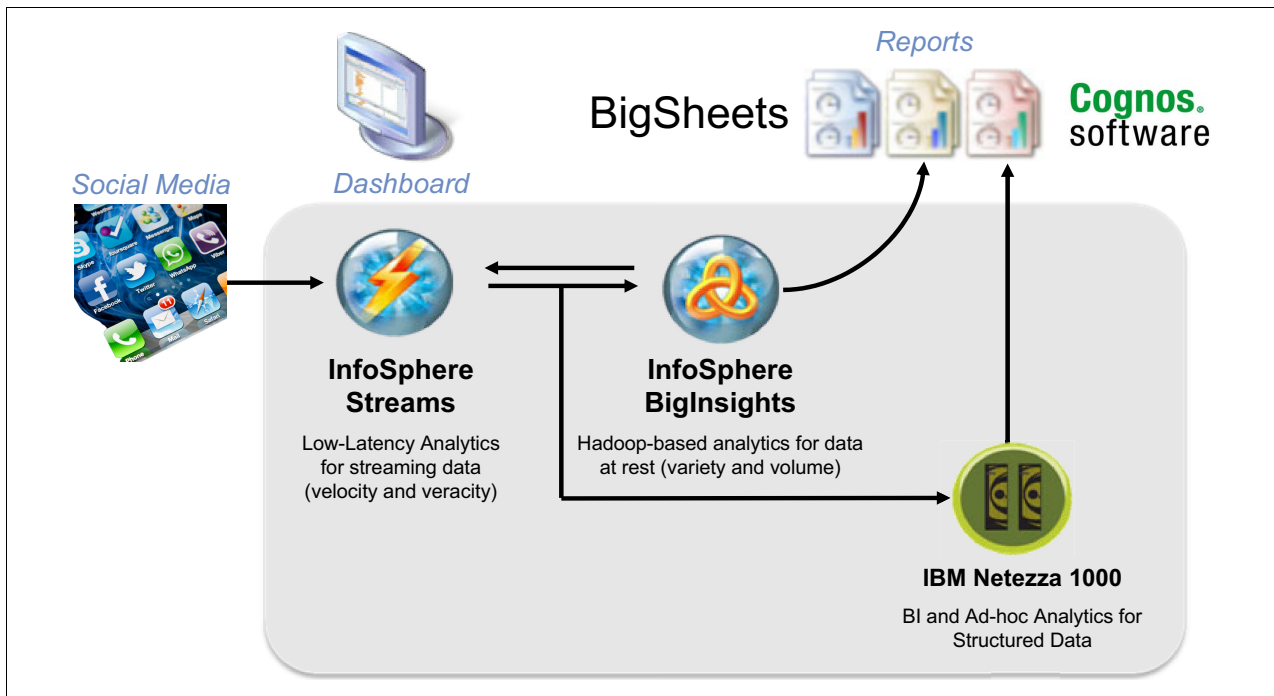


Figure 2-4 Big data platform that is applied to social media

As a quick explanation of the arrows, the data flow, and the processing of the data that is shown in Figure 2-4, start on the left side. The data from the social media aggregators flows into the big data platform. In this case, the need to gauge the positive and negative details about the trailer were determined and displayed on a real-time dashboard. Next, the raw data was stored in BigInsights for future processing. This area provides the baseline for the quantity of comments for each detailed topic.

Then, depending on the details that are contained in the text data, the demographic data is stored in the IBM Netezza 1000 system as structured data. This type of data was analyzed by using Cognos software to produce reports for users to slice-and-dice through for a better understanding of the audience that is commenting on their marketing efforts. Each portion of the platform serves a specific purpose and we apply them where they fit. This fit-for-purpose approach is the key to your potential success when you are considering the IBM big data platform for this, or any other use case.



BigInsights network architecture

This chapter describes the considerations to think about when you design and implement the network portion of a BigInsights cluster. Also described are the various terms to be aware of regarding networks and network architecture. Additionally, goals and design options are described as well as various types of configurations. When you select the hardware components for your BigInsights network, certain advantages might be gained by selecting IBM hardware. We cover some of the advantages in which to be aware. Finally, we cover the suggested steps to follow when building racks of machines for BigInsights that must be networked together.

3.1 Network design overview

Network connectivity and the correct configuration are key aspects of a well-functioning Hadoop cluster. During the sort and shuffle phase of the MapReduce framework, the network can often be a bottleneck to running jobs in the cluster. Network throughput is also important during phases when data is being ingested into the cluster. Considerations of network throughput, cost, port bonding, network isolation, network scaling, and high availability, all must be evaluated based on the current and future cluster requirements of the client. This chapter focuses on the design decisions and trade-offs that must be made to enable the optimal network in a Hadoop cluster.

When creating the design of the IBM suggested network configuration, we considered many design goals. First, we wanted to ensure that performance and cost were the main drivers. Selecting the correct components to achieve the wanted cost-for-performance ratio is important. We created the following information to assist you with the decisions that are required, although allowing for the possibility to consider alternatives.

Additionally, we wanted to ensure that the network topology is scalable to several thousand servers. We use the building block approach of growing one rack at a time, and adding the new rack to the current cluster while keeping the network reconfiguration to a minimum. We also chose to adhere to open standards where possible. Last, we kept the design simple by providing only a few suggested configurations.

Typical network data speeds are 1 gigabit per second (Gbps), 10 Gbps, and InfiniBand (which is typically 40 - 56 Gbps). As you begin to look at different hardware that can be used to create a network of computers, you discover that there are different data rates listed on these hardware devices. A 1 Gbps network device is able to transmit data between machines across the network at speeds up to 1 gigabit per second. It is generally assumed that full duplex mode is supported such that 1 Gb can be independently received and transmitted. Similarly, a 10 Gbps device runs 10 times faster than a 1 Gbps device. And finally, InfiniBand networking technology can deliver 40 Gbps or 56 Gbps, depending on the InfiniBand hardware options that you decide to use.

InfiniBand: InfiniBand technologies are not covered.

Networks are connected by hardware that is known as *switches*.

Switch: A *switch* is a device that receives, filters, and forwards data packets between nodes on a network. A switch usually runs on the data link layer of the OSI Reference Model but can also run in the network layer because it supports most packet protocols, for example, IPv4. With switches, you can build a network of connected computers and are also able to intelligently forward requests from machine to machine efficiently. Switches are a required hardware component of a BigInsights network and are described in this chapter as a key component of the BigInsights network architecture.

3.2 Logical network planning

To properly design the network architecture for the cluster, several questions regarding the cluster must be addressed:

- ▶ Does the cluster require a 1 Gb or 10 Gb node to switch throughput for the performance network?

- ▶ Is switch redundancy required?
- ▶ Is node adapter redundancy required?
- ▶ What is the current and planned future size of the cluster?
- ▶ For multi-rack configurations, what is the spine switch throughput requirement?
- ▶ What are the specific network zones required for the cluster?
- ▶ How will the cluster be connected into the existing client's infrastructure?

The answers to the preceding questions determine, to a large extent, the network capabilities of the cluster.

3.2.1 Deciding between 1 Gbps and 10 Gbps

Today, 1 gigabit (Gb) Ethernet is the standard connection between commodity servers and switches. Ten Gb Ethernet is typically used to network switches together by a spine router and is used for more high-end performance applications. See the following examples:

- ▶ Commodity: 1 Gb TOR Switch to Server Node, 10 Gb spine switch to TOR switch
- ▶ Performance: 10 Gb TOR Switch to Server Node, 40 Gb spine switch to TOR switch

Networking: IBM System Networking (see <http://ibm.com/systems/networking/>). This technology is now included in many of the IBM switch offerings. The advantages of using it over other switch technologies are described later in Chapter 3.

Top of Rack switch: Also denoted as TOR, the *Top-of-Rack switch* is the hardware component that is responsible for connecting all machines within a single rack to each other within one network or network segment.

Spine switch: A *spine switch* is a switch that is used to connect two or more TOR switches together, typically used in a multi-rack cluster. To not become a bottleneck in terms of performance between racks of hardware, these switches typically connect to multiple, faster ports such as 10 Gbps, 40 Gbps, or even faster.

With the advent of Hadoop, 10 Gb Ethernet is quickly becoming a requirement of many clients that require high performance. Because Hadoop is a batch processing environment that deals with large sequential payloads, more focus is given to the throughput metrics in the network rather than latency considerations. The main value of the 10 Gb node-to-switch connection and the 40 Gb TOR-switch-to-spine-switch networking occurs for clusters that have very high input and output throughput requirements during a Hadoop Distributed File System (HDFS) data ingest or data extraction. Or, the main value is for Hadoop applications that have a very large requirement during the sort and shuffle phase of MapReduce.

Performing a sort is a simple example that highlights the usage of the network during the sort and shuffle phase of MapReduce. If a 10 TB data set must be sorted, at least 10 TB must be read from HDFS into the map tasks. The key value pair outputs of the map tasks are temporarily spilled to disk before being sent to the correct reducer. In the worst case, assuming that there is no data locality during the reduce phase, 10 TB of data is transferred across the network. However, in practice the real percentage is lower. In situations such as this, 10 Gb signaling improves the run time of the application. To answer the data locality question during MapReduce, it is useful to run small-scale Hadoop jobs and analyze the output. See 9.1.3, “Job and task counters” on page 129 for a description of some of the typical job counters and how to interpret the values.

3.2.2 Switch and Node Adapter redundancy: costs and trade-offs

Depending on how you plan to use your BigInsights cluster, redundancy might be worth the additional cost. If a networking component has a failure and there is only one component of this type within the system, the cluster becomes unavailable. Any jobs that were running, might need to be restarted. Depending on the job that was running, some form of data loss might also occur.

If your BigInsights cluster contains data that is vital to the operation of your business, you might want to protect the cluster from a network hardware failure. If the cluster is a single-rack cluster with a single TOR Switch, a failure in the switch causes the cluster to have an issue. If this type of failure is acceptable, the extra expense of a redundant switch can be avoided. If not, consider a redundant switch.

Likewise, within the network topology, ask yourself, “What if a network connection device failure occurs within the management node or data node?” Again, if your data and workload can accept this type of failure, the expense of a redundant network adapter for each node can be avoided. If not, a redundant adapter for each node needs to be purchased. As a reminder when you are deciding, HDFS provides a degree of availability by storing three copies of the data (across multiple racks, when available), by default. The number of copies can be changed to a higher or lower number if wanted for each file, or even across the entire cluster.

3.3 Networking zones

Another important topic involves networking zones. You might be aware of virtual local area networks (VLANs) or subnet addressing as a way to separate network traffic that flows through your network switches. In this case, we are calling a VLAN a *zone*. By assigning different IP address ranges within a zone, network traffic is kept separate, which provides better performance and a way to restrict direct access to certain servers from certain users on the corporate network. We configured three zones within the cluster that we used while writing this publication; namely, a Corporate Management network, a Corporate Administration (Admin) network, and a Private Data network. The corporate network that we used was connected to the Corporate Management and Corporate Admin networks within our cluster only.

The corporate network that we are referring to is the network within your company that users access with their machines. This network might also have various VLANs and subnet assignments. Therefore, when we refer to the corporate network, assume that you have access to this network as a way to obtain access to the BigInsights cluster, as shown in Figure 3-1.

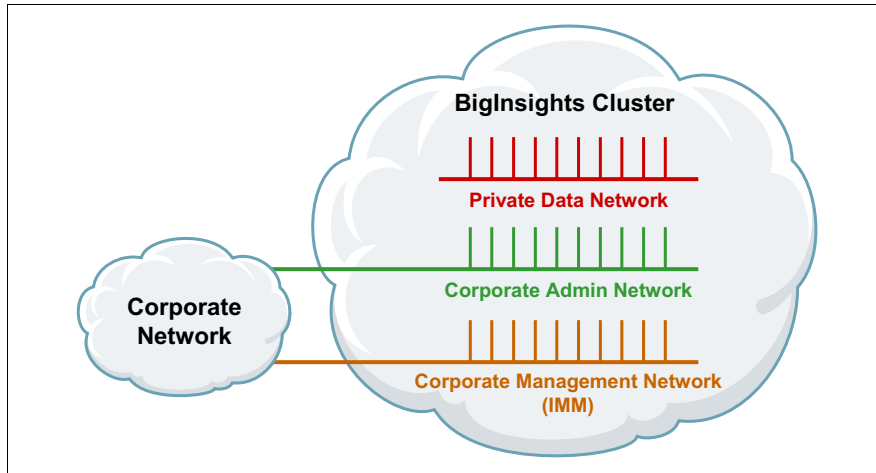


Figure 3-1 BigInsights network zones

3.3.1 Corporate Management network

It has become standard practice to provide a *Corporate Management network* that is separate from other networks. By connecting this network, sometimes called the *integrated management module (IMM)*, to the corporate network, authorized users have access to each node within the cluster to perform activities. Examples of applicable activities include: Checking the system health summary information, restarting the machine, updating the firmware, changing system settings, optionally generating alerts that are based on thresholds, and so on.

Because of the importance of these actions, a separate network zone is typically designated for this type of communication with the nodes. By keeping this traffic separate, better response time from the nodes is usually achieved, allowing hardware management and monitoring activities to not interfere with the network performance of the cluster. This network can be run on slower hardware because of the reduced amount of overall data that is required to do these activities.

3.3.2 Corporate Administration network

On a separate VLAN and subnet, the *Corporate Administration network* can be used for data ingestion from other machines on the corporate network. The settings for this network within the BigInsights cluster potentially need to follow specific corporate requirements, such as Dynamic Host Configuration Protocol (DHCP) and valid address ranges. This configuration provides flexibility for data ingest and node administration. This structure also allows for Hadoop clients to be on the corporate network.

A disadvantage to this configuration approach is the requirement for each node in the cluster to have an IP address within the corporate network. Additionally, this network requires more overall administration and configuration to be performed within the cluster and the corporate network hardware.

3.3.3 Private Data network

The network zone that is labeled *Private Data Network* in Figure 3-1 on page 33, provides a private interconnect for all BigInsights nodes. It is called private because users on the corporate network are not able to directly access machines by their private data network IP address. This network needs its own VLAN and subnet, just like the other two network zones we already described. Typically, the *private data network* uses network address schemes that are similar to 192.168.x.x, 172.6.x.x, or even 10.x.x.x. This information is helpful to know because these network address ranges are not generally used within the corporate network. Using one of these numbering schemes avoids confusion when you configure your network to run BigInsights and connecting your nodes within your private data network.

There are a few advantages to setting up a Private Data network. One example is the dedication of this network, both in terms of hardware and addressing, to potentially provide better performance by not competing for network resources when BigInsights jobs are running. Transferring data between nodes on this dedicated Private Data network typically runs faster than the same speed network that is connected to the corporate network. Another advantage is the ability to block communication from users on the corporate network from directly using network resources within the Private Data network. This capability can assist with security of the servers when configured correctly.

One more option for this Private Data network is to use this network for dedicated data ingestion. For example, if you wanted to transfer data from one BigInsights cluster to another, the fastest way to do this is to have both clusters on the same Private Data network. This way, data transfer between clusters is allowed to use all of the resources that are dedicated to the Private Data network. This process allows the transfer to occur at the fastest network rate that is based on the current workload across the network.

3.3.4 Optional Head Node configuration considerations

As a node, the optional *Head Node* is a computer that acts as the boundary between the BigInsights cluster and your corporate network. Any communication that flows into and out of the cluster must go through the Head Node without any routing. This node contains an interface into the customer network and another interface into the BigInsights cluster, as shown in Figure 3-2.

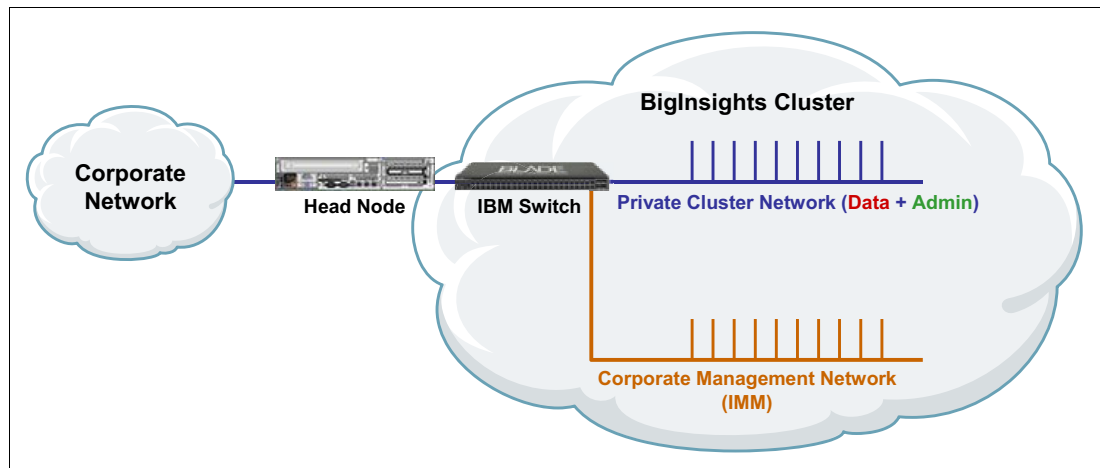


Figure 3-2 BigInsights cluster using a Head Node

Because the Head Nodes are essentially gateways, they need higher network throughput than the other nodes within the cluster. The Head Nodes must be properly sized to ensure

that they do not become a performance bottleneck for network traffic flowing into and out of the cluster. For example, this scenario might become an issue when running a data ingest job where data is coming from the corporate network. As a way to alleviate this potential bottleneck issue, there can be multiple Head Nodes that are connected to one BigInsights cluster. This configuration potentially provides improved performance and higher availability.

3.4 Network configuration options

As a way to more easily incorporate all of the decisions regarding network redundancy, cost, and performance, we created the following named configuration options:

- ▶ Value configuration
- ▶ Performance configuration
- ▶ Enterprise configuration

For each of these configurations, cost for performance is considered and the advantages and disadvantages of each are documented, for a single rack cluster, in the following sections. (Multi-rack clusters are covered in 3.6, “How to work with multiple racks” on page 40.)

3.4.1 Value configuration

The *value configuration* is the least expensive configuration option. It has the lowest cost network because of the selection of a 1 Gbps network speed for its TOR switch. The disadvantages of this configuration include the slower data movement speed between nodes as compared to the other configurations. Another disadvantage is the single point of failure (SPOF) that exists by having only one TOR switch. As described in the preceding chapter, certain phases of a MapReduce job can involve extensive movements of data between nodes in the cluster. Therefore, if you find that your workload requires more network traffic than usual, you might consider one of the following options.

An example of the value configuration is shown in Figure 3-3.

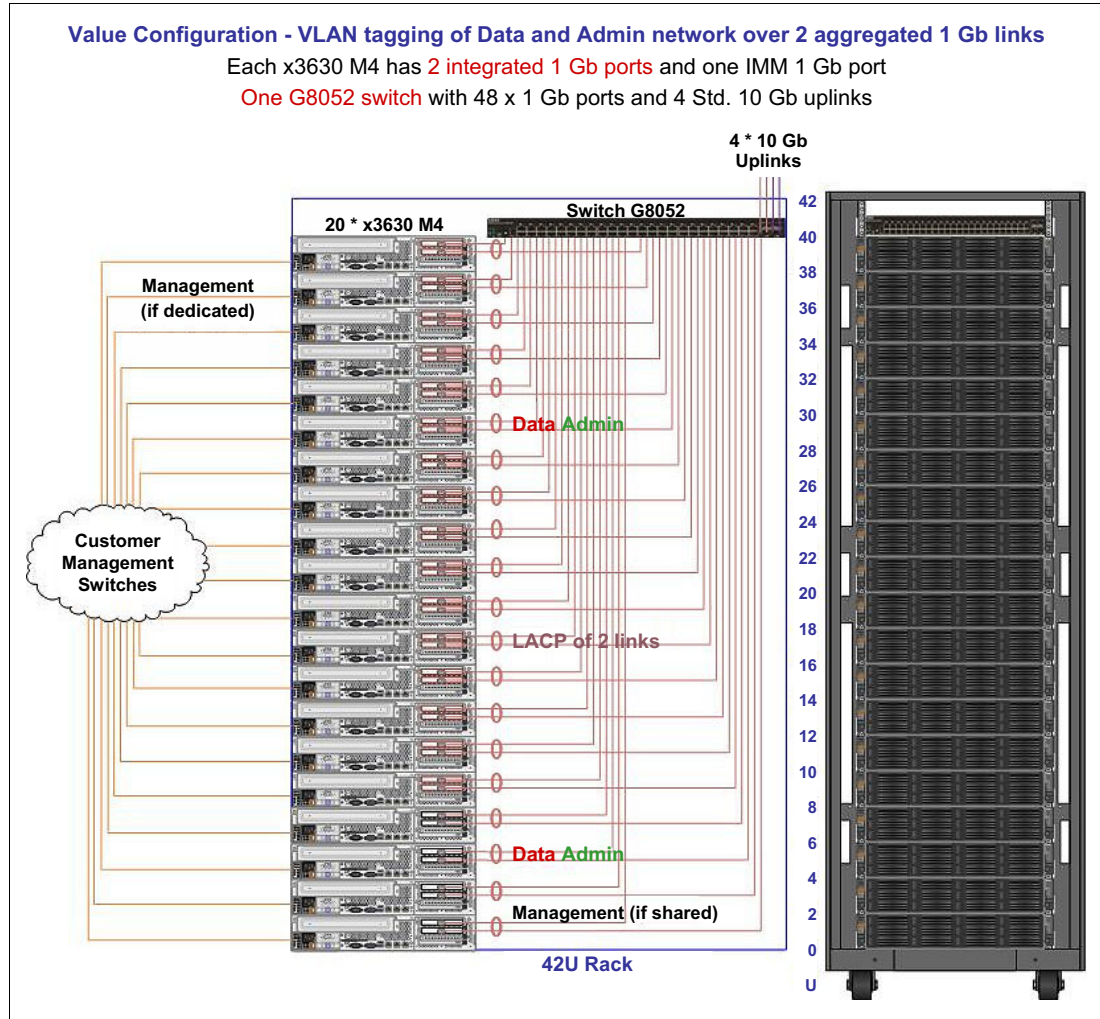


Figure 3-3 Value network configuration

3.4.2 Performance configuration

To provide better network performance, the *performance configuration* uses the 10 Gbps network hardware instead of the 1 Gbps switch that is seen in the value configuration. This setup is a simple configuration that involves one 10 Gbps switch and a dual-port adapter in each node. By linking these ports together, improved high availability and throughput are achieved. However, like the value configuration, a SPOF exists if the switch fails or if the adapter fails. If these multiple SPOFs are not acceptable, you might want to consider adding the *enterprise* option to your performance configuration. This option is covered in 3.4.3, “Enterprise option” on page 37.

An example of the performance configuration can be seen in Figure 3-4.

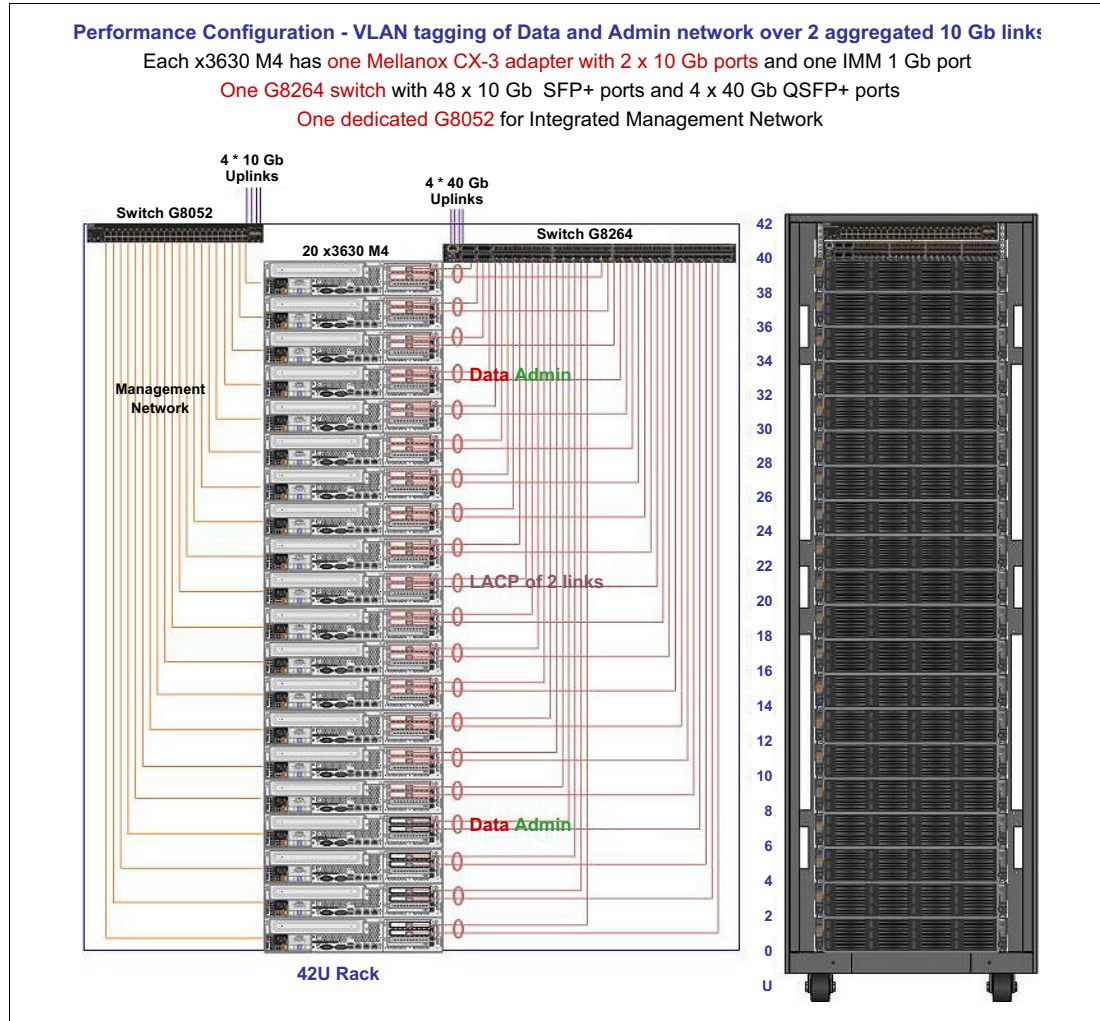


Figure 3-4 Performance network configuration

3.4.3 Enterprise option

When you think of enterprise-level software, you might think of reliability as one of the main characteristics. The *enterprise network option* provides reliability at the network level through hardware redundancy. This option can be added to either one of the preceding configuration options.

Value enterprise configuration

To remove the SPOF from the environment, a redundant TOR switch is added as part of the *value enterprise configuration*. Redundant adapters are also added to each node to connect to the redundant switch. Additional throughput and improved high availability are provided by this configuration. The only disadvantages are the additional expense and cabling that are required to connect up all of the additional hardware.

An example of the *value enterprise configuration* is shown in Figure 3-5.

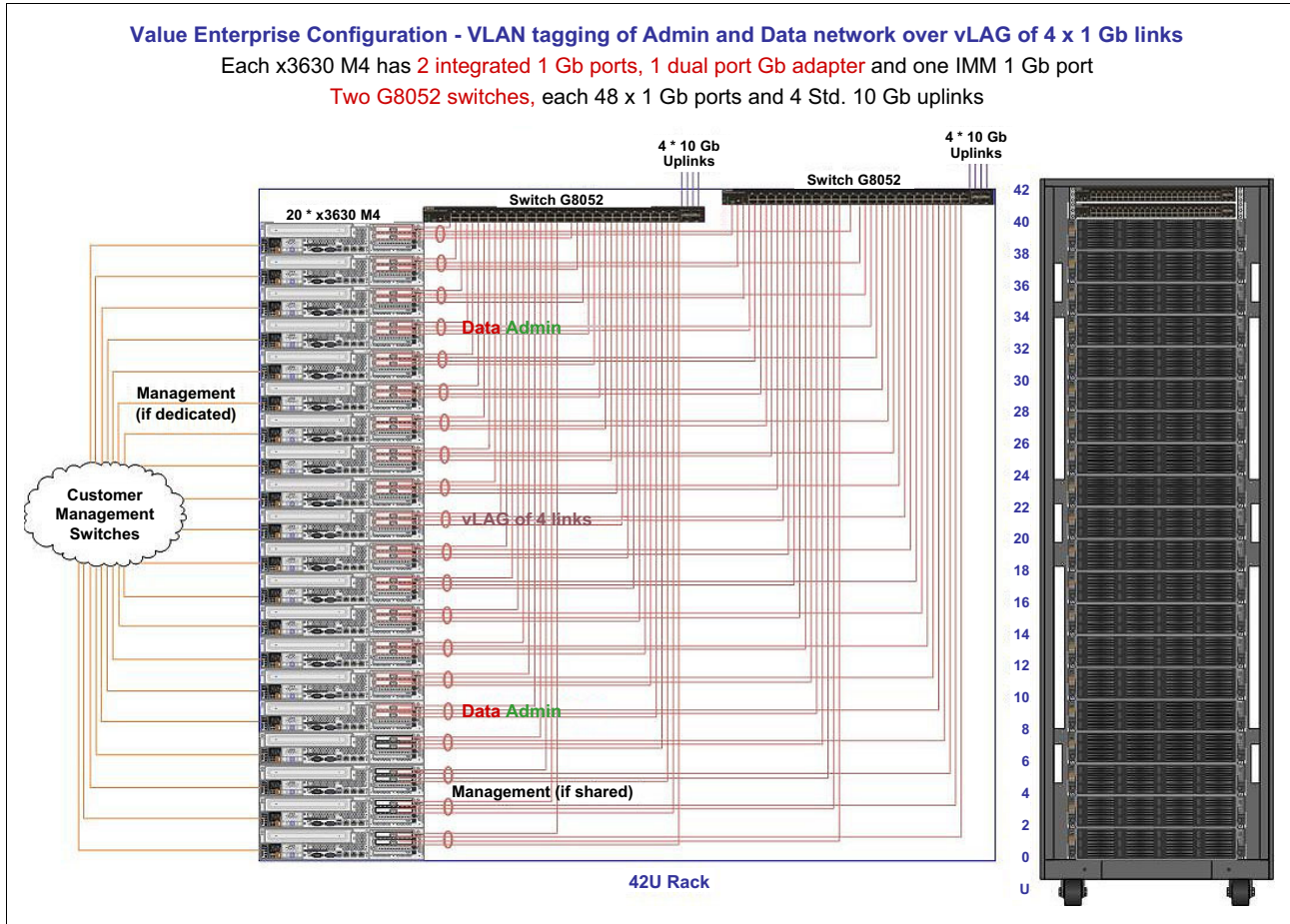


Figure 3-5 Value enterprise network configuration

Performance enterprise configuration

To remove the SPOF within the performance configuration, the *performance enterprise configuration* introduces redundant switches and redundant network adapters in each node. As a result of this process, high availability and throughput are increased and the SPOF is removed. The increased throughput is achieved by adding *Virtual Link Aggregation Groups (vLAGs)* and connecting the two ports of each adapter to different 10 Gbps TOR switches. As you might expect, the same disadvantages exist with this option. These drawbacks exist because more cabling is involved and there are more expenses that are related to the additional hardware that is required by this configuration.

An example of the *performance enterprise network configuration* is shown in Figure 3-6.

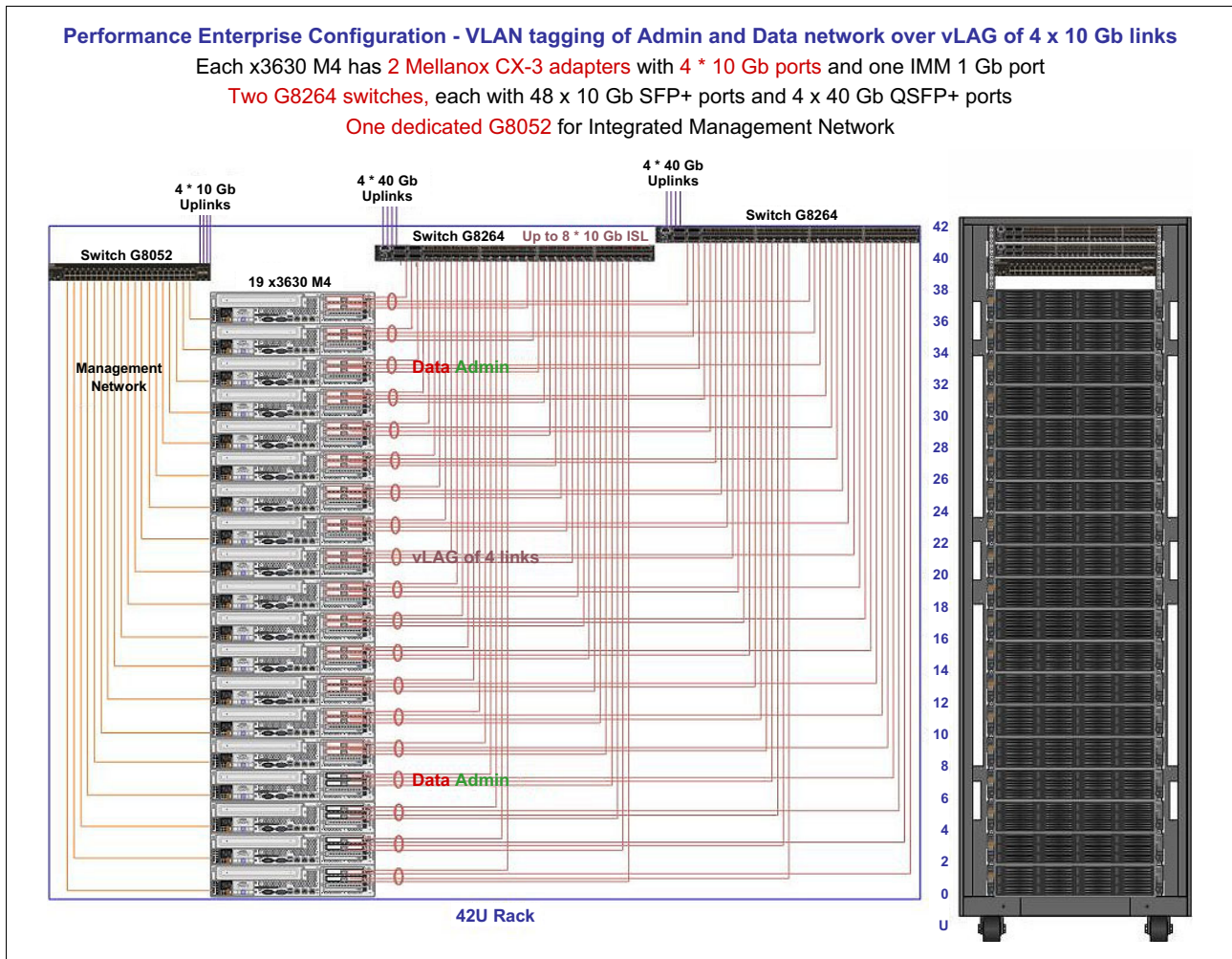


Figure 3-6 Performance enterprise network configuration

3.5 Suggested IBM system networking switches

Up until this point, we only covered clusters that fit within one rack. In Chapter 4, “BigInsights hardware architecture” on page 49, we describe the various options for single rack hardware configurations. The images of the various network configurations thus far show full-rack implementations. Within each of these configurations, one or more TOR switches are included in the design. To be more specific, this section names the particular TOR switch models that are included as part of the *BigInsights Reference Architecture*.

3.5.1 Value configuration switches

The main TOR switch that is used in the *value configuration* is the IBM System Networking RackSwitch™ G8052, as shown in Figure 3-7.

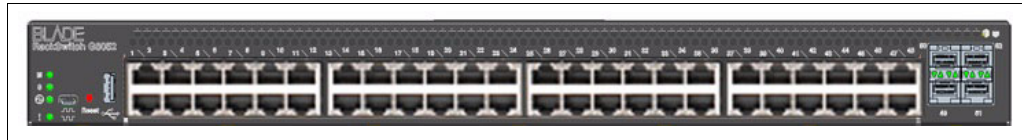


Figure 3-7 IBM RackSwitch G8052

This switch provides 48 10/100/1000 RJ-45 ports with built-in 10 Gbps uplinks so there is no need for separate modules. With the 48 RJ-45 ports, there is no need for RJ-45/SFP transceivers. This switch also provides the option to use vLAG for better performance and scalability. To match the airflow of your rack, the *G8052R* provides rear-to-front airflow. And finally, this unit has full Layer 2 and Layer 3 support.

3.5.2 Performance configuration switch

The main TOR switch that is used in the *performance configuration* is the IBM System Networking RackSwitch G8286, as shown in Figure 3-8.

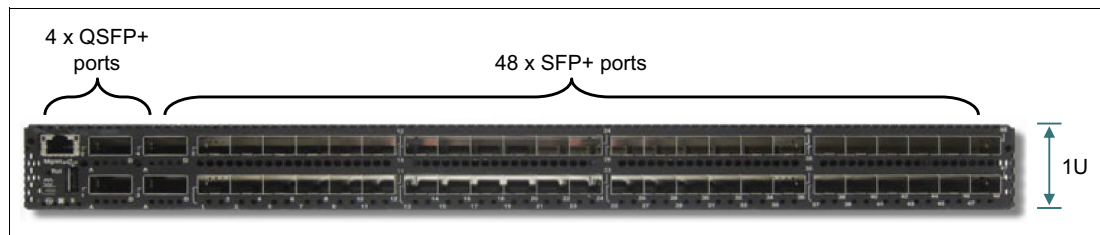


Figure 3-8 IBM RackSwitch G8286

This higher performance switch provides 48 RJ-45 ports that can support 10 Gbps or 1 Gbps connections. To support the slower, 1 Gbps connection to a node, an additional transceiver for the port connection at the switch must be purchased. This switch also has four, 40 Gbps connections that support a QSFP to SFP+ breakout cable. Therefore, this switch can be configured with 40 Gbps uplinks or, with the breakout cable, provide up to 64, 10 Gbps ports. This unit also provides IBM VMReady™ and Virtual Fabric support for virtualization. Lastly, the *G8264R* is an optional model that provides rear-to-front airflow to make it easier to match your rack's airflow across this switch, depending on your preference for mounting orientation.

3.6 How to work with multiple racks

Thus far, we described how to plan for, logically configure, physically arrange, and potentially wire up nodes to your network within a single rack. If you want to run a cluster with more nodes than a single rack can contain, a multiple-rack setup is required. Because this chapter is focused on networking, we describe the network items to consider when you work with multiple-rack clusters.

Similar to the approach for single rack cost-for-performance decisions, the multi-rack configurations have similar challenges. For example, rack-to-rack connection speeds, SPOFs, and how best to handle the concept of circular traffic as more switches are added to the

architecture, now must be considered. To simplify things, we organized multiple-rack configurations into *value* and *performance* selections, with an *enterprise* option for each of these configurations. In the following sections, we describe the options to consider when you work with various quantities of racks within a single cluster.

Before describing the configurations, the concept of a *spine switch* is defined. A spine switch is used to connect TOR switches together to create a single cluster from multiple racks of machines. For the higher performance option, a unique piece of hardware is required. However, for the value configuration, you discover that the same, IBM G8264 TOR switch can now be used as a spine switch. Generally speaking, your spine switch needs to provide more speed and overall capacity to handle rack-to-rack TOR switch traffic. The switches in a cluster are typically connected with SFP+ and QSFP+ connection cables.

SFP+ and QSFP+: *SFP* stands for *small form-factor pluggable*. When the plus sign is added, this means that the SFP+ connector can support speeds up to 10 Gbps. The Q in front of SFP+ stands for quad (or four). *QSFP+* has four 10 Gbps connections for a total connection speed of 40 Gbps.

In a multi-rack cluster, it is worth considering where to place the spine switch. Generally, we find that the best positioning is to place the switch in the most central rack to minimize the amount of cable that is required. This placement is beneficial because longer cables have slightly higher lag times.

3.6.1 Value configuration

Based on the number of racks that you want to include in the cluster, the additional amount of hardware that is required varies. Similar to the single-rack configuration, the multi-rack value configuration is the lowest priced option, but also contains a SPOF at the spine switch.

Value configuration: up to 12 racks

The easiest way to explain the value configuration for clusters up to 12 racks is to understand the idea of placing a faster switch with enough ports in-between all of the TOR switches. An example of this configuration is shown in Figure 3-9.

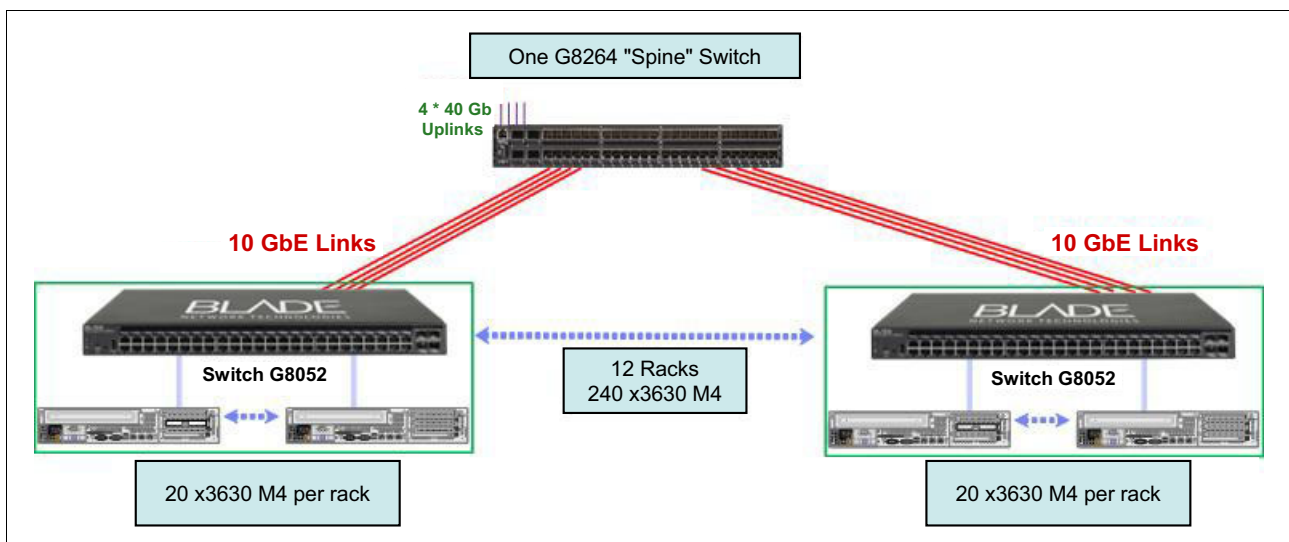


Figure 3-9 Multi-rack value network configuration for up to 12 racks

Value configuration: 13 - 24 racks

An example of the value configuration for up to 24 racks is shown in Figure 3-10.

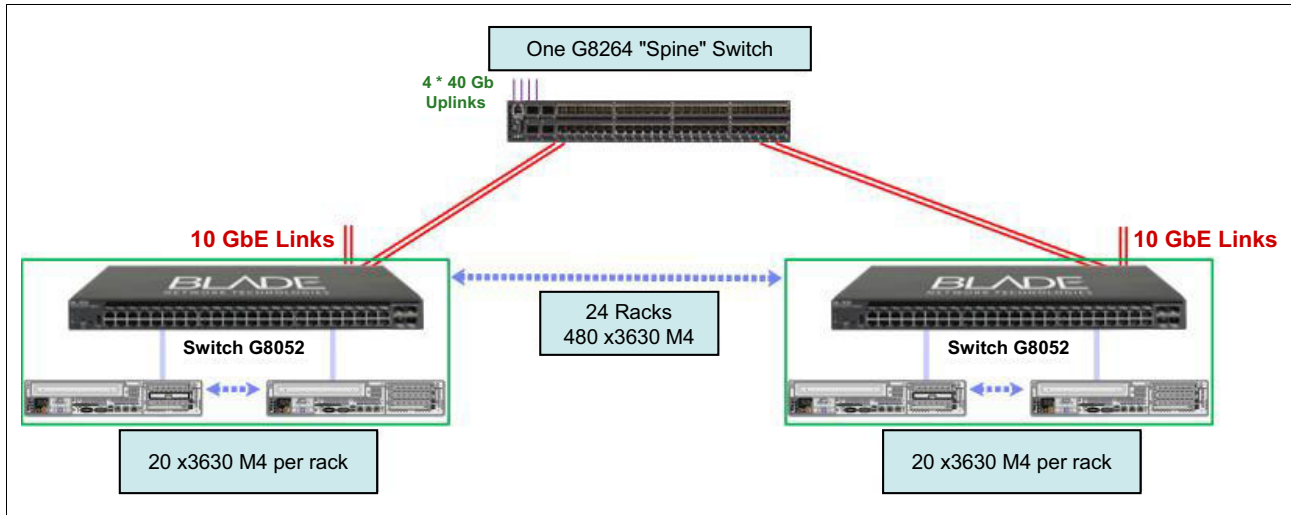


Figure 3-10 Multi-rack value network configuration for 13 - 24 racks

Value configuration: 25 - 48 racks

An example of the value configuration for clusters with 25 - 48 racks is shown in Figure 3-11.

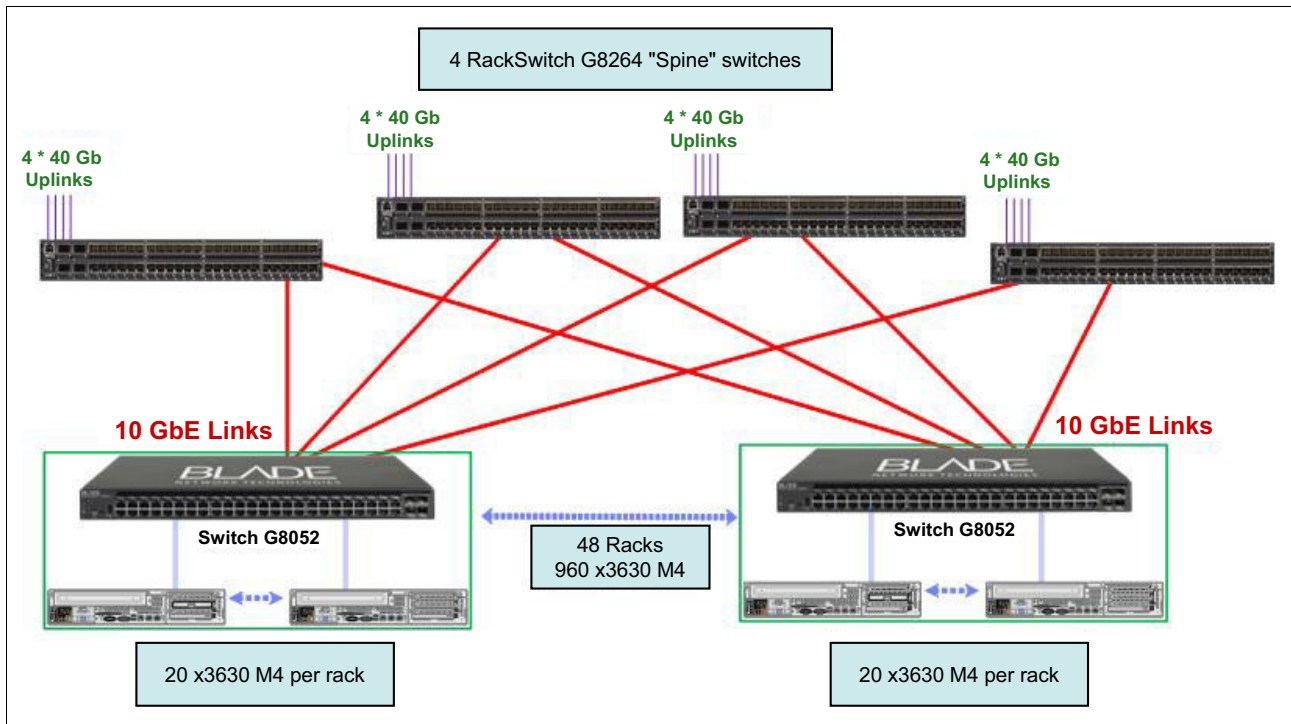


Figure 3-11 Multi-rack value network configuration for up to 48 racks

3.6.2 Performance configuration

The performance configuration for multiple racks is more expensive than the multi-rack value configuration. This is mainly because of the higher speed network hardware that is required. To connect the performance configuration's TOR switches together, a higher speed spine

switch is now introduced. Then, we provide examples of different multiple-rack configurations that provide higher network data transfer speeds.

High-speed spine switch

To provide a high-speed spine switch for a multi-rack cluster that is built from single-rack performance configurations, a high-speed spine switch must be used. As a way to provide a faster Management Processor (MP) than a G8264, the multi-rack performance configuration includes the *IBM System Networking RackSwitch G8316*. A picture of this switch is shown in Figure 3-12.

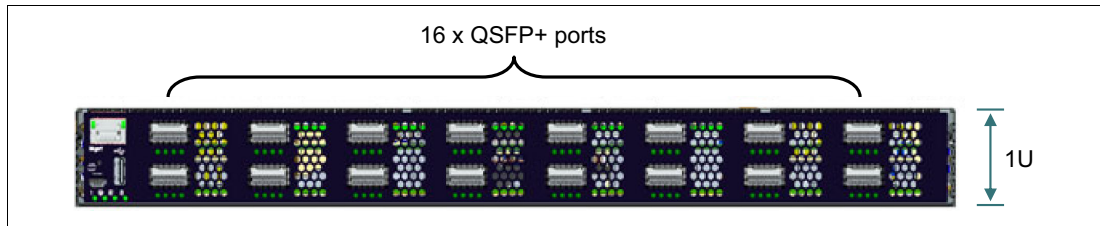


Figure 3-12 IBM RackSwitch G8316

Performance configuration: up to three racks

As you might expect, the performance configuration includes one RackSwitch G8316 as a spine switch. Because this configuration is not the *enterprise* version of the configuration, this spine switch adds a SPOF to the environment. Again, more spine switches are needed as the number of racks gets larger.

An example of the performance configuration that supports up to three racks is shown in Figure 3-13.

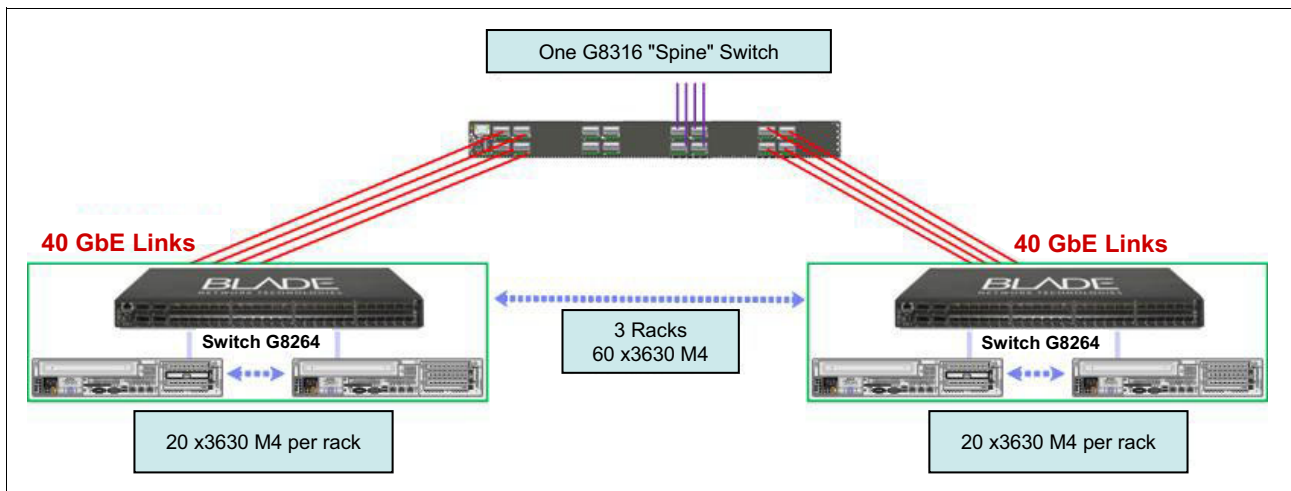


Figure 3-13 Multi-rack performance network configuration for up to three racks

Performance configuration: four - seven racks

An example of the performance configuration that supports four - seven racks is shown in Figure 3-14 on page 44.

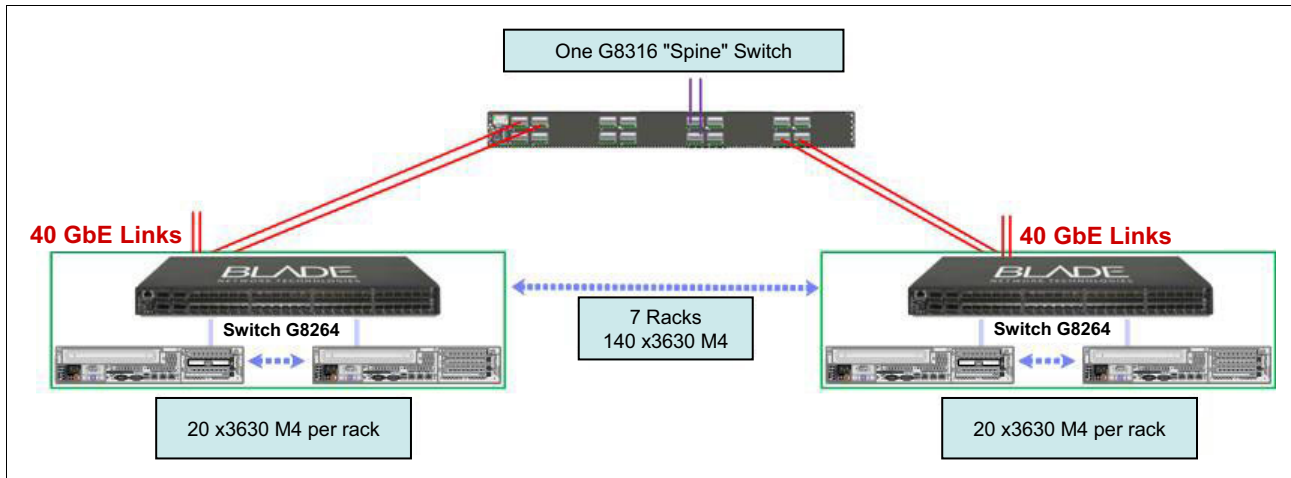


Figure 3-14 Multi-rack performance network configuration for four to seven racks

Performance configuration: 8 - 15 racks

An example of the performance configuration that supports 8 - 15 racks is shown in Figure 3-15.

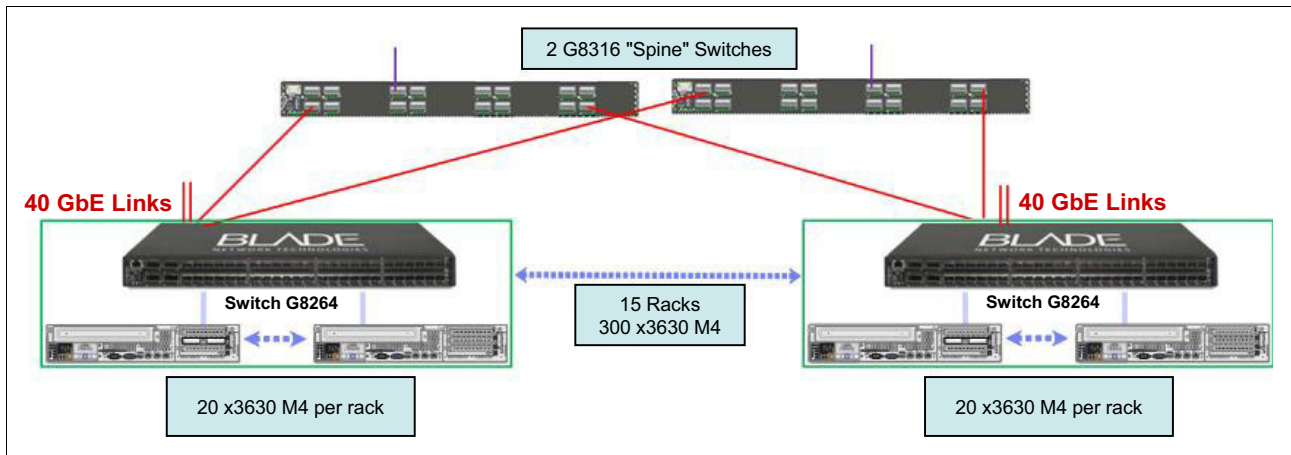


Figure 3-15 Multi-rack performance network configuration for 8 - 15 racks

3.6.3 Enterprise option

For the single-rack configurations, a SPOF exists if you do not select the enterprise option. For all of the previous multi-rack examples, they too have SPOFs between racks. If your BigInsights cluster contains data or performs processing that is critical to your business, you might want to consider one of the two types of enterprise configuration options provided in the following examples.

Value enterprise configuration: 13 - 24 racks

The *multi-rack value enterprise configuration* starts with the single-rack value enterprise option and then adds redundant spine switches. These additional switches remove the SPOF, although, they potentially improve network performance by providing more channels for data to flow through (depending on the number of racks within your cluster).

There are many example diagrams that we can provide here. However, to keep things simple, we show only one of the examples that is based on 13 - 24 racks. This example is shown in Figure 3-16.

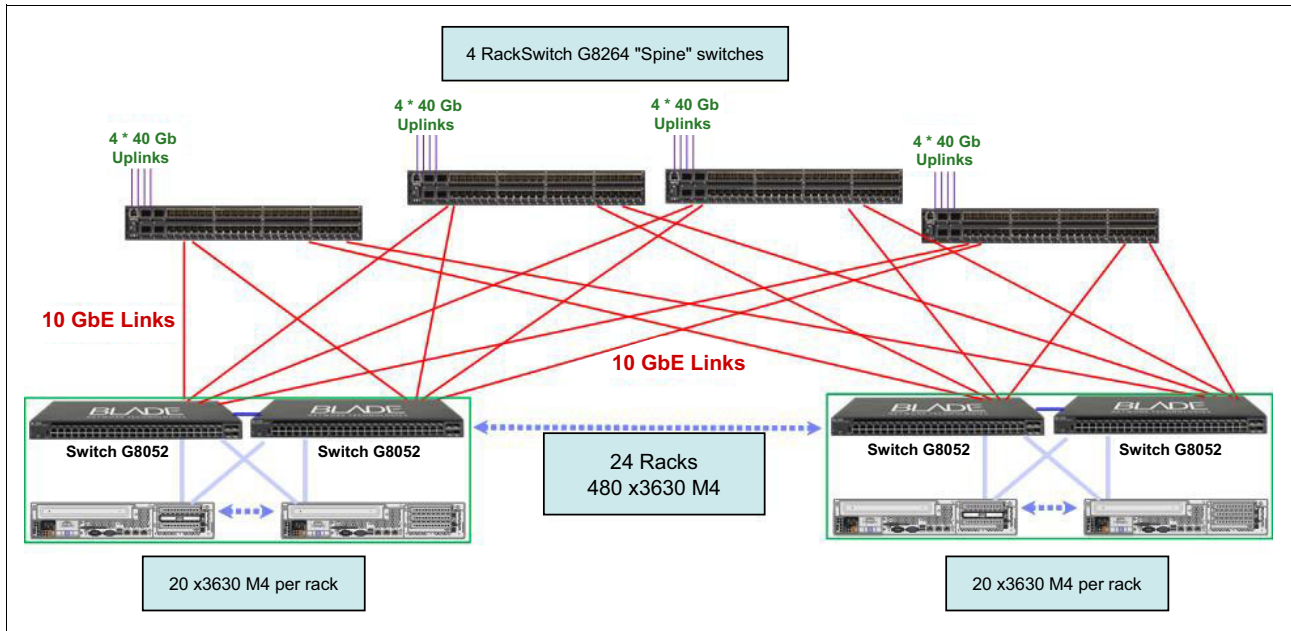


Figure 3-16 Multi-rack value enterprise network configuration for 13 - 24 racks

Performance enterprise configuration: three - seven racks

To reinforce the concept, we include only one multi-rack performance enterprise configuration example. One of many possible examples can be seen in Figure 3-17.

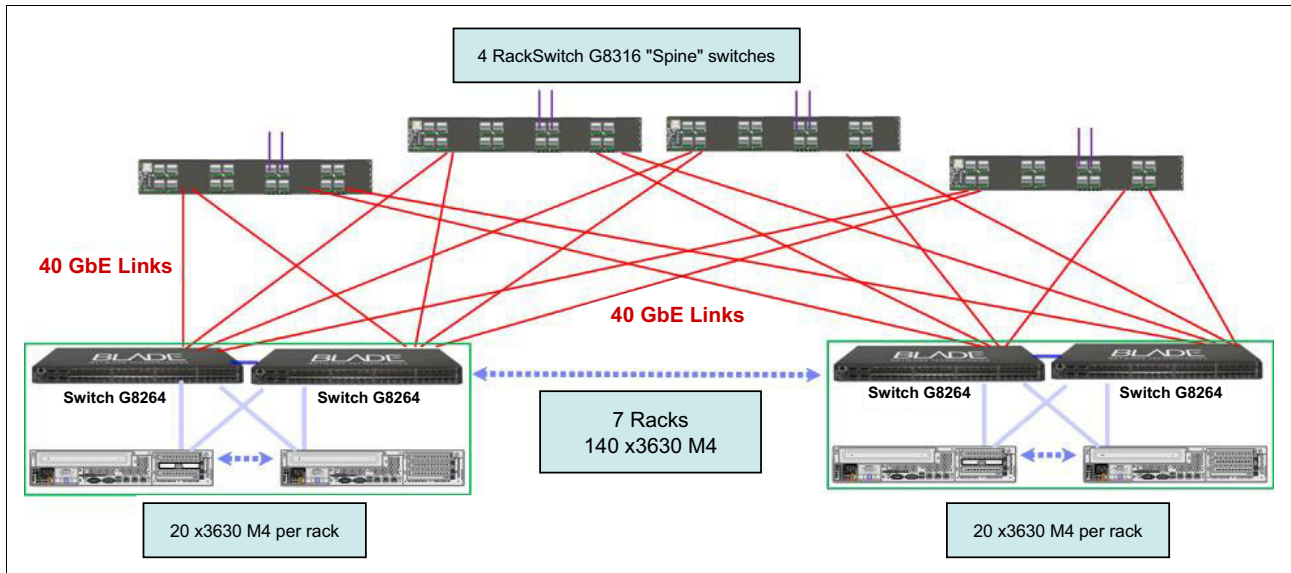


Figure 3-17 Multi-rack performance enterprise network configuration for three - seven racks

3.7 How to improve performance

As shown in this chapter, there are many considerations when it comes to networking. The decisions and trade-offs of these options were described. Now you should have enough information to determine your logical network design. However, there are a few more decisions that you might want to consider based on potential ways to ensure better performance. These decisions and trade-offs are described in the following sections.

3.7.1 Network port bonding

When a computer contains two network ports of the same speed, there is a way to tie them together in a process called *bonding*. This process involves software settings on the node and matching settings on the switch. Because the most common failure within a network is the cable between the node and the switch, bonding is a way to avoid a SPOF across this cable. By assigning and sharing network traffic across the two ports instead of just one, the SPOF is removed.

The additional benefit of bonding is that the data transfer rate can be doubled because both ports are now responding to the data transfer request. We used “can be” here because most situations use both ports, but there are some instances where only one of the two bonded ports is used. This is one of the reasons that this section describes how you can improve your *chances* of better performance, instead of something more definitive.

3.7.2 Extra capacity through more hardware provided for redundancy

The concept of *redundancy* was covered earlier in this chapter. By introducing the concept of an *enterprise option* for your network, more hardware is added to provide redundancy. Similar to the network port bonding example, more network cards can be added to the nodes. Because the cards are primarily provided for redundancy, it also provides more ports that can be bonded for more throughput in most cases, as mentioned in 3.7.1, “Network port bonding” on page 46.

The enterprise option also provides for more network switches. Although switches are provided for redundancy, more throughput is also a potential benefit when you choose from one of the enterprise options. Although not a guarantee of better performance, the enterprise option might improve your chances of better performance.

3.7.3 Virtual Link Aggregation Groups for greater multi-rack throughput

Another area of discussion is the idea of networking loops. In a multi-rack configuration, data can be sent from one node to the TOR switch and then to the spine switch. In some cases, the data can be sent from the spine switch back down to a location that can be viewed as a loop within the network. Loops are not good for network performance and need to be avoided because they can quickly affect performance in a negative way.

To take advantage of the additional throughput potential in rack-to-rack communication across the network, IBM System Networking switches have the *Virtual Link Aggregation Group (vLAG)* feature. Provided by a software update to the switches, vLAGs are an enhancement that allows redundant uplinks to remain active, thus potentially allowing the network hardware to use all of the available bandwidth when necessary. Because vLAGs stop network loops but still provide a throughput channel, consider the use of hardware that supports vLAGs and consider including them in your solution.

3.8 Physical network planning

To assist you with the site preparations that are needed for deployment of your cluster, there are a few areas to consider. You need to do some physical preparations before bringing the network hardware and wiring into your data center. Keep in mind that the items that are listed here are for the network portion of the cluster only. In Chapter 4, “BigInsights hardware architecture” on page 49, we provide more considerations regarding the hardware in your cluster. Also, we provide a list of high-level summary items in Appendix C, “Checklist” on page 189.

3.8.1 IP address quantities and networking into existing corporate networks

Based on 3.3, “Networking zones” on page 32, it is desirable to have IP addresses to enable the three different networking zones: *Corporate Management*, *Corporate Administration network*, and *Private*. As an example in a cluster with one management node and nine data nodes, the quantity of IP addresses is 10 for each segment. Considerations for future cluster size drives how large the subnets should be for each network. If you expect the cluster to grow to 100 nodes, it is advisable to allocate at least 128 or 256 IP addresses for each subnet. It is also a good practice to make these ranges contiguous to ease the administration and configuration of your Hadoop cluster. Make sure that the segment allocated in your corporate network has enough room for IP allocation growth.

3.8.2 Power and cooling

A significant part of the site developments includes ensuring adequate power and cooling supplies. The following values relate to the requirements for a full rack that uses the M4 architecture:

- ▶ Current (at 200 V) = 85.27A
- ▶ Maximum power load = 17604.00 W
- ▶ Total BTU = 60118.89 BTU/hr
- ▶ Total Weight = 893.83 kg

Additional racks are more multiples of these numbers. The power cables to use are DPI single-phase 30A/208 V C13 enterprise PDU (US).



BigInsights hardware architecture

This chapter describes the hardware architecture of BigInsights management nodes and data nodes. Furthermore, there is an analysis of some of the considerations to take into account when you build a hardware solution.

4.1 Roles of the management and data nodes

In a Hadoop system, the management nodes and data nodes have different roles, as outlined in the following subsections.

4.1.1 The management node

There might be several management nodes in a cluster. Management nodes host the NameNode, Secondary NameNode, Backup node, BigInsights Console, and JobTracker services. Because the NameNode is a single point of failure (SPOF) (depending on the version of Hadoop you are running), the management node must not be prone to failure. Therefore, consider high-end, non-commodity-level hardware, the best in the cluster if possible. The NameNode needs a large amount of memory because the metadata for the Hadoop Distributed File System (HDFS) data in the cluster is stored in the memory that is allotted to the NameNode service. Smaller clusters might run one or more of these services on one management node. Situations where this configuration is appropriate are described in 4.2, “Using multiple management nodes” on page 50.

On clusters with multiple management nodes, consider placing them on separate racks to avoid the rack becoming a SPOF.

4.1.2 The data node

The *data nodes* host the TaskTracker and store the data that is contained in the Hadoop Distributed File System (HDFS). In contrast, the NameNode is used exclusively for storing the metadata. Every file in the cluster is divided into blocks that are distributed across the data nodes in the cluster. The location of these blocks is saved as metadata on the NameNode. The data node can be commodity hardware because they provide built-in redundancy.

Block size: Increasing the default block size from 128 MB reduces the amount of metadata that is being stored by the NameNode. However, if your access patterns are not reading large chunks of data linearly, then a larger block size greatly increases the disk loading that is required to service the input/output (I/O).

4.2 Using multiple management nodes

Clusters with less than 20 nodes can generally combine the NameNode, JobTracker, and Console onto one management node. In this instance, the *Secondary NameNode* is hosted on a data node. Having all three processes running on one management node is unlikely to overwhelm the capabilities of the machine. Separate the secondary NameNode so that it is still available if the NameNode fails.

A cluster that expands to 20 - 50 nodes should separate the NameNode, JobTracker, and Console onto three separate management nodes and include the Secondary NameNode on the same server as the Console. This configuration is to ensure that the appropriate daemons have sufficient resources to avoid a bottleneck. Also, consider spreading them across racks.

When the number of nodes exceeds 50, host the Secondary NameNode on its own management node because its memory requirements can overwhelm the memory capacity of the node.

These are guidelines only: Consider all clusters on an individual basis. Factors like available memory and processor speed also affects how many nodes can be supported by individual management nodes.

4.3 Storage and adapters used in the hardware architecture

Before hardware setup decisions can be made, there are a few Hadoop-specific considerations to consider.

4.3.1 RAID versus JBOD

As described in 2.2.1, “Hadoop Distributed File System in more detail” on page 15, the HDFS automatically replicates data to three different locations. This allows the cluster to access the data even if one of these copies becomes corrupted or if the data node that hosts one of the copies fails. *Redundant Array of Independent Disks (RAID) striping* is a technique that is used by many systems to ensure data redundancy. Because HDFS replicates data automatically, RAID is not required on the data nodes. In fact, leaving the hard disk drives as *just a bunch of disks (JBOD)* allows them to perform faster than if they were combined in a RAID array. This faster performance is because the read/write operations of the RAID-arrayed-disks are handicapped to the speed of the slowest drive. However, JBOD disks can perform at the average speed of the disks because the read/write disk operations are independently handled by each disk. Putting disks into a RAID array also has the disadvantage that storage capacity is lost.

4.3.2 Disk virtualization

Combining disks into virtual drives is not usually desirable because this might impair the ability of the cluster to break down MapReduce tasks across the disks. Disk virtualization potentially hinders performance. In our own testing, we discovered Hadoop performs much better when it has simultaneous access to the data disks within a data node.

4.3.3 Compression

When you determine the amount of hardware that your cluster requires, you might consider the use of *compression*. Data compression can be used to reduce the amount of storage that is required for data and speed up data transfers across the network because the files stored on the disk are smaller. Although compression has benefits, there are a few extra items to consider if you decide to use data compression on a Hadoop-based system.

When a file is stored in HDFS, it is divided into blocks that are spread across multiple nodes. This process is true for compressed files in HDFS as well. Because Hadoop uses this file, Hadoop’s MapReduce wants to have each node process the block independently. However, with many common forms of compression, this function is not possible because the file must be fully reassembled before it can be extracted. This process ultimately degrades the overall performance of the cluster. What is needed is a way to compress data along appropriate boundaries; for example, at the end of a record, where a compressed block contains a full set of records. This type of compression is sometimes referred to as *splittable compression*. There are currently two forms of splittable compression that can be used.

bzip2 (.bz2)

bzip2 is a type of compression that is suitable to use within the HDFS. The compressed blocks, in this case, are splittable and work without any special programming. However, the decompression of these files is much slower than you might normally expect, which affects system performance.

IBM LZO (.cmx)

An IBM version of the *Lempel-Ziv-Oberhumer (LZO)* data compression algorithm was created to work with HDFS. IBM LZO creates many smaller blocks of compressed data without the use of an index that is used by the open source version of LZO. To use IBM LZO with HDFS, extra coding is required. This coding is already done by IBM to provide a form of compression that is fully integrated into BigInsights and included in the BigInsights Enterprise Edition license. In comparison, the compression is only about half of what is achieved by using *compression utility A*¹, but the expense of compression and decompression is far more efficient.

Figure 4-1 shows a comparison between compression options in terms of processing usage and observed compression amounts.

	Size (Mbytes)	Comp. speed (sec)	Comp. memory used (MBytes)	De comp. speed	Decomp. memory used (Mbytes)
uncompressed	96				
compression utility A	23	10	0.7	1.3	0.5
bzip2	19	22	8	5	4
IBM-LZO	36	1	1	0.6	0
lzm	18	63	14	3	1.8

Figure 4-1 Comparison of compression options

4.4 The IBM hardware portfolio

IBM has a range of hardware that can be implemented based on performance requirements and budget. This section explores the portfolio and identify considerations to take into account when you select hardware for a BigInsights solution.

4.4.1 The IBM System x3550 M4 as a management node

Any management node in a Hadoop cluster should be implemented on premium hardware. IBM offers a server model that fits this purpose. The *IBM System x3550 M4* is a 1U server with up to 2 x Intel Sandy Bridge - EP 2.9 GHz 8-core processors. The server can be fitted with up to 768 GB of RAM over 24 DIMM slots (32 GB LR DIMM) and can support up to 8 x 2.5-inch hot swappable SAS/SATA HDDs or solid-state drives (SSDs). Powered by a maximum of two 750 W DC, this box offers an excellent performance to power ratio. And, with integrated slotless RAID, the x3550 provides a resilient architecture that is suitable for the critical applications that Hadoop runs on the management node.

¹ This name is used for instructional purposes only and is not the name of the actual utility used in this comparison.

A view of the front of the x3550 is shown in Figure 4-2. A full bill of materials can be found in Appendix A, “M4 reference architecture” on page 173.

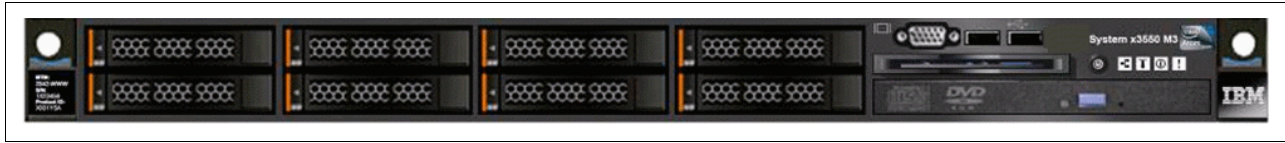


Figure 4-2 Front view of the IBM System x3550 M4

Figure 4-3 shows the benefits of using the current model over a previous generation management node.

What's new with System x3550 M4?			
	x3550 M3	x3550 M4	Benefits
Processor	<ul style="list-style-type: none"> Intel Xeon 5600 series 40 W, 60 W, 80 W, 95 W, 130 W 4-core and 6-core 	<ul style="list-style-type: none"> Intel E5-2600 series 50 W, 60 W, 70 W, 80 W, 95 W, 130 W 4-core, 6-core, 8-core 	<ul style="list-style-type: none"> Increased performance
Memory	<ul style="list-style-type: none"> 3 channels per CPU 18 slots RDIMM 12 slots UDIMM 1.5 V or 1.35 V Max 1333 MHz Max. 192 GB RDIMM Max. 48 GB UDIMM 	<ul style="list-style-type: none"> 4 channels per CPU 24 slots RDIMM 16 slots UDIMM 1.35 V memory Max 2DPC@1600 MHz (RDIMM) Max. 768 GB LRDIMM Max. 64 GB UDIMM 	<ul style="list-style-type: none"> Significant memory increase means more VMS and faster applications Greater choice of memory options
Disk	<ul style="list-style-type: none"> Up to 8 2.5" SAS/SATA with ODD (optional) 	<ul style="list-style-type: none"> 8 2.5" SAS/SATA with ODD (optional) 3 3.5" SAS/SATA 	<ul style="list-style-type: none"> Flexible storage options Increased storage capacity Improved RAID options
I/O	<ul style="list-style-type: none"> 1 x16 FHHL, 1 x16 LP Dual port Ethernet 	<ul style="list-style-type: none"> 1 CPU -1 x16 FHHL, 1 x8 LP (x16 with 2 CPU) Quad port Gigabit Ethernet Slotless 10 GbE mezz (optional) 	<ul style="list-style-type: none"> Additional I/O lanes offer 2 additional x16 slots
Power	<ul style="list-style-type: none"> 460 W, 675 W, 675 HE Redundant 	<ul style="list-style-type: none"> 550 W, 750 W, 900 W Redundant Power Platinum level 	<ul style="list-style-type: none"> Additional power supply options Great energy efficiency
Misc.	<ul style="list-style-type: none"> IMM 	<ul style="list-style-type: none"> IMM2 FoD Components 	<ul style="list-style-type: none"> Scalable management Double the connectivity Advance LPD Simple, low-cost upgrades

Figure 4-3 M3 versus M4 architecture

4.4.2 The IBM System x3630 M4 as a data node

The *IBM System x3630 M4* is a 2U base model that the BigInsights range of configurations are based upon. The x3630 M4 can be equipped with up to 2 x Sandy Bridge EN 8-core processors and 384 GB of RAM using 12 x registered dual inline memory module (RDIMM) slots. The data node has space for up to 14 x 3.5-inch SAS drives and is compatible with both 2 TB and 3 TB disks. As with the management node, a full bill of materials can be found in Appendix A, “M4 reference architecture” on page 173.

A picture of the front of the IBM System x3630 M4 is shown in Figure 4-4.



Figure 4-4 Front view of the IBM System x3630 M4

A picture of the back of the x3630 M4 can be seen in Figure 4-5

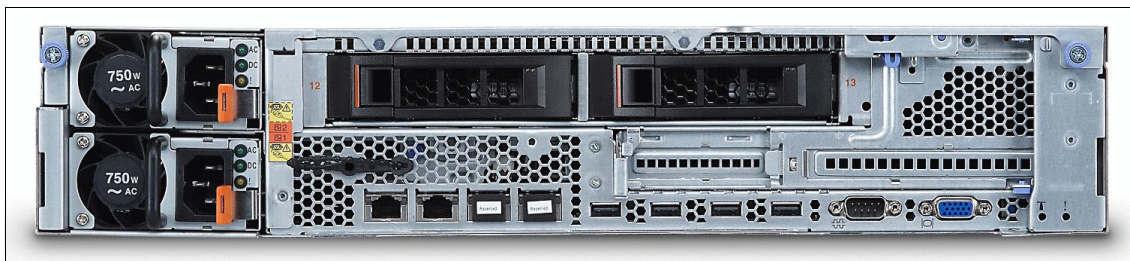


Figure 4-5 Rear view of the x3630 M4 storage-rich model which shows the extra two HDDs

A picture of an optional configuration can be seen in Figure 4-6.



Figure 4-6 Rear view of x3630 M4 without the storage-rich option. See the extra LP PCIe slot (left) and FHHL PCIe slot (right)

Figure 4-7 shows how the storage-rich M4 model compared to the previous generation of M3 data node.

x3630 M3 vs. x3630 M4 (Storage Rich Model)			
	x3630 M3	x3630 M4 Storage Rich Model	
Processor	<ul style="list-style-type: none"> X5600 up to 6 cores up to 95 W 	E5-2400 up to 8 cores up to 95 W	<i>Increased Performance</i>
Memory	<ul style="list-style-type: none"> 1333 MHz Support 12 RDIMM (1.5 V and 1.35 V) 	<ul style="list-style-type: none"> 1333 MHz /1600 MHz Support 12 RDIMM (1.5 V and 1.35 V) 	<i>Better Memory Performance</i>
Storage	<ul style="list-style-type: none"> 12+2 x3.5" Hot Swap SATA HDDs 24+4 x2.5" Hot Swap SAS/SATA HDDs Up to 42 TB disk 	<ul style="list-style-type: none"> 12+2 x3.5" Hot Swap SATA HDDs Up to 42 TB disk 	<i>Higher storage performance and reliability</i>
Expansion Slots	<ul style="list-style-type: none"> 3 PCI-e slots 2 x8 PCIe Gen II slots for 2 FH/HL on first slot 1 x4 PCIe Gen II buried slot for BR10iI 	<ul style="list-style-type: none"> 5 x PCIe 3.0 slots 1 CPU <ul style="list-style-type: none"> 2 (1 FH/FL + 1 FH/HL) or 1 x16 FH/FL 1 x4 Slotless RAID 2 CPUs <ul style="list-style-type: none"> + 2 slots (x16/x8) (1/0 or 0/2) LP 14 HDDs models support 2 PCIe slots, optional upgrade Slotless RAID from x4 to x8 with 2nd CPU installed 	<i>Flexible Options</i>
RAID	<ul style="list-style-type: none"> 6 Gbps RAID infrastructure – backplane, RAID card 	<ul style="list-style-type: none"> Slotless RAID, Up to 1 GB Flashback cache with Supercap support 	
Ethernet	<ul style="list-style-type: none"> Intel 82575 Dual 1 GbE 	<ul style="list-style-type: none"> Intel® Ethernet Controller I3504 1 Gb on board (2 std, 2 ports upgradeable via FoD) 	
Lightpath	<ul style="list-style-type: none"> Basic LED LightPath 	<ul style="list-style-type: none"> Basic LED Lightpath 	
Power Supply	<ul style="list-style-type: none"> 1+1 460 W, 675 W, 675 W HE Redundant Hot Swap power supply 	<ul style="list-style-type: none"> 1+1 750 W HE Redundant Hot Swap power supply (including Platinum 80 plus options) 	<i>Efficient PSU</i>

Figure 4-7 A comparison of the upgrades to the x3630 from M3 to M4

4.5 Lead configuration for the BigInsights management node

This section addresses the configuration of a server that is suitable for the role of a management node. We describe the architectural decisions that are made to arrive at this configuration and alternative solutions that can be reached based upon different goals.

4.5.1 Use two E5-2650, 2.0 GHz, 8-core processors in your management node

Assuming the performance is sufficient to handle the requirements of the cluster, this processor provides an excellent trade-off between price and performance. This processor comes equipped with four memory channels. In comparison, the high performance E5-2665 comes equipped with only three memory channels. Therefore, to achieve the same amount of memory, more RDIMMs are required which potentially increase the overall cost of the solution.

If better performance is required, two E5-2665, 2.4 GHz, 6-core processors can be used to raise the clock speed and increase the number of cores.

4.5.2 Memory for your management node

There are several things to take into account when you decide how to set up the memory on the x3550 M4. We address the considerations here.

Use four memory channels for each processor

To maximize the processor performance, use all memory channels that are available. In the case where the E5-2650 is being used, there are four available memory channels for each processor (and two processors within the node). Therefore, for optimum performance, populate all eight memory channels.

Use 8 GB 1333 MHz CL9 RDIMMs

When you assess which registered dual inline memory module (RDIMM) size to use, the priority is to maximize the cost-effectiveness of the solution. There are several options that can be used here, including the following configurations:

- ▶ 8 GB 1600 MHz CL11
- ▶ 8 GB 1333 MHz CL 9
- ▶ 4 GB 1600 MHz CL11
- ▶ 4 GB 1333 MHz CL 9

To *minimize the cost* while you maximize the memory, the *8 GB 1333 MHz CL 9* is the lowest cost per GB.

If a *performance upgrade* is wanted, the *8 GB 1600 MHz CL11* facilitates the full performance of the E5-2650 while it maintains the use of highly reliable memory.

Use at least 128 GB of memory for each management node

When you size how much memory that you need for the management node, an important thing to remember is that virtual memory swapping is exceptionally bad for performance in a Hadoop system. With that in mind, tend to overestimate the RAM requirements. The Java services (NameNode, JobTracker, Console, Secondary NameNode) each require several GB of RAM, as do other applications that you might have running on the management node. The amount of metadata that is stored in RAM by the NameNode service is relatively small, in comparison, and uses 1 GB of RAM per petabyte (PB) of input data (assuming a 128 MB block size). Assuming the 8 GB cards are being used, this means that 16 x 8 GB is the calculation for the total RAM within the management node where most of this memory is available to the NameNode service.

If memory swapping begins, a significant drop in performance is to be expected. If this drop occurs, you might consider increasing the amount of memory in your x3550 M4. The x3550 M4 can support up to 768 GB of RAM.

4.5.3 Dual power cables per management node

The management node requires only one power cable to run. However, the use of just one cable produces a SPOF. To minimize the risk to the cluster, use two power supplies to enhance high availability (HA). If the NameNode is on a management node that fails, the entire cluster fails. Therefore, be careful to ensure that this type of failure does not easily happen.

4.5.4 Two network adapters per management node

The minimum for Hadoop to function is one network connection that is set up on a 1 Gb network. However, this configuration produces a SPOF, which for many enterprise applications is not acceptable. To address this situation, add a second PCI Express (PCIe) adapter in addition to the 4 x 1 Gb base offering.

If the network connection performance that is wanted is greater than what can be provided by a 1 Gb network, there is the option for a performance upgrade to a 10 Gb network. However, this is a potential SPOF because it relies on one card. Therefore, consider two dual port SFP + 10 GbE adapters. If each rack has two network switches, connect the two ports of each network adapter to different switches, as described in 3.6.3, “Enterprise option” on page 44.

4.5.5 Storage controllers on the management node

RAID 1 mirroring is suitable protection for the OS and application disks. The use of ServeRAID M1115 with no cache is suitable for this purpose and can support up to eight drives.

If more storage controller cache is required, the storage controller can be upgraded to the ServeRAID M5110 with a 256 MB cache or even a 512 MB cache.

4.5.6 Hard disk drives in the management node

Hard disk drives (HDDs) are required on the management node for the OS, metadata, and other applications. To prevent the loss of data on these drives, the OS must be RAID 1 mirrored as described in the preceding section and therefore needs the use of two disks. Store the application in a RAID array; therefore, it also needs two disks. The x3550 can hold only 3 x 3.5-inch disks but can hold up to 8 x 2.5-inch HDDs. Therefore, the management node needs at least 4 x 2.5-inch disks to support the mentioned configuration.

4.6 Lead configuration for the BigInsights data node

There are many things to consider when you look at the data node. We explore these considerations in this section.

4.6.1 Processor options for the data node

The processor specification that is required depends on the performance that you require from your nodes.

Value

If you are more focused on *cost* and do not need ultimate performance from your machines, opt for two E5-2420, 1.9 GHz, 6-core processors. Choosing this configuration gives the best performance for the lowest price. The M4 provides a 44% increase in power from the previous generation of servers (M3) with no increase in price.

Performance

If the *performance* that is offered by the value processor option is not powerful enough for your business solution, the processor can be upgraded to two, E5-2450, 2.1 GHz, 8-core

processors. Although this processor has an associated increase in cost, it makes up for it with extremely good performance.

4.6.2 Memory considerations for the data node

There are several parameters to consider when you try to produce the optimal memory configuration at the lowest cost. These parameters are explored as memory considerations for the data node.

Memory channels

To maximize the processor performance, use all available memory channels. In this instance, where the E5-2400 processor is being used, there are three available memory channels for each processor (and two processors per node). Therefore, for optimum performance, populate all six memory channels.

RDIMM size

When you assess which registered dual inline memory module (RDIMM) size to use, the priority is to maximize the cost-effectiveness of the solution. There are several options that can be used here, including the following configurations:

- ▶ 8 GB 1600 MHz CL11
- ▶ 8 GB 1333 MHz CL 9
- ▶ 4 GB 1600 MHz CL11
- ▶ 4 GB 1333 MHz CL 9

To *minimize price* while you maximize memory, the *8 GB 1333 MHz CL 9* is the least expensive option per GB.

If a *performance upgrade* is wanted, the *8 GB 1600 MHz CL11* facilitates the full performance of the E5-2400 and provides highly reliable memory.

Memory size

The amount of memory in a data node is important for performance, but is not as crucial as a high amount of memory in the management node. For that reason, different configurations exist which depend on the performance that is required.

The value data node

To get the most cost-effective memory size, occupy all of the memory channels on the processor. Assuming that two E5-2420 1.9 GHz 6-core processors are being used, there is a total of six memory channels. Using the 8 GB 1333 MHz CL 9 as described in this section, will therefore give the *value data node* 48 GB of memory.

The performance data node

There is an option to increase the amount of memory if a higher level of performance is required. The *performance data node* that is offered by IBM uses 12 x 8 GB 1333 MHz CL 9 memory cards, which provide a total of 96 GB of memory per data node.

4.6.3 Other considerations for the data node

Number and type of HDDs

NL (Near Line) serial-attached SCSI (SAS) and NL Serial Advanced Technology Attachment (SATA) drives can both be used in the x3630 M4. The expected performance from both drives is similar. However, although NL SATA drives have a lower cost, NL SAS drives feature better reliability and better overall performance; therefore, use them if possible. The x3630 has 14 available HDD slots. Thus, a maximum of 14 disks can be used. Based on your storage requirements per node, consider using as many as is required. The more disks that you make available, the better that Hadoop performs (even with the same amount of total input data). The x3630 M4 supports both 2 TB and 3 TB disks drives.

Storage controller

A 6 GB performance-optimized host bus adapter (HBA) is used as the storage controller in the IBM predefined configurations. Leaving the disks configured as JBOD allows Hadoop to provide redundancy for the storage data without wasting disk space when it uses RAID.

If the data is business critical or if the replication factor is reduced and RAID is wanted for both the OS and the HDFS data, hardware RAID can be integrated as required. However, it is not part of the default configuration.

Network adapter

Because Hadoop replicates the data three times, a network HA adapter is not typically required. For this reason, when you run a 1 Gb network, use two server integrated 1 Gb ports. Or for a 10 Gb network, use one dual port SFP+ 10 GbE adapter. If HA is essential, a second network card can be used.

Power supplies

Hadoop replicates data across three data nodes, by default. Unless this configuration setting is changed, no single data node becomes a SPOF. For this reason, each data node is equipped with only one 750 W power supply.

4.6.4 Data node configuration options

A short summary of configuration options is shown in Figure 4-8.

Hadoop x3630M4 Data Node Configuration Options			
	Value Configuration	Enterprise Options	Performance Options
Processor	2 x E5-2420 1.9 GHz 6-core	2 x E5-2420 1.9 GHz 6-core	2 x E5-2430 2.2 GHz 6-core 2 x E5-2450 2.1 GHz 6-core
Memory-base	48 GB – 6 x 8 GB	48 GB – 6 x 8 GB	72 GB – 6 x 8 GB + 6 x 4 GB 96 GB – 12 x 8 GB
Disk (OS)	1 x 2 TB 3.5"	2 x 2 TB (mirrored) 3.5"	1 x 2 TB 3.5"
Disk (data)	24 TB, 12 x 2 TB NL SAS 3.5" 36 TB, 12 x 3 TB NL SAS 3.5"	24 TB, 12 x 2 TB NL SAS 3.5" 36 TB, 12 x 3 TB NL SAS 3.5"	24 TB, 12 x 2 TB NL SAS 3.5" 36 TB, 12 x 3 TB NL SAS 3.5"
HDD controller	6 Gb JBOD Controller	ServeRAID M5015	6 Gb JBOD Controller
Hardware storage protection	None (JBOD)	RAID5 11+P RAID6 10+P+Q (business critical)	None (JBOD)
User space*	6 TB w/2 TB drives 9 TB w/3 TB drives	5.5 TB w/RAID5 and 2 TB 5 TB w/RAID6 and 2 TB	6 TB w/2 TB drives 9 TB w/3 TB drives
Network	1 GbE switch w/4 10 GbE uplinks (IBM G8052)	Redundant switches	10 GbE switch w/4x 40 GbE uplinks (IBM G8264)

*Assumes 3 copies of data, uncompressed, 25% capacity reserved for map/reduce intermediate files

Figure 4-8 Data node configuration options

4.6.5 Pre-defined rack configurations

IBM has a series of offerings which consist of multiple solution entry points with a focus on value, enterprise, or performance, depending on your wanted entry point. These scalable configurations all use the x3550 M4 as a management node with two 2 GHz, 8-core processors and 128 GB of RAM. They also use four, 600 GB HDDs for the OS.

For the data nodes, these configurations use the x3630 M4. This system is also equipped with two 2.2 GHz, 6-core processors, 48 GB of RAM, two HDDs dedicated to the OS, and applications with 12 disks that are dedicated for data storage. The HDDs used are 2 TB each, by default, but can be increased to 3 TB disks if required. The configurations also use one dual-port, a 10 GbE network switch for the data network, and one dual-port 1 GbE switch for the management node, which provides a total of two switches per rack.

The full configurations are shown in Figure 4-9.

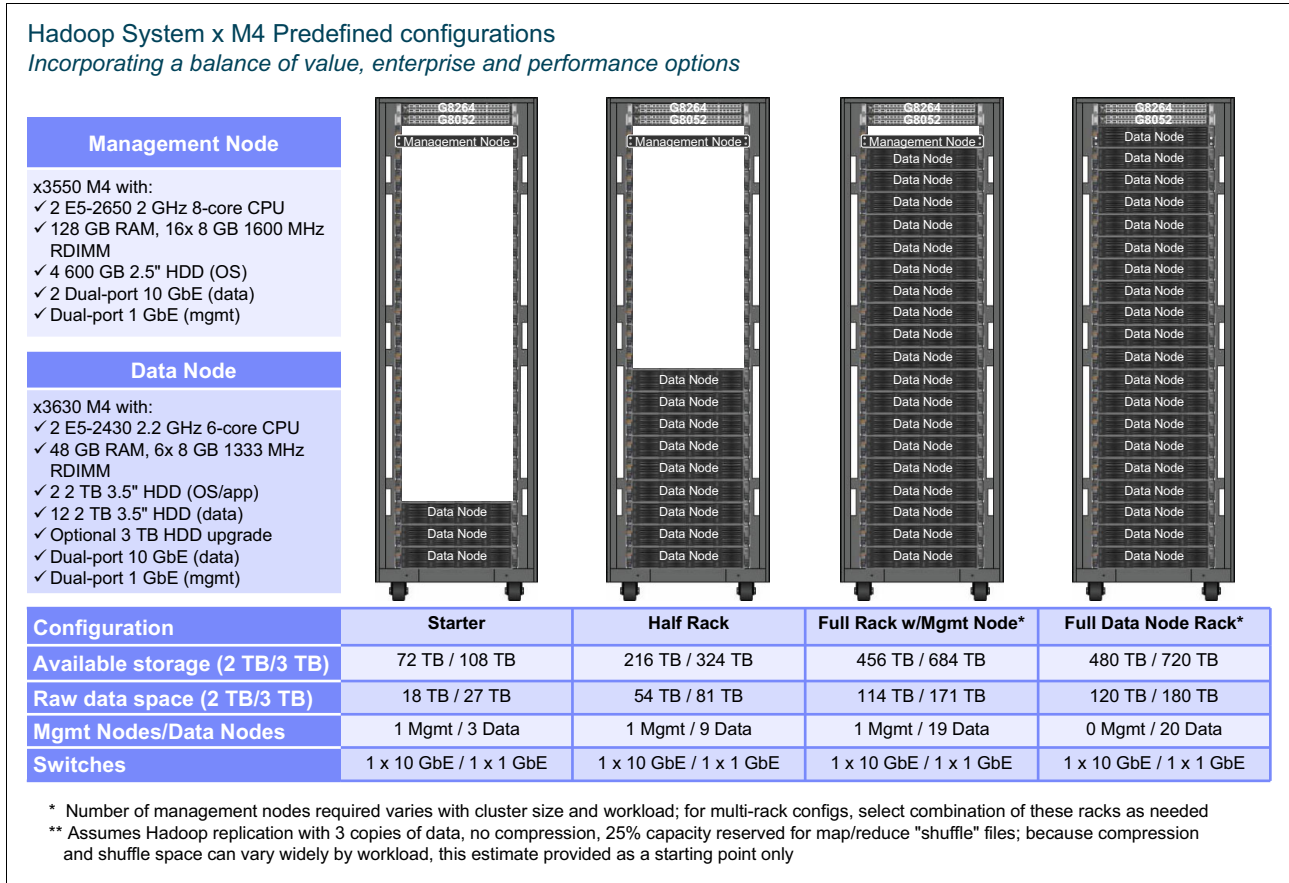


Figure 4-9 Hadoop predefined rack configurations

4.6.6 Storage considerations

The Hadoop solution is targeted to run on commodity servers with direct-attached disks. A storage area network (SAN) can be used, but it is expensive and requires much administration work.

To calculate the amount of storage capacity that you require for your cluster, the first thing to note is the *replication factor*. This factor is set at 3 by default. Assume that this is the value that we used in our calculations in Example 4-1 on page 62. The second thing to consider is that 25% of the cluster should remain available as a Hadoop-required work area. This is a generalized assumption that might vary with your workload.

Another thing to consider is the rate of data growth over time. Generally, assign roughly four times the size of your input data as your total raw storage capacity. An example is shown in Example 4-1. Our sizing suggestions do not include data compression.

Example 4-1 Storage sizing estimation example

Current Database = 300TB

Data growth = 1 TB per day

Replication factor is set to the default of 3.

Capacity needed for initial data (incl. shuffle space & assuming 0% compression) =
300 x 4 = 1200TB

Extra capacity needed for growth in a year (incl. shuffle space) =
365 x 4 = 1460TB

Total capacity required = 2660TB

This example would require four full racks with 3TB disks and would support up to 2736TB

4.6.7 Basic input/output system tool

The *IBM Advanced Settings Utility (ASU)* for IBM System x® can be used to check and modify BIOS/firmware settings. ASU settings can be changed and applied without the need for a restart. This program can be useful if you must list all of the settings from the BIOS and then compare these settings across all of your nodes. This comparison allows you to confirm that all of the nodes are set up in the same way and respond in the same manner if a problem related to the BIOS settings occurs. If the settings are not the same, they are highlighted by a **diff** command and can then be edited on a per node basis. An example of how to check the settings between two nodes is shown in Example 4-2.

A restart might be required: Changing certain settings might require a restart before they take effect.

Example 4-2 Analyzing the BIOS settings on two nodes

```
[root@bddn20 logs]# ./asu64 show all > bddn20.txt
```

```
[root@bddn20 logs]# ssh bddn21
```

```
[root@bddn21 logs]# ./asu64 show all > bddn21.txt
```

```
[root@bddn21 logs]# exit
```

```
[root@bddn20 logs]# scp root@bddn21:/home/bddn21.txt /PATH/to/bddn21.txt
```

```
[root@bddn20 logs]# diff bddn20.txt bddn21.txt > diff.txt
```

asu64 command: The **asu64** command must be entered in the directory in which it is installed (by default, this directory is set to `/opt/ibm/toolscenter/asu/`), unless it is added to the **PATH** variable.

As Example 4-3 shows, the diff file output can be long, without any significant details being different. One might expect that all of the IP and MAC addresses would be different. However, this example also shows that the boot order is different (in bold), which might be significant for your cluster if something goes wrong.

Example 4-3 Diff file showing the different BIOS settings over two nodes

```
27c27
< IMM.IMMInfo_Name=SN# KQ6R984
---
> IMM.IMMInfo_Name=SN# KQ6R976
88,89c88,89
< IMM.HostName1=bddn20m
< IMM.HostIPAddress1=129.40.109.20
---
> IMM.HostName1=bddn21m
> IMM.HostIPAddress1=129.40.109.21
93,94c93,94
< IMM.DHCPAssignedHostname=bddn20m
< IMM.DHCPAssignedHostIP1=129.40.109.20
---
> IMM.DHCPAssignedHostname=bddn21m
> IMM.DHCPAssignedHostIP1=129.40.109.21
117c117

.
.
.

< PXE.NicPortPxeMode.10=Legacy Support
194,198d186
< PXE.NicPortPxeMode.5=Legacy Support
< PXE.NicPortPxeMode.6=Legacy Support
< PXE.NicPortPxeMode.7=Legacy Support
< PXE.NicPortPxeMode.8=Legacy Support
< PXE.NicPortPxeMode.9=Legacy Support
200d187
< PXE.NicPortPxeProtocol.10=IPv4
204,213c191
< PXE.NicPortPxeProtocol.5=IPv4
< PXE.NicPortPxeProtocol.6=IPv4
< PXE.NicPortPxeProtocol.7=IPv4
< PXE.NicPortPxeProtocol.8=IPv4
< PXE.NicPortPxeProtocol.9=IPv4
< iSCSI.MacAddress.1=34-40-B5-A3-51-D8
< iSCSI.MacAddress.2=34-40-B5-A3-51-D9
< iSCSI.MacAddress.3=00-00-C9-F5-EA-D8
< iSCSI.MacAddress.4=00-00-C9-F5-EA-DC
< BootOrder.BootOrder=Legacy Only=CD/DVD Rom=Hard Disk 0=Floppy Disk=PXENetwork
---
> BootOrder.BootOrder=Legacy Only=CD/DVD Rom=Hard Disk 0=PXENetwork
325c303
> iSCSI.MacAddress.1=34-40-B5-A3-65-E8
> iSCSI.MacAddress.2=34-40-B5-A3-65-E9
> iSCSI.MacAddress.3=00-00-C9-F5-C7-F4
```

```
> iSCSI.MacAddress.4=00-00-C9-F5-C7-F8
```

The BIOS tool can now correct this setting to whichever setup that you deem to be correct, by using the **asu64 set** command. See Example 4-4.

Example 4-4 Correcting a BIOS setting by using the asu command

```
[root@bddn20 asu]# ./asu64 set BootOrder.BootOrder "Legacy Only=CD/DVD Rom=Floppy Disk=Hard Disk 0=PXE Network"
```

```
IBM Advanced Settings Utility version 9.21.78C
Licensed Materials - Property of IBM
(C) Copyright IBM Corp. 2007-2012 All Rights Reserved
Successfully discovered the IMM via SLP.
Discovered IMM at IP address 169.254.95.118
Connected to IMM at IP address 169.254.95.118
BootOrder.BootOrder=Legacy Only=CD/DVD Rom=Floppy Disk=Hard Disk 0=PXE Network
Waiting for command completion status.
Command completed successfully.
```

In Example 4-4, we changed the bddn20 boot order to be the same as the bddn21 boot order. Many of the BIOS settings can be edited in this manner.

ASU functions: The full functionality of the ASU can be explored in the User's Guide: http://publib.boulder.ibm.com/infocenter/toolctr/v1r0/topic/asu/asu_guide.pdf



Operating system prerequisites for BigInsights

So far, you have been taken through a chronological set of information regarding the setup of your BigInsights cluster. In Chapter 3, “BigInsights network architecture” on page 29, we first worked with you on your logical and physical network considerations. Then, in Chapter 4, “BigInsights hardware architecture” on page 49, we helped you with your potential hardware selections. Now, we assume that your hardware is powered up, that you set your basic input/output system (BIOS) settings, and you selected and installed your operating system (OS).

This chapter covers a few of the prerequisites, settings, and OS level settings to become familiar with as you prepare for the next chapter, where you install BigInsights.

5.1 Prerequisite software

Prerequisites are required for many software solutions and BigInsights is no different. For each release of BigInsights, we update the installation requirements section of the online documentation. To see the current requirements for *BigInsights V1.4*, you can view the web page that is shown in Example 5-1.

Example 5-1 BigInsights installation requirements URL

<http://pic.dhe.ibm.com/infocenter/bigins/v1r4/index.jsp?topic=%2Fcom.ibm.swg.im.infosphere.biginsights.install.doc%2Fdoc%2Fc0057608.html>

This chapter focuses on items that we believe you should be aware of regarding the OS level of each node on your cluster.

Example cluster: As a reminder, we selected Red Hat Enterprise Linux 6 update 2 for the example cluster that we used at time of writing. As such, most examples are specific to that selection. Therefore, you might need to use slightly different actions on your machines, based on your selection of the OS and hardware.

5.1.1 Operating provisioning software

For small test clusters, you might choose to manually install OS software and perform the manual configurations that are required to efficiently run BigInsights. For clusters with a larger number of nodes, it is advisable to use OS provisioning software to an installation of a network-based OS. Although a detailed description of enabling this software is beyond the scope of this book, we believe that it is a good practice to use a provisioning tool. This tool enables the images across the cluster to be the same regarding many of the settings that follow. We previously used the following products for provisioning software:

- ▶ XCAT
- ▶ Cobbler

Both products use the Red Hat *kickstart file* that is used to define items that range from the packages that are installed on the OS to the partitioning and file system layout of the hard disk drives (HDDs).

5.1.2 Yellowdog Updater Modified repository

This is a repository of installation programs which makes updating and extending your systems' OS and software environment easier. If your OS is installed correctly, the installation probably requested your login credentials to connect to the YUM repository. When we set up our cluster, having the Yellowdog Updater Modified (YUM) repository was a real time saver. In appendix , "Non-package-related items" on page 191, we provide an example of the repository items that we believe you should be aware of. If you plan to use the checklist in the appendix as part of your installation process, you see the details that are shown there.

5.1.3 Operating system packages

After you install the OS and set up your YUM repository, you will want to ensure that all prerequisite packages are installed. Some of these packages are required, though others make your environment easier to maintain. For a complete list of the packages we installed on the systems in our cluster, see 2.3, "What is BigInsights?" on page 18.

5.2 Operating system settings related to software

At this point, everything that must be installed before BigInsights, needs to be installed. Now we look at some of the settings at the OS level that are beneficial to understand and might benefit the health and performance of your cluster.

5.2.1 System clock synchronization

Often times, when you must research an issue that occurred in a multi-machine environment (such as the cluster you are building), it is helpful to have the system clocks synchronized to the exact same time of day. In this way, you can look at the sequence of events across the systems and determine which happened in which sequence.

When we set up our cluster, we used the Network Time Protocol (NTP) service that is provided by the OS. By running a set of instructions similar to the ones shown in Example 5-2, your nodes should all report the same time of day.

Example 5-2 Example ntp script file

```
# Install and configure ntp
service ntpd stop
ntpdate -u 0.rhel.pool.ntp.org
chkconfig ntpd on
service ntpd start
```

5.2.2 Services to disable for improved performance

Within the OS, you might find that certain services and processes are running by default. Some of these functions are not currently required and might generate more processing usage that might adversely affect the performance of BigInsights. Here are a few of the settings we changed on our cluster as root when being logged in to each node:

- ▶ disable selinux
- ▶ disable ipv6
- ▶ edit `/etc/sysctl.conf` and set `vm.swappiness = 5`

You can learn more about selinux at <http://www.nsa.gov/selinux>. You can also learn more about `ipv6` by using the `man ipv6` command on one your nodes. Lastly, to better understand the `swappiness` setting, you might want to do an Internet search on the topic. To simplify, the lower the number, the less likely the OS performs memory page swapping to disk. Hadoop and BigInsights Java virtual machines (JVMs) are designed to run in main memory. It is important to ensure that little, if any, swapping is occurring. If swapping is detected, it is better to reduce the number of map and reduce task slots that are assigned to the data node, or add more memory to the system. If swapping is occurring in the management node, similar actions, such as reducing the amount of JVM memory usage or adding memory, might be actions for you to consider to eliminate swapping.

5.2.3 Raising the `ulimits` setting to accommodate Hadoop's data processing within BigInsights

Every OS has a limit for the number of files it can have "open" at any one time. In our Red Hat systems, the `ulimit` parameter is the max setting. If you were to run the `ulimit -a` command, you see all of the details for the current session of the user ID that issued the command.

Typically, the default amount that is assigned to ulimit is too low for Hadoop to work properly under a larger workload.

Therefore, increase the ulimit to avoid errors such as “too many open files”. Because the BigInsights installation creates a user ID called *biadmin*, we can prepare for Hadoop jobs to be run as this user ID by raising its ulimit. On our cluster, we ran the commands that are shown in Example 5-3 as a preparation step.

Example 5-3 Commands for setting higher ulimits for Hadoop

```
echo "biadmin hard nfile 16384" >> /etc/security/limits.conf
echo "biadmin soft nfile 16384" >> /etc/security/limits.conf
echo "biadmin hard nproc 32000" >> /etc/security/limits.conf
echo "biadmin soft nproc 32000" >> /etc/security/limits.conf
```

5.2.4 Optional: set up password-less Secure Shell

The BigInsights installation requires you to have “write” and “run” scripts privileges on all nodes in your cluster. Users authenticate into the InfoSphere BigInsights web console to do most of their BigInsights tasks. Click the URL shown in Example 5-4 to review the supported user ID and permissions configurations.

Example 5-4 Setting up users and permissions

http://pic.dhe.ibm.com/infocenter/bigins/v1r4/index.jsp?topic=%2Fcom.ibm.swg.im.infosphere.biginsights.install.doc%2Fdoc%2Finstall_users_permissions.html&resultof%3D%2522%2573%2573%2568%2522%2520

After you review the different options that are shown under the URL in Example 5-4, you can see that setting up password-less Secure Shell (SSH) before the installation, is optional. There are many options that are based on the requirements within your data center for the nodes in your cluster.

When we set up our cluster, we chose to set up password-less SSH by using root to ensure that all nodes can be accessed by SSH without a password from the management node. We then opted to select the BigInsights installation option that allowed us to tell the installation that password-less SSH was already set up. We saw the installation complete more smoothly by taking this path.

SSH and cluster security: If cluster security is required, then do not configure password-less SSH.

5.3 Optionally configure /etc/hosts

If a private network was set up using the guidelines that are defined by RFC 1918, it can be advisable to configure an */etc/hosts* file with host name entries that can resolve to the private network. This can make some configurations easier to read. On a small cluster, we used a class C IP address, as shown in Example 5-5.

Example 5-5 Sample /etc/hosts for a small cluster

```
127.0.0.1 localhost localhost.localdomain
::1 localhost localhost.localdomain
192.168.2.60 bddn20p
```

```
192.168.2.61    bddn21p
192.168.2.62    bddn22p
192.168.2.63    bddn23p
192.168.2.64    bddn24p
192.168.2.65    bddn25p
192.168.2.66    bddn26p
192.168.2.67    bddn27p
192.168.2.68    bddn28p
192.168.2.115 bdmn05p
```

5.4 Operating system settings related to hardware

Depending on the redundancy that is required in your cluster at the node level, you might need to do some additional configuration commands to properly set up your network on the nodes (because you should have already done the equivalent within the network switches in the network side). Depending on how many drives you have in each of your nodes, you might also want to verify that they are configured properly at the OS level before moving on to Chapter 6, “BigInsights installation” on page 73.

5.4.1 Operating system level settings if optional network cards were added

Based on your hardware configuration within each node and the prior setup that you performed in the network switch before connecting the nodes, you might want to configure your networking options on each node now.

There is a networking concept that is known as bonding. It might also be referred to as network bonding, channel bonding, NIC Bonding, or NIC Teaming. You must first understand that the most common failure within a network occurs at the cable level. When a cable fails, data is no longer able to travel across the wire, and the “port” on the node seems to be malfunctioning. If bonding is not in place, no network traffic is able to flow.

Bonding this traffic to two ports allows for traffic to continue to flow even if one of the two bonded ports’ wires fails. Removing this single point of failure (SPOF) is accomplished by bonding, which requires more configuration at the OS level. Figure 5-1 shows what port bonding looks like.

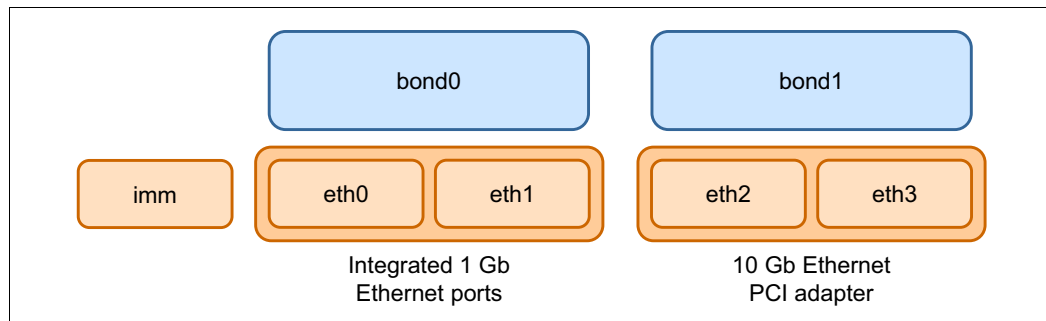


Figure 5-1 How port bonding works

Benefits of port bonding: Another advantage to bonding network ports together is more network throughput and data transfer rates under certain conditions. Your work in this area might have more than one benefit.

Three-zone network: The details that are provided in the remainder of section 5.4.1, “Operating system level settings if optional network cards were added” on page 69 are based on the selection of a *three-zone network* as described in 3.3, “Networking zones” on page 32.

Non-enterprise option: value configuration

Whether you choose to share your management network with your private and administration networks, you still must bond your private and administration networks together to take advantage of network bonding. To save you time in doing the bonding exercise, Example 5-6 shows a method of how you can bond with a value configuration. This configuration is specific to *Red Hat 6*. Red Hat uses the configuration files in `/etc/sysconfig/networking-scripts` to set up the networking ports. The file names are used to determine physical versus bonded ports. For more information, see the URL in example Example 5-6.

Example 5-6 Bonding two 1 Gbps ports within a value configuration node

Reference URL

https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/s2-networkscripts-interfaces-chan.html

Physical Ports = eth0, eth1
Bonded Port = bond0

Contents of `ifcfg-eth0` or `ifcfg-eth1`
DEVICE=ethN (replace N with the interface number)
HWADDR=00:00:00:00:00:00 (replace with your network card MAC address)
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
MASTER=bond0
SLAVE=yes

Contents of `ifcfg-bond0`
DEVICE=bond0
IPADDR=(insert your IP address here)
NETMASK=(insert your netmask here)
GATEWAY=(insert your gateway here)
USERCTL=no
BOOTPROTO=none
ONBOOT=yes
BONDING_OPTS="miimon=100 mode=4 xmit_hash_policy=layer2+3"

Additionally, make sure you add a `bonding.conf` file in `/etc/modprobe.d`

Non-enterprise option: performance configuration

If you chose to use the non-enterprise performance configuration for your cluster, you added one 10 Gbps dual-port network card to each node. You have the option of bonding those two 10 Gbps ports to carry administration and data across the two ports. Refer to the bonding example in Example 5-6 to see how this process is done. A new `bond1` with `eth2` and `eth3` can also be used.

Enterprise option selected: value or performance configuration

Within the enterprise option for networking, the node configuration changes to include a second redundant switch and a second redundant network card within each node. In the preceding section, we described an example of how to bond. For the sake of brevity, we do not provide an example here because it should be similar to the one already provided.

Sharing the administration network: If you choose to share the administration network with your private network, more configuration work might have to be done.

5.4.2 Storage configuration

Hadoop Distributed File System (HDFS) is used by the cluster to enable a global shared nothing file system across the data nodes. HDFS runs atop already installed file systems. The exact layout and partitioning of the system is important to enable maximum parallelism across the cluster.

Formatting file systems

When you format the HDDs in each node that is used for data, use the ext3 or ext4 file system. Also consider the use of the `-noatime` option. This option improves the file read performance because file system *reads* typically *write* file system metadata about the last access time and date of the file read operation.

Determining whether to mirror operating system drives

The concept of using a mirrored set of drives within a node is based on the cost of resiliency. If you want the node to continue running when an OS HDD fails, you have to mirror the OS drive to a minimum of one or more drives. The additional drives add costs because their storage is used to hold an exact copy of the data from its mirrored partner drive. However, if you want to save costs and are willing to rely on HDFS to avoid data loss during a node failure, you can choose to not mirror your OS.

Within our cluster, we decided not to mirror the OS drives in the data nodes. We also did not want to lose the entire cluster if a HDD failed in the management node. To prevent this SPOF, we chose to mirror the OS drives within the management node by using hardware *Redundant Array of Independent Disks (RAID)* supported by our storage adapter.

Ensuring JBOD is used on your data nodes

In 4.3, “Storage and adapters used in the hardware architecture” on page 51, we described the design considerations behind the selection of *just a bunch of disks (JBOD)*. We now look at what is required to ensure that your systems are configured correctly to take advantage of this option.

We now provide example commands that help you learn more about your data nodes and the quantity of drives that are included in each node. The installation runs more quickly (because of less configuration for any of the nodes that are not identical to the rest) if all data nodes have the same number of drives. Although, having the same number is not a requirement.

Our cluster used IBM System x3630 for data nodes. To determine if a node has the correct JBOD configuration, the command and output that is shown in Example 5-7, displays the wanted result.

Example 5-7 JBOD verification command

```
[root@bddn20 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       193G  3.5G  180G   2% /
tmpfs           48G   0    48G   0% /dev/shm
/dev/sda1       124M  32M   87M  27% /boot
/dev/sda4       2.5T  203M  2.3T   1% /hadoop/sda
/dev/sdb1       2.7T  201M  2.6T   1% /hadoop/sdb
/dev/sdc1       2.7T  201M  2.6T   1% /hadoop/sdc
/dev/sdd1       2.7T  201M  2.6T   1% /hadoop/sdd
/dev/sde1       2.7T  201M  2.6T   1% /hadoop/sde
/dev/sdf1       2.7T  201M  2.6T   1% /hadoop/sdf
/dev/sg1        2.7T  201M  2.6T   1% /hadoop/sg
/dev/sdh1       2.7T  201M  2.6T   1% /hadoop/sdh
/dev/sdi1       2.7T  201M  2.6T   1% /hadoop/sdi
/dev/sdj1       2.7T  201M  2.6T   1% /hadoop/sdj
/dev/sdk1       2.7T  201M  2.6T   1% /hadoop/sdk
/dev/sdl1       2.7T  201M  2.6T   1% /hadoop/sdl
/dev/sdm1       2.7T  201M  2.6T   1% /hadoop/sdm
/dev/sdn1       2.7T  201M  2.6T   1% /hadoop/sdn
```

Example 5-8 shows all 14 drives within the node as independent devices. If these were together in a RAID grouping, you do not see the individual drives that are shown as a result of the **df -h** command.

To prepare for your BigInsights installation, you have to assign the following two sections of the installation to a directory on each of the drives. For our cluster, we provided the following list of folders for the installation to process and configure.

Example 5-8 MapReduce local directories

```
/hadoop/sdb/mapred/local,/hadoop/sdc/mapred/local,/hadoop/sdd/mapred/local,/hadoop/sde/mapred/local,/hadoop/sdf/mapred/local,/hadoop/sg/mapred/local,/hadoop/sdh/mapred/local,/hadoop/sdi/mapred/local,/hadoop/sdj/mapred/local,/hadoop/sdk/mapred/local,/hadoop/sdl/mapred/local,/hadoop/sdm/mapred/local,/hadoop/sdn/mapred/local
```

Example 5-9 shows an example for where HDFS places the data.

Example 5-9 HDFS data directories

```
/hadoop/sdb/hdfs,/hadoop/sdc/hdfs,/hadoop/sdd/hdfs,/hadoop/sde/hdfs,/hadoop/sdf/hdfs,/hadoop/sg/hdfs,/hadoop/sdh/hdfs,/hadoop/sdi/hdfs,/hadoop/sdj/hdfs,/hadoop/sdk/hdfs,/hadoop/sdl/hdfs,/hadoop/sdm/hdfs,/hadoop/sdn/hdfs
```

The examples that are shown here are potentially different on your cluster. We use these settings in 6.2, “Installing BigInsights using the graphical user interface” on page 74 as part of the installation. To prepare for installation, you might want to research the equivalent settings on your systems at this time.



BigInsights installation

BigInsights provides a web-based graphical user interface (GUI) that installs and configures your selected features and also displays the details of the progress of the installation. When the installation process is complete, you as an administrator, can check the status of the *BigInsights* components by using a web-based management console. Through this console, you can start and stop components, add or remove nodes, track MapReduce jobs statuses, analyze log records and the overall system health, view the contents of the distributed file system, and other numerous possibilities.

Before you continue: Ensure that you understand all the hardware and software requirements that are described in Chapter 4, “BigInsights hardware architecture” on page 49, and 3.8.2, “Power and cooling” on page 47. In addition, we encourage you to access the link to learn more about the hardware and prerequisites:

<http://pic.dhe.ibm.com/infocenter/bigins/v1r4/topic/com.ibm.swg.im.infosphere.biginsights.install.doc/doc/c0057608.html>

In addition, be aware that the installation might not run correctly if the language of your system is set to anything other than English (US).

6.1 Preparing the environment for installation

The IBM InfoSphere BigInsights installation program supports new installations, overlays of existing installs, and upgrades by using the installation console or by doing a silent installation.

Before you install BigInsights, the installation program completes the following functions:

- ▶ Ensures that the node from where you start the installation is part of your cluster.
- ▶ Ensures that port number *8300* is not currently being used so that it can be assigned temporarily to the installation console.

For your reference, Appendix B, “Installation values” on page 179 contains default installation options and values that are provided by BigInsights.

6.2 Installing BigInsights using the graphical user interface

There are many ways to configure BigInsights. This section describes how to install a new cluster for the first time by using the web-based installation feature of *BigInsights Version 1.4*. In addition, it shows you the steps that are required to run a BigInsights environment. To install the BigInsights product using the web-based option, we used the following procedure:

1. Log in to your Linux operating system (OS) using your *root ID*, as shown in Example 6-1:

Example 6-1 Log in as root

```
$ ssh root@bdmn05.pbm.ihost.com
root@bdmn05.pbm.ihost.com's password:
Last login: Tue Sep  4 09:16:32 2012 from bdvm06.pbm.ihost.com
[root@bdmn05 ~]#
```

2. Go to the directory where you placed the IBM InfoSphere BigInsights tar file and expand it, as shown in Example 6-2:

Example 6-2 Expand the tar file

```
[root@bdmn05 ~]# cd /home
[root@bdmn05 home]# tar -xvf biginsights14.tar
```

3. Go to the directory where you just expanded the BigInsights tar file and run the **start.sh** script, as shown in Example 6-3:

Example 6-3 Start the installation program

```
[root@bdmn05 home]# cd biginsights-enterprise-linux64_b20120604_2018
[root@bdmn05 biginsights-enterprise-linux64_b20120604_2018]# ./start.sh

Extracting Java ....
Java extraction complete, using
JAVA_HOME=/home/biginsights-enterprise-linux64_b20120604_2018/_jvm/ibm-java-x86
_64-60
Verifying port 8300 availability
port 8300 available
Starting BigInsights Installer
....
```

The `start.sh` script starts an instance of the WebSphere Application Server Community Edition on port `8300`. The script provides a URL to the installation wizard, which is available at: `http://<your-server>:8300/Install/` where `<your-server>` is the server on which you ran the `start` command.

URL of installation wizard: Sometimes, the URL of the installation wizard might not be displayed after the installer starts. With a web browser, you can still access the URL `http://<your-server>:8300/Install/` and continue the installation process.

4. In the welcome panel, which is shown in Figure 6-1, click **Next**.

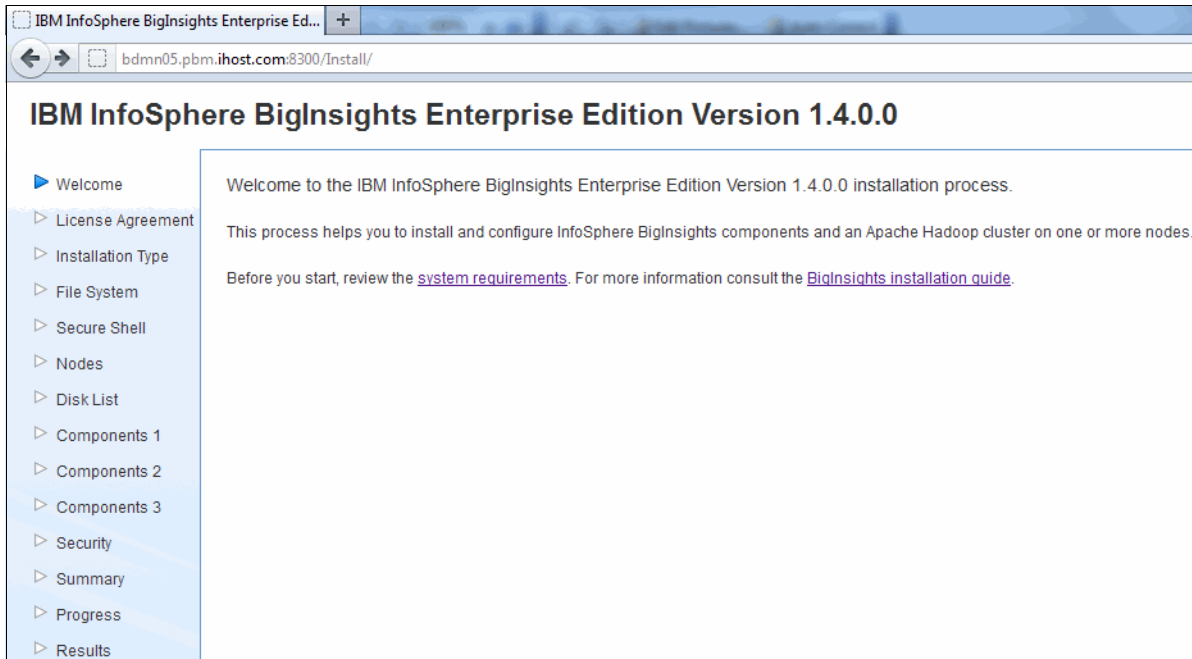


Figure 6-1 *BigInsights welcome panel*

5. If acceptable to you, accept the *terms in the license agreement*, and click **Next**. See Figure 6-2.

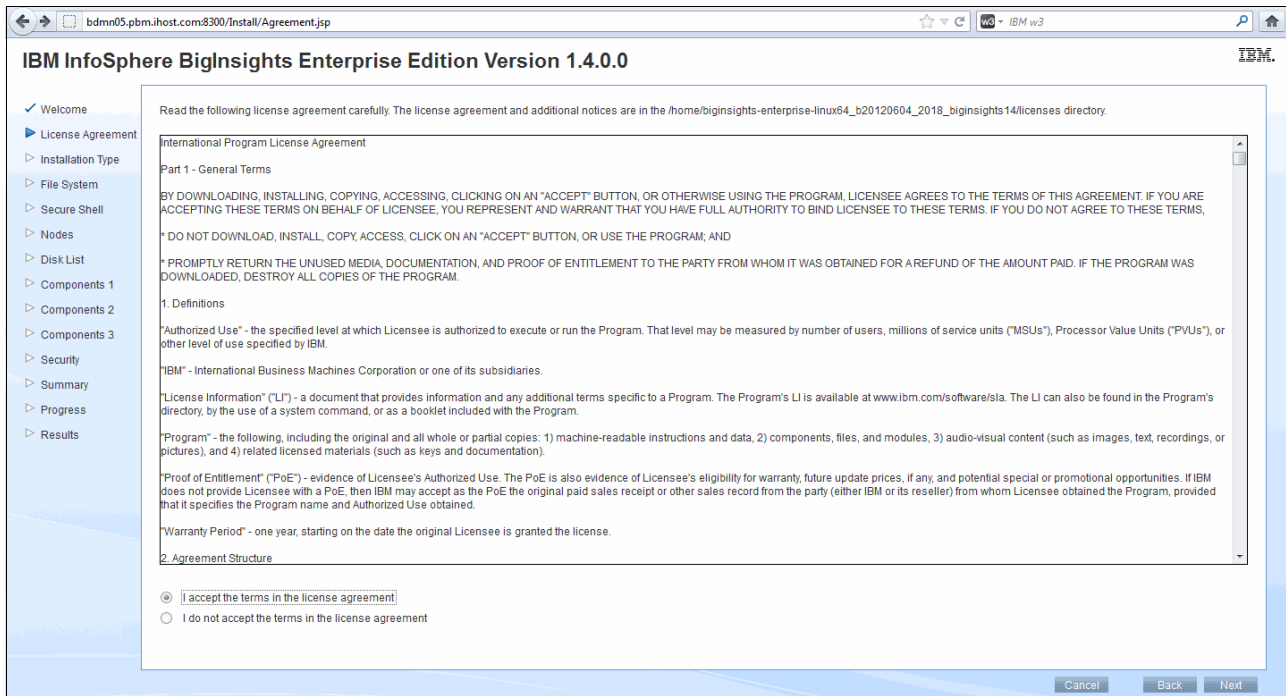


Figure 6-2 BigInsights license agreement terms

6. For the installation type, we chose **Cluster installation**. Click **Next**. See Figure 6-3.

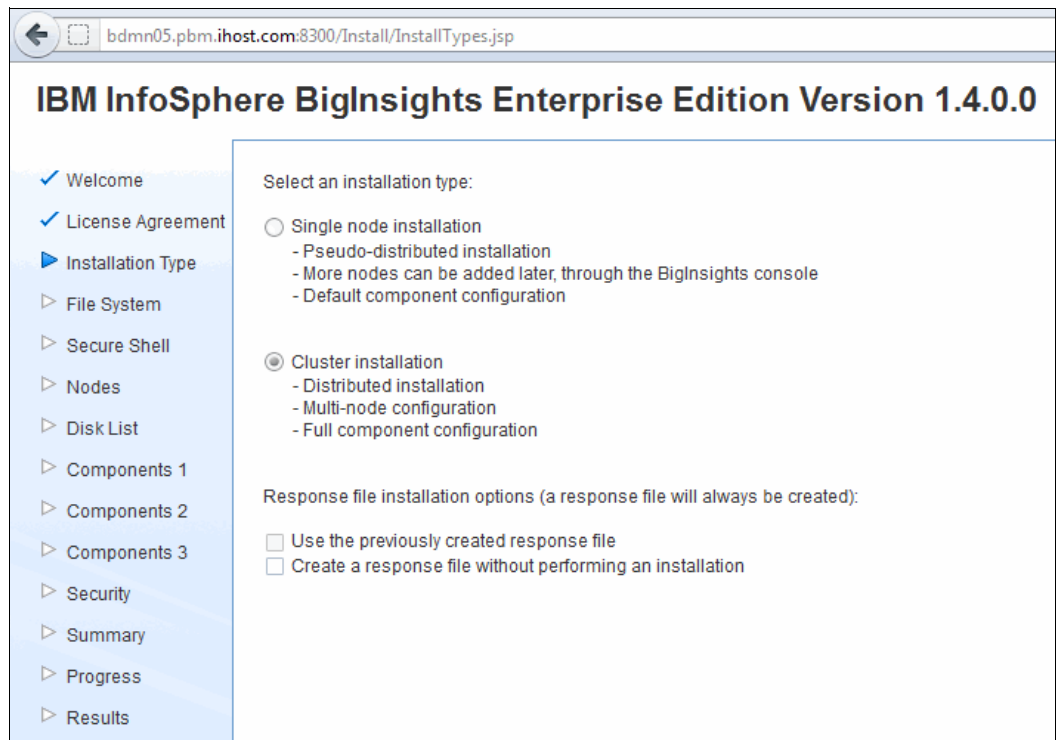


Figure 6-3 Cluster installation type

7. Leave the installed Hadoop Distributed File System (HDFS) enabled. In our case, we expand the MapReduce general settings and set the Cache directory to this value: `/hadoop/sdb/mapred/local,/hadoop/sdc/mapred/local,/hadoop/sdd/mapred/local,/hadoop/sde/mapred/local,/hadoop/sdf/mapred/local,/hadoop/sdg/mapred/local,/hadoop/sdh/mapred/local,/hadoop/sdi/mapred/local,/hadoop/sdj/mapred/local,/hadoop/sdk/mapred/local,/hadoop/sdl/mapred/local,/hadoop/sdm/mapred/local,/hadoop/sdn/mapred/local`. Click **Next**.

Figure 6-4 shows an example of how to set the cache directory.

The screenshot shows the 'File System' configuration page for IBM InfoSphere BigInsights Enterprise Edition Version 1.4.0.0. The left sidebar lists installation steps: Welcome, License Agreement, Installation Type, File System (selected), Secure Shell, Nodes, Disk List, Components 1-3, Security, Summary, Progress, and Results. The main content area is divided into sections:

- Specify the BigInsights installation directories.**
 - Overwrite existing files and directories
 - Specify the path to a directory that the current user can access. [?](#)
 - * BigInsights installation root directory:
 - Specify the BigInsights installation directories. [?](#)
 - * BigInsights installation directory:
If specified as relative path (without a leading /), this directory will be appended to /.
 - Specify the BigInsights data/log directory.
 - * BigInsights data/log directory:
If specified as relative path (without a leading /), this directory will be appended to /.
- MapReduce general settings**
 - Specify the local file system paths where temporary MapReduce data will be written on each node.
 - * Cache directory:
Separate multiple paths with a comma. Each path specified as relative path (without a leading /), will be appended to /.
 - Specify the local file system path where Hadoop log files will be written on each node.
 - * Log directory:
If specified as relative path (without a leading /), this directory will be appended to /.
 - Specify the HDFS path where MapReduce stores system files.
 - * Map/Reduce system directory:

Figure 6-4 Setting the cache directory

Defining your directories: For a complete description of how to properly define the directories for your cluster, see 5.4.2, “Storage configuration” on page 71.

- See Figure 6-5. Select **Use the root user to make any necessary configuration changes** and type the **Root password**, **BigInsights administrator user ID**, and **BigInsights administrator password**. Then, **Confirm BigInsights administrator password**, and **BigInsights administrator group ID**, click **Next**.

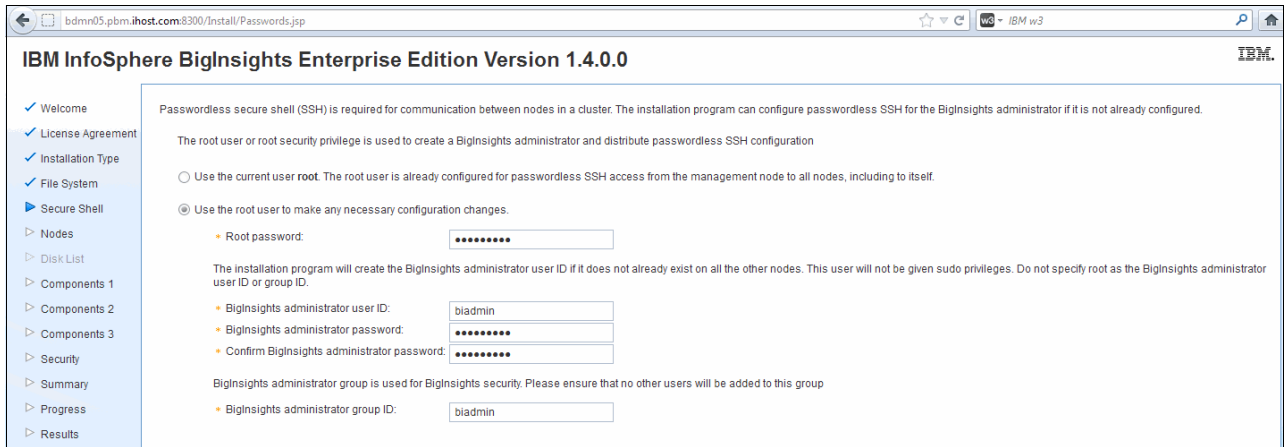


Figure 6-5 Setting root password, administrator user ID and password, administrator group ID

Default user ID: The default administrator user ID and administrator group ID for BigInsights installation is *biadmin*.

- By default, the management node should already be set, as shown in Figure 6-6. Click **Add Multiple Nodes**.

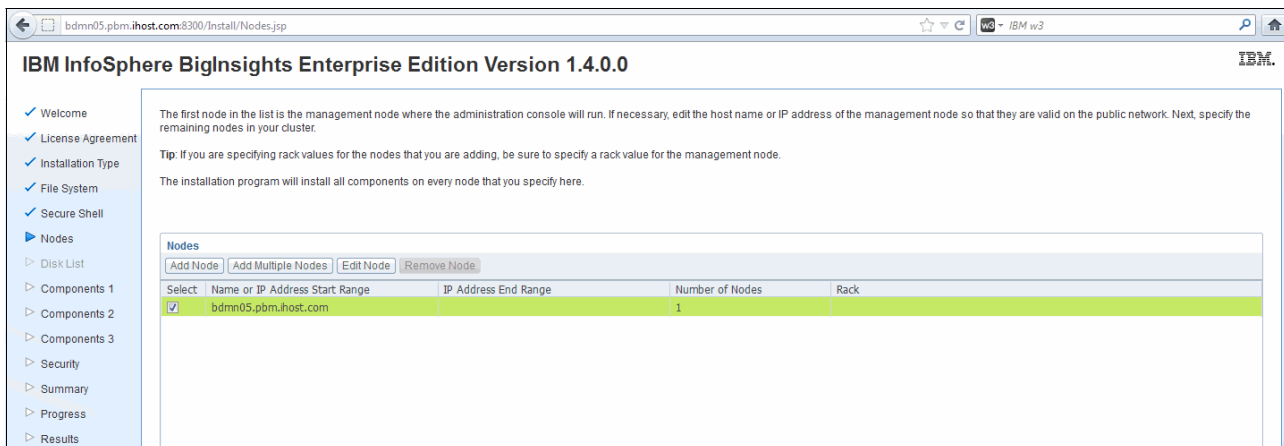


Figure 6-6 Management node

Adding a node: If the management node is not displayed in the initial list, add it by clicking *Add Node*.

10. For the *Starting IP address of the range* when we built our cluster, we used *192.168.2.61*; for *Number of nodes*, we used *9*; we typed the *Root password*, and clicked **OK**. Your cluster might require different settings than the ones that we displayed here, especially if you do not use the three-network-zone approach that is described in 3.3, “Networking zones” on page 32.

Figure 6-7 shows an example of adding multiple data nodes.

Figure 6-7 Adding multiple data nodes

11. A list of *Nodes* are presented, as shown in Figure 6-8.

Nodes					
<input type="button" value="Add Node"/> <input type="button" value="Add Multiple Nodes"/> <input type="button" value="Edit Node"/> <input type="button" value="Remove Node"/>					
Select	Name or IP Address	Start Range	IP Address End Range	Number of Nodes	Rack
<input checked="" type="checkbox"/>	bdmn05.pbm.ihost.com			1	
<input type="checkbox"/>	192.168.2.61		192.168.2.69	9	

Figure 6-8 List of Nodes

12. Click **Next**.

13. Leave the default settings for *Configure the nodes and ports for the BigInsights components*, and click **Next**. Figure 6-9 shows an example of this window.

bdmn05.pbm.ihost.com:8300/Install/Components1.jsp

IBM InfoSphere BigInsights Enterprise Edition Version 1.4.0.0

- ✓ Welcome
- ✓ License Agreement
- ✓ Installation Type
- ✓ File System
- ✓ Secure Shell
- ✓ Nodes
- ▷ Disk List
- ▶ Components 1
- ▷ Components 2
- ▷ Components 3
- ▷ Security
- ▷ Summary
- ▷ Progress
- ▷ Results

Configure the nodes and ports for the BigInsights components.

* BigInsights console node: 192.168.2.60

Use https for BigInsights console

* BigInsights console port: 8080

Configure Jaql UDF server

* Jaql UDF server node: bdmn05.pbm.ihost.com [Assign...](#)

* Jaql UDF server port: 8200

* Derby node: bdmn05.pbm.ihost.com [Assign...](#)

* Derby port: 1528

* BigInsights orchestrator node: bdmn05.pbm.ihost.com [Assign...](#)

* BigInsights orchestrator port: 8888

Figure 6-9 Configure the nodes and ports for the BigInsights options

14. Under **DataNode/TaskTracker** (see Figure 6-10), expand **Advanced settings** and set the Data directory to this value:

/hadoop/sdb/hdfs,/hadoop/sdc/hdfs,/hadoop/sdd/hdfs,/hadoop/sde/hdfs,/hadoop/sdf/hdfs,/hadoop/sdg/hdfs,/hadoop/sdh/hdfs,/hadoop/sdi/hdfs,/hadoop/sdj/hdfs,/hadoop/sdk/hdfs,/hadoop/sdl/hdfs,/hadoop/sdm/hdfs,/hadoop/sdn/hdfs

Define your directories: For a complete description of how to properly define the directories for your cluster, see section 5.4.2, “Storage configuration” on page 71.

15. Click **Next**.

The screenshot shows the 'Components settings' page in the IBM InfoSphere BigInsights Enterprise Edition Version 1.4.0.0 installation wizard. The browser address bar shows 'bdmn05.pbm.ihost.com:8300/Install/Components2.jsp'. The page title is 'IBM InfoSphere BigInsights Enterprise Edition Version 1.4.0.0'. On the left, a navigation pane lists various installation steps, with 'Components 2' selected. The main content area is divided into sections for NameNode, Secondary NameNode, JobTracker, and DataNode/TaskTracker. Each section has a text input field for the host name (all set to 'bdmn05.pbm.ihost.com') and an 'Assign...' button. The 'DataNode/TaskTracker' section is expanded, showing three radio button options: 'All nodes except the NameNode' (selected), 'All nodes except the NameNode, JobTracker, and Secondary NameNode', and 'Specify nodes'. Below these are four text input fields for ports: DataNode port (50010), DataNode IPC port (50020), DataNode HTTP port (50075), and TaskTracker HTTP port (50060). At the bottom, there is a section for specifying local file system paths, with the 'Data directory' field containing the path: '/hdfs,/hadoop/sdl/hdfs,/hadoop/sdm/hdfs,/hadoop/sdn/hdfs'. A note below this field states: 'Separate multiple paths with a comma. Each path specified as relative path (without a leading /), will be appended to /.'

Figure 6-10 Components settings

16. Click **Next** for the following component settings window. See Figure 6-11.

The screenshot shows the 'Components 3' settings window for IBM InfoSphere BigInsights Enterprise Edition Version 1.4.0.0. The browser address bar shows 'bdmn05.pbm.ihost.com:8300/Install/Components3.jsp'. The left sidebar contains a navigation menu with the following items: Welcome, License Agreement, Installation Type, File System, Secure Shell, Nodes, Disk List, Components 1, Components 2, Components 3 (selected), Security, Summary, Progress, and Results. The main configuration area is divided into several sections:

- Configure Hive:** A checked checkbox and a right-pointing arrow labeled 'Advanced settings'.
- Configure Pig:** A checked checkbox.
- ZooKeeper nodes:** A text input field containing 'bdmn05.pbm.ihost.com' and an 'Assign...' button. A right-pointing arrow labeled 'Advanced settings' is below it.
- Flume nodes:** A text input field containing 'bdmn05.pbm.ihost.com' and an 'Assign...' button.
- Flume master nodes:** A text input field containing 'bdmn05.pbm.ihost.com' and an 'Assign...' button.
- ZooKeeper mode:** Two radio button options: 'Use a shared ZooKeeper installation' (selected) and 'Use a separate ZooKeeper installation'.
- Oozie node:** A text input field containing 'bdmn05.pbm.ihost.com' and an 'Assign...' button. A right-pointing arrow labeled 'Advanced settings' is below it.
- Configure HBase:** A checked checkbox.
- HBase master servers:** A text input field containing 'bdmn05.pbm.ihost.com' and an 'Assign...' button.
- HBase region servers:** A text input field containing 'bdmn05.pbm.ihost.com' and an 'Assign...' button.
- ZooKeeper mode:** Two radio button options: 'Use a shared ZooKeeper installation' (selected) and 'Use a separate ZooKeeper installation'.
- A right-pointing arrow labeled 'Advanced settings' is at the bottom right of the HBase section.

Figure 6-11 Components settings

17. For security settings, we selected the **No user authentication** option (for our development servers, but you might consider another setting for your environment), then click **Next**. Figure 6-12 shows the security options.

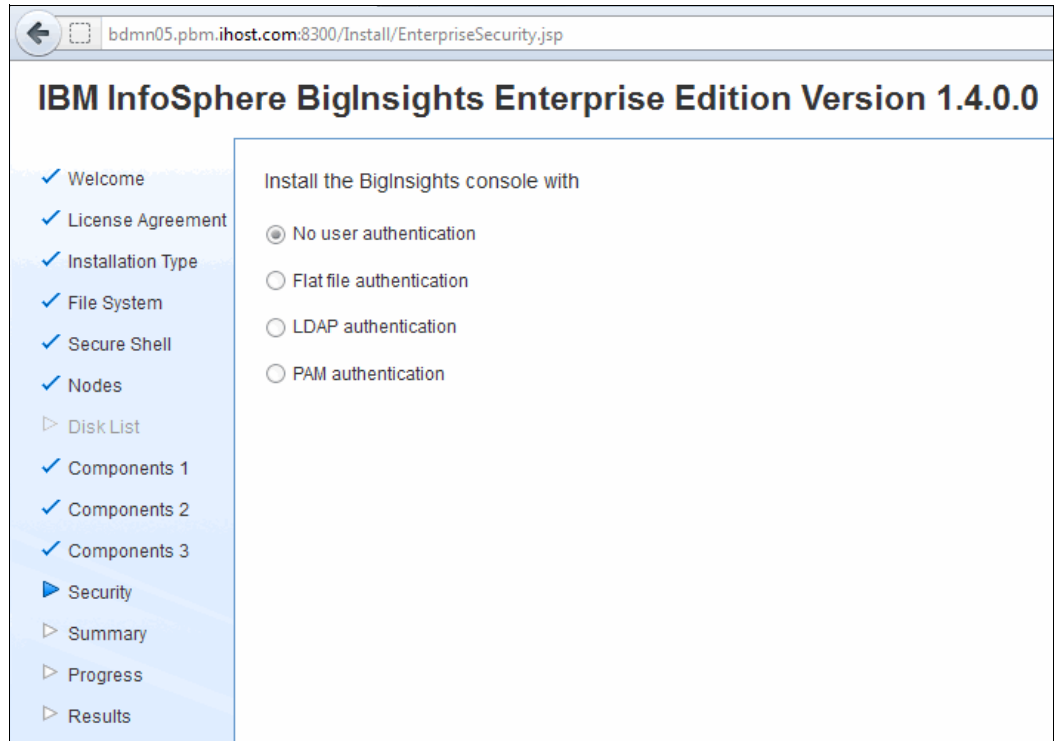


Figure 6-12 BigInsights security options

18. Review the Settings, Nodes, and Components tabs. If correct, click **Install**. See Figure 6-13.

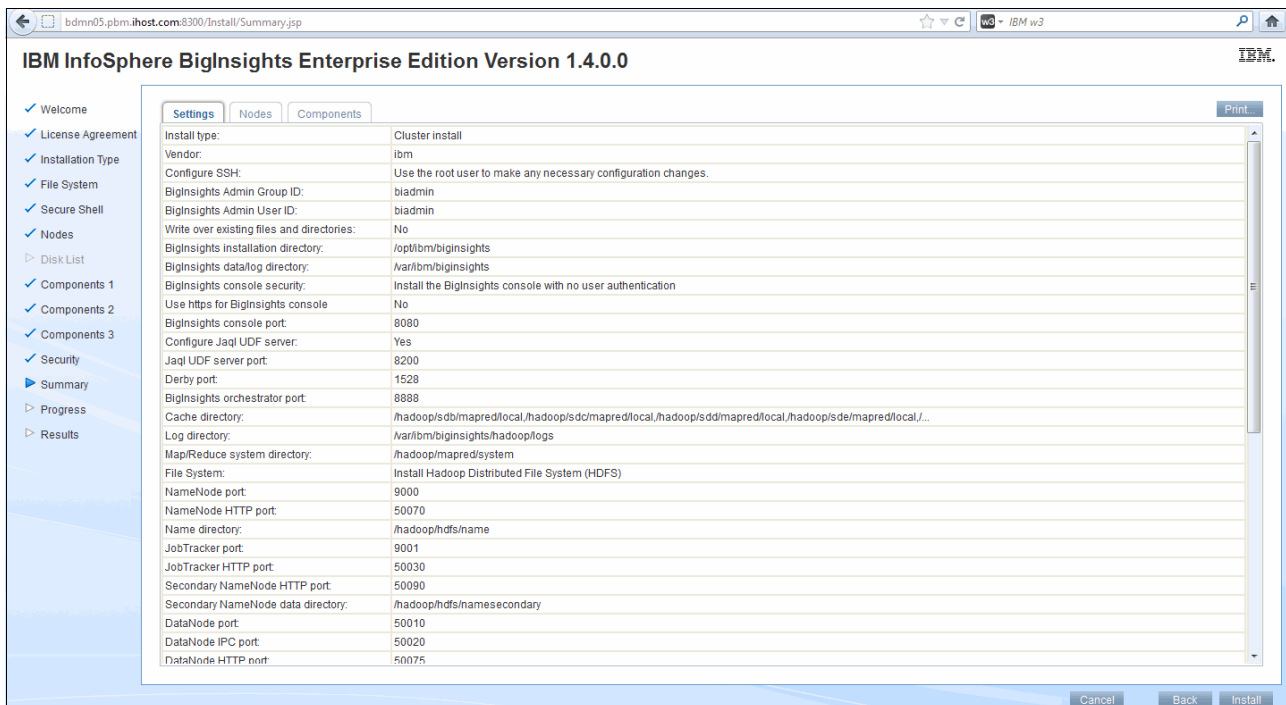


Figure 6-13 BigInsights installation summary

19. When the installation process completes successfully, check the installation log and BigInsights console. See Figure 6-14.

Review the results and click **Finish**, as shown in Figure 6-14.

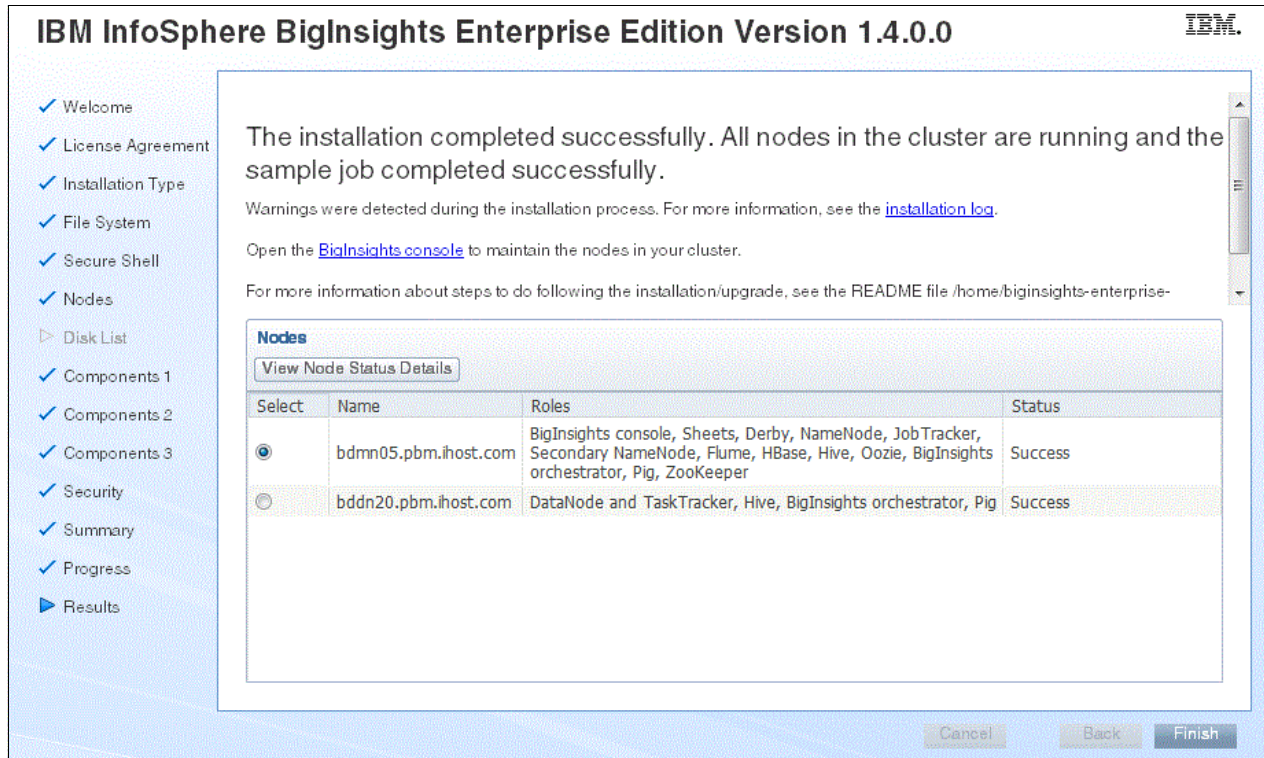


Figure 6-14 The installation completed successfully

If installer does not stop: Upon completion, if the installer does not stop automatically, you can run the following command to stop the installer web server manually:
[root@bdmn05 biginsights-enterprise-linux64_b20120604_2018]# ./start.sh shutdown

6.3 Silent installation of BigInsights

A *silent installation* is an installation that uses settings that are stored in a configuration file and runs in the background, requiring no input from the user while the installation completes. A silent installation does not use a GUI and is typically started from a command prompt on the management node as the *root user*.

6.3.1 Installing BigInsights using the silent installation option

A configuration file is required to install BigInsights by using the silent installation option. If an installation already occurred, a configuration file exists that can be used to install BigInsights with the previous installation settings. BigInsights also comes with five example configuration files that can be used. If a custom configuration file is required and does not exist, it must be created before a silent installation can take place. To simplify the creation of this file, there is an option within the GUI installation to assist you in creating the configuration file that is used as input into the silent installation program.

Previous installations: If an installation already ran, there is a configuration file that is called `fullinstall.xml` in the uncompressed installation directory. To use the same settings as the previous installation, you can rename (or copy) this file to `install.xml` and leave it in the uncompressed installation directory.

If a configuration file does not exist, a silent installation can be completed in three steps:

1. Create a configuration file.
2. Run the `silent-install.sh` command.
3. Verify that the installation completed successfully.

The configuration file can be called whatever you want. You then run the `silent-install.sh` command and supply in the location of your configuration file name. If you run the silent installation without providing the name of a configuration file, you need an `install.xml` file that is stored in the uncompressed installation directory because it is the default configuration file name. It can be created in a text editor or by selecting **Create a response file without performing an installation** during the web installation, as shown in Figure 6-15.

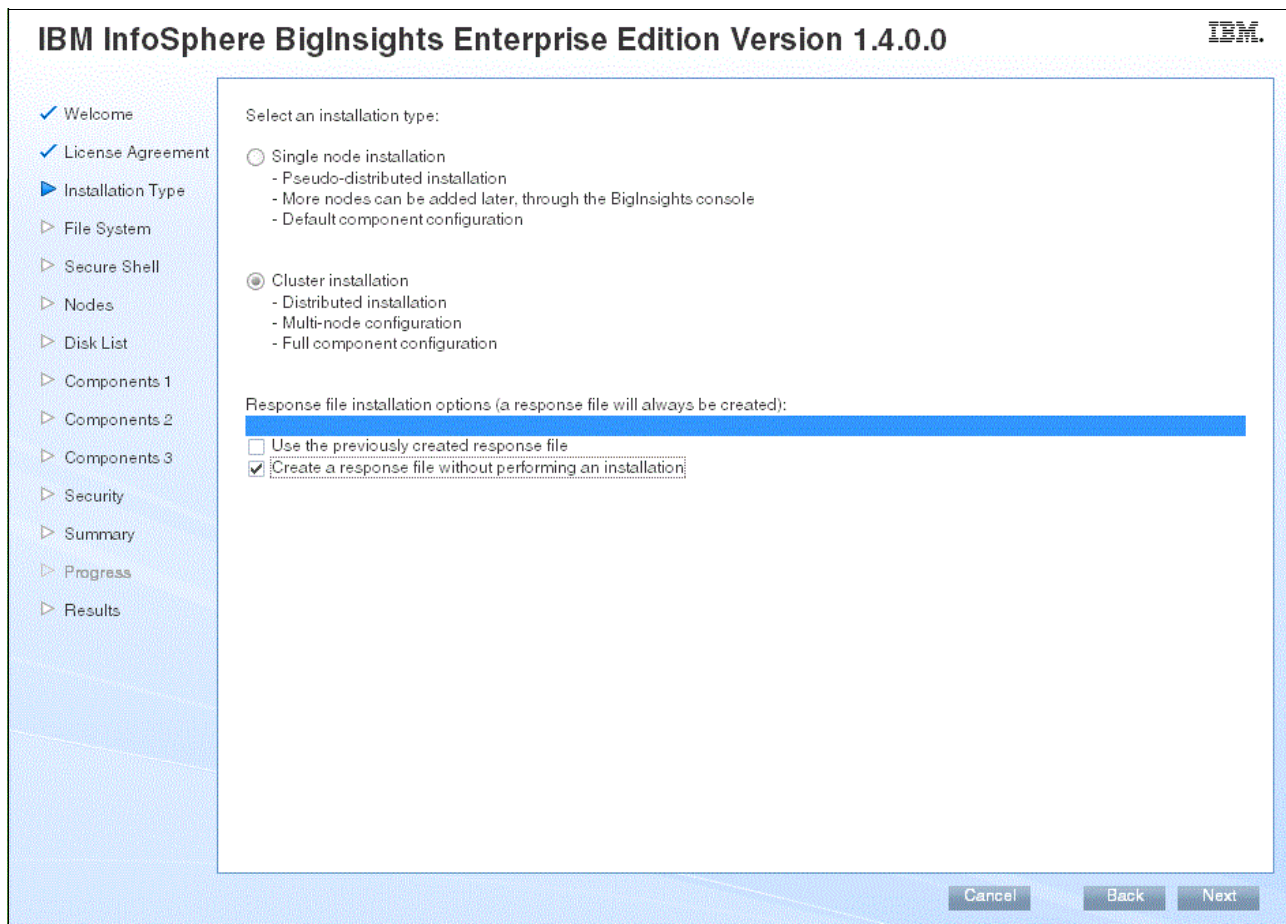


Figure 6-15 Create a response file only for a silent installation

When the configuration file is created, the silent installation can be run.

The files that are required for the silent installation can be found in the uncompressed installation directory under the `silent-install` folder. To begin the installation, log in as `root` and run the script `silent-install.sh`, shown in Example 6-4. If you want to use one of the example configuration files, specify this option as an argument after the `silent-install.sh` command.

Example 6-4 `silent-install.sh`

```
#!/usr/bin/env bash
#-----
# IBM Confidential
# OCO Source Materials
# 5725-C09 and 5725-D15
# (C) Copyright IBM Corp. 2010, 2011
# The source code for this program is not published or
# otherwise divested of its trade secrets, irrespective of
# what has been deposited with the U.S. Copyright Office.
#-----
# =====
# Usage: install.sh [install.xml]
# =====
silentinstallscript=`dirname "$0"`
silentinstallhome=`cd "$silentinstallscript"; pwd`
installerhome=`cd "$silentinstallhome/.."; pwd`

. "$installerhome/installer/hdm/bin/_info.sh"
# -----
xmlpath=$1

# get time stamp for silent install log file name
timestampfile=$silentinstallhome/timestampfile
# construct classpath
#CP=$installerhome/installer/hdm
hdm_home=$installerhome/installer/hdm
hdm_conf_dir=$hdm_home/conf

CP="$hdm_conf_dir"
for f in "$hdm_home"/lib/*.jar; do
    CP=${CP}:$f;
done
"$installerhome/_jvm/ibm-java-x86_64-60/bin/java" -cp "$CP"
com.ibm.xap.mgmt.installparser.silentInstallTimeStamp "$timestampfile"
timeStamp=`sed -nr -e "s/^timeStamp=(\S+)\$/\1/p" "$timestampfile"`
silentlog=$silentinstallhome/silent-install_${timeStamp}.log

echo "-----"
echo "Beginning the silent installation of BigInsights. For more information, see
the log $silentlog."
echo "-----"

$installerhome/installer/bin/install.sh $xmlpath 2>&1 | tee $silentlog

rm -rf $timestampfile

awk 'END {print}' $silentlog | grep "Installation Successful!" 1>&2
silentInstallExitCode=$?
```



```

if [ $silentInstallExitCode -eq 0 ]; then
    echo
    echo "Silent Installation Successful!"
    exit 0
else
    echo
    fatal 1 "Failed to silent install BigInsights."

```

Unless you specify an argument to the contrary, the script directs the installer towards a pre-configured XML installation file that is in the uncompressed BigInsights directory, called `install.xml`. When the script is run, expect to see the text that is shown in Example 6-5.

Example 6-5 Example output from a silent installation

```

-----
Beginning the silent installation of BigInsights. For more information, see the
log
/home/biginsights-enterprise-linux64_b20120604_2018_biginsights14/silent-install/s
ilent-install_2012-08-16T15.14.09.273_EDT.log.
-----
[INFO] Running as root,
/home/biginsights-enterprise-linux64_b20120604_2018_biginsights14/installer/bin/in
stall.sh
[INFO] Distribution Vendor : ibm
[INFO] Progress - Initializing install properties
[INFO] Progress - 0%

```

When the installation is complete, you see the text that is shown in Example 6-6.

Example 6-6 Confirmation example of a successful installation

```

[INFO] Progress - Health check sheets
[INFO] Progress - 100%
[INFO] DeployManager - Health check; SUCCEEDED components: [guardiumproxy,
zookeeper, hadoop, derby, jaql, hive, pig, hbase, flume, text-analytics, oozie,
orchestrator, jaqlserver, console, sheets]; FAILED components: []

Hadoop and BigInsights shell environment was setup for user 'biadmin'.
Please login as 'biadmin' to work from command line.
Or if you are already 'biadmin', you can 'source ~/.bashrc'

Installation Successful!

Silent Installation Successful!

```

Successful installation: If the installation completes successfully, the **FAILED components:** `[]` reference is displayed with an empty set of brackets.

6.4 How to install the Eclipse plug-in

BigInsights enables users to develop their own Text Analytics projects, develop HiveQL for interacting with Hive tables, write and test Jaql code, or publish MapReduce programs as applications within the BigInsights console for repeat usage. To unlock these functions, you first must install the Eclipse plug-in that is provided by BigInsights.

There is a link in the web browser welcome page, which is shown in Figure 6-16, that shows the initial link to select. An example of the instructions is shown in Figure 6-17 on page 89.

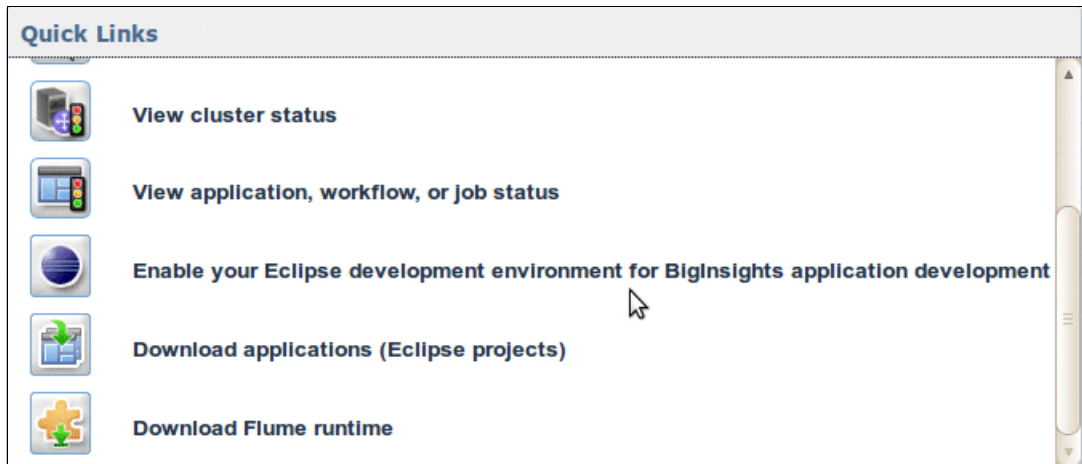


Figure 6-16 Eclipse plug-in download in Quick Links

Figure 6-17 shows the window with instructions on how to install Eclipse.

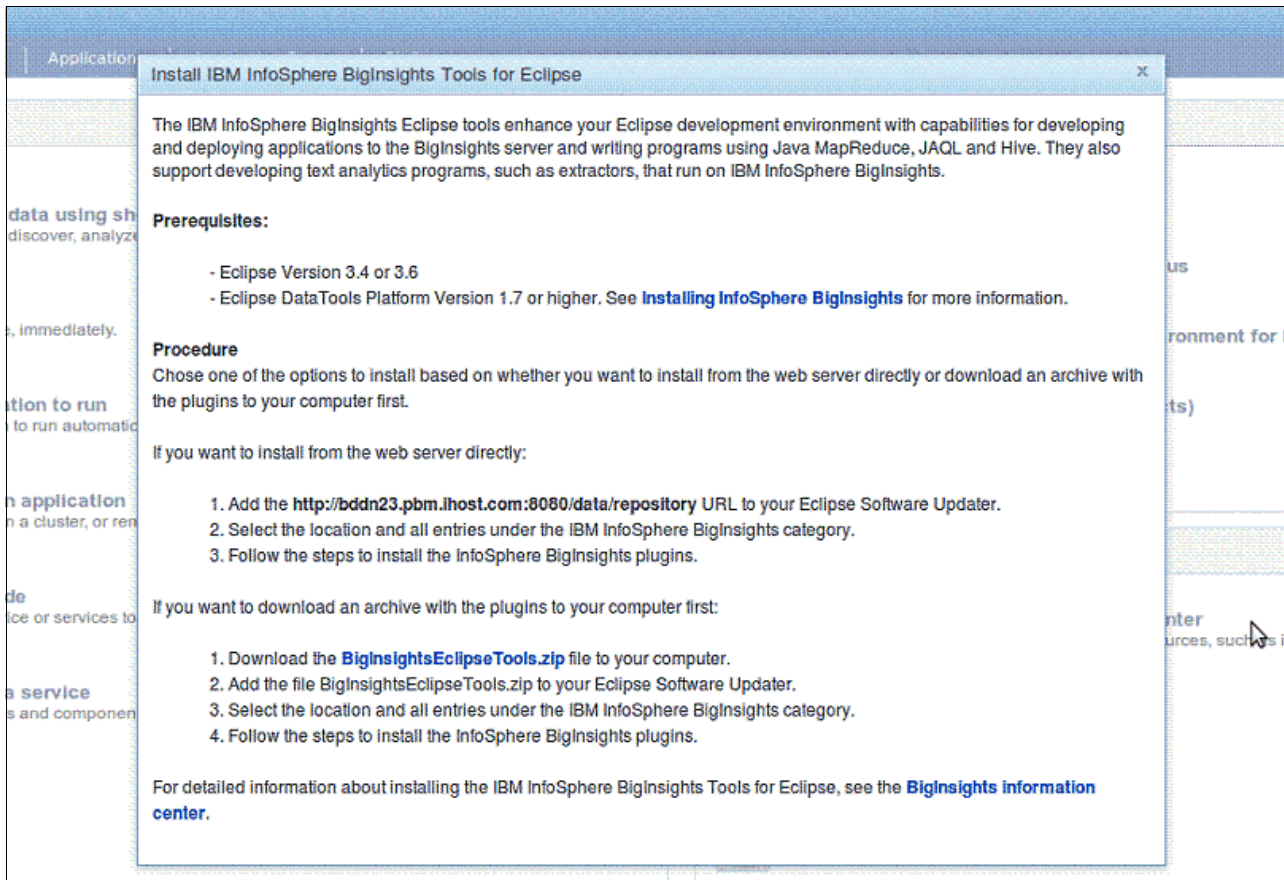


Figure 6-17 How to install Eclipse

When the plug-in is properly downloaded and configured, you see something similar to Figure 6-18 within your Eclipse environment.

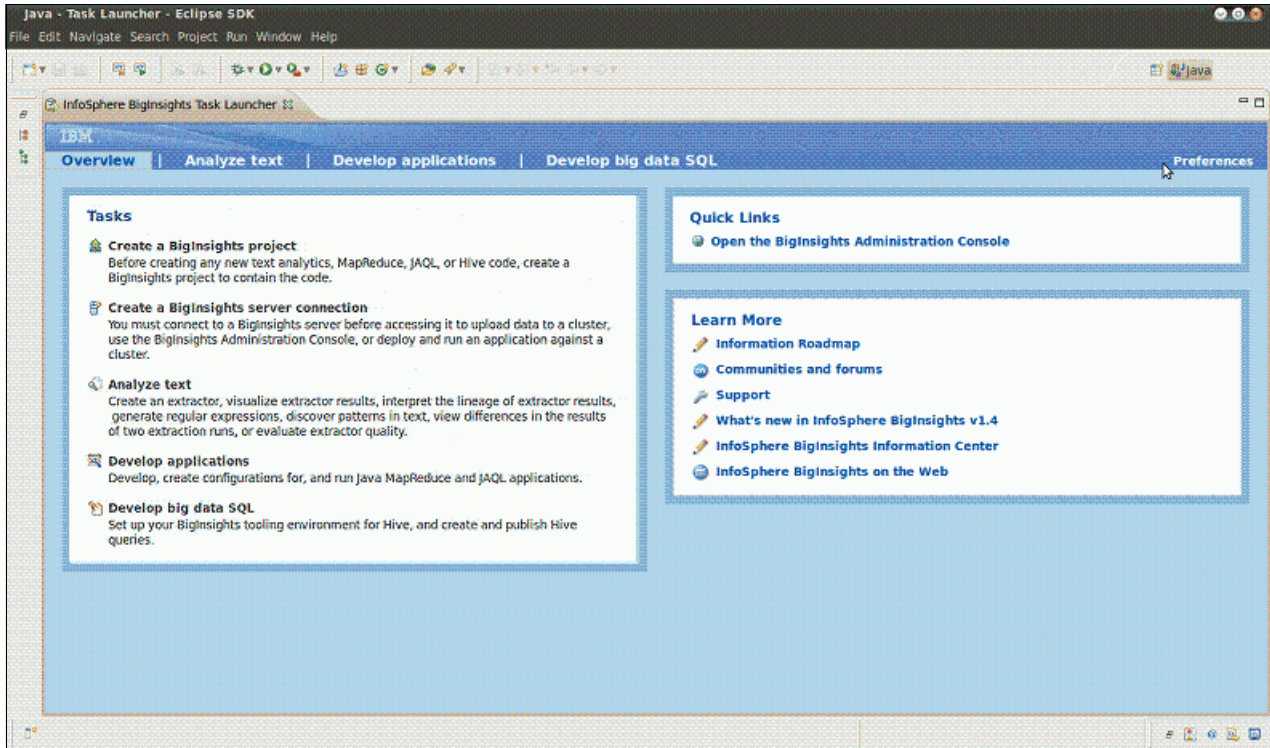


Figure 6-18 Eclipse BigInsights front page

An example of how to use the Eclipse interface for text analytics is covered in 8.4, “Text Analytics” on page 115.

6.5 Common installation pitfalls

If your BigInsights installation fails, it provides an error message, such as IUI0005E. As a troubleshooting first step, it is a good practice to review the *installation log* and look for a descriptive error message. We now cover many of the common errors.

Figure 6-19 shows the log file that you can expect to see if you try to install two components by using the same port numbers. To correct this error, read carefully through the settings and ensure that no components are configured to use the same ports.

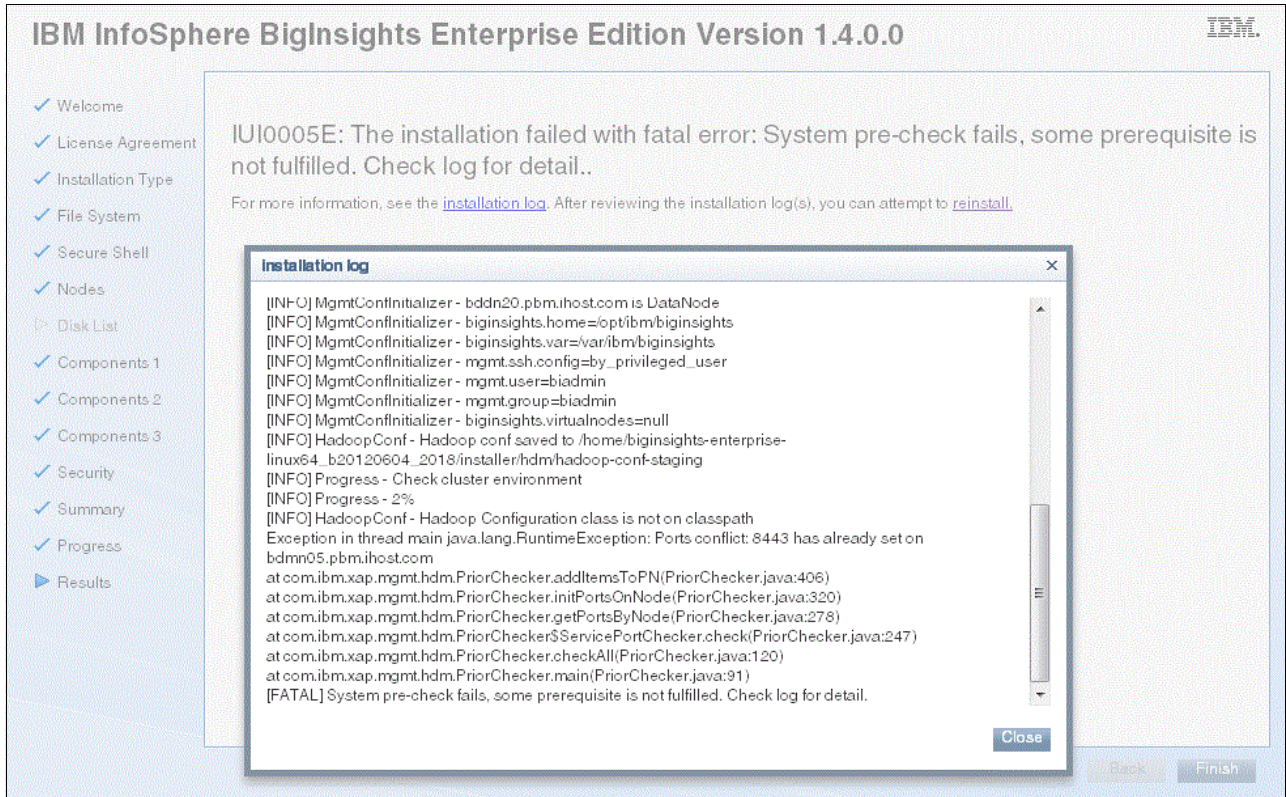


Figure 6-19 Duplicate ports error message

Figure 6-20 shows an error that can be caused by entering the wrong password for the administrator. To correct this error, go back and carefully reenter the correct password for the administrator where required.

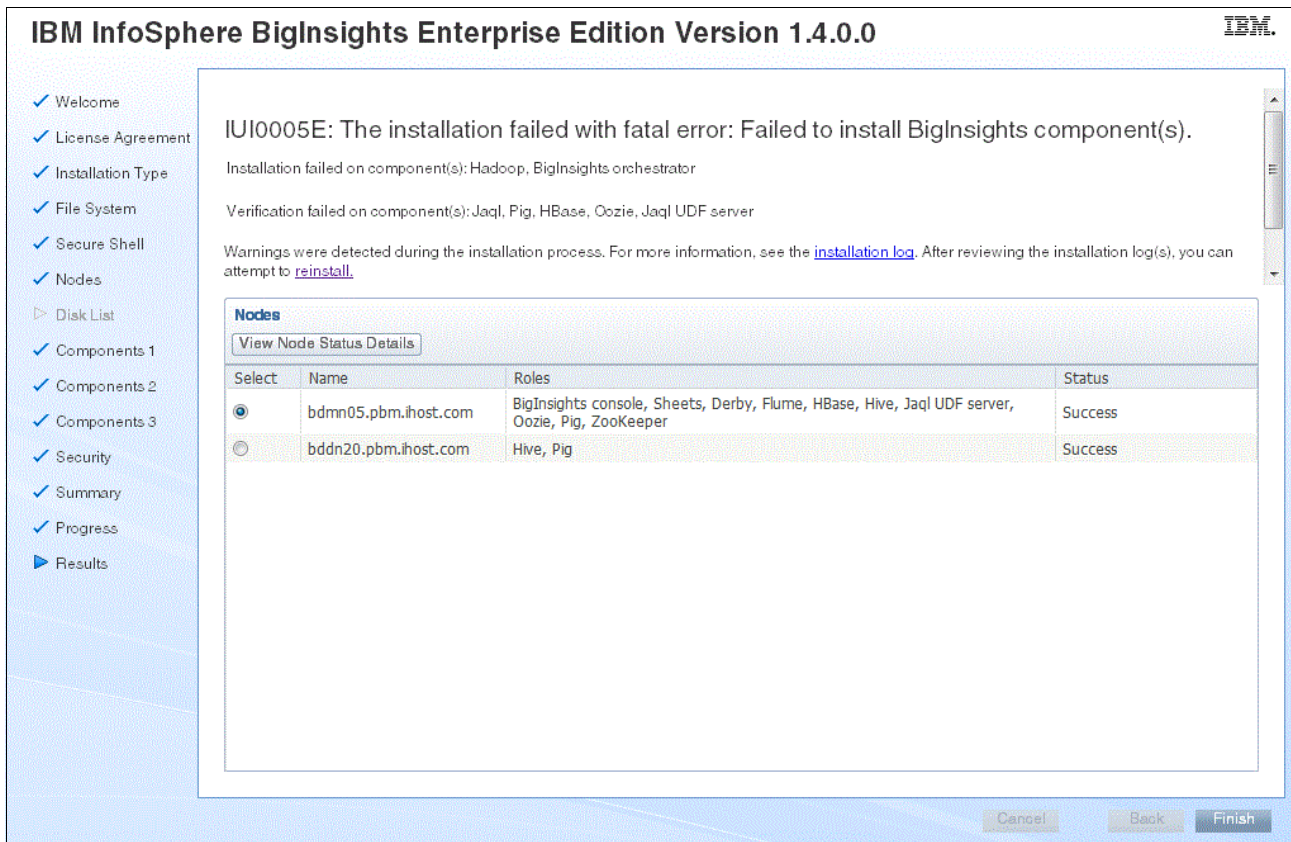


Figure 6-20 Error because of incorrect but matching admin passwords

Figure 6-21 shows the error that you can expect if you are reinstalling BigInsights into the same location and do not select the *Overwrite existing files and directories* option on the File System tab. To correct this error, reload the installation. Under the File System tab, ensure that you click **Overwrite existing files and directories**.

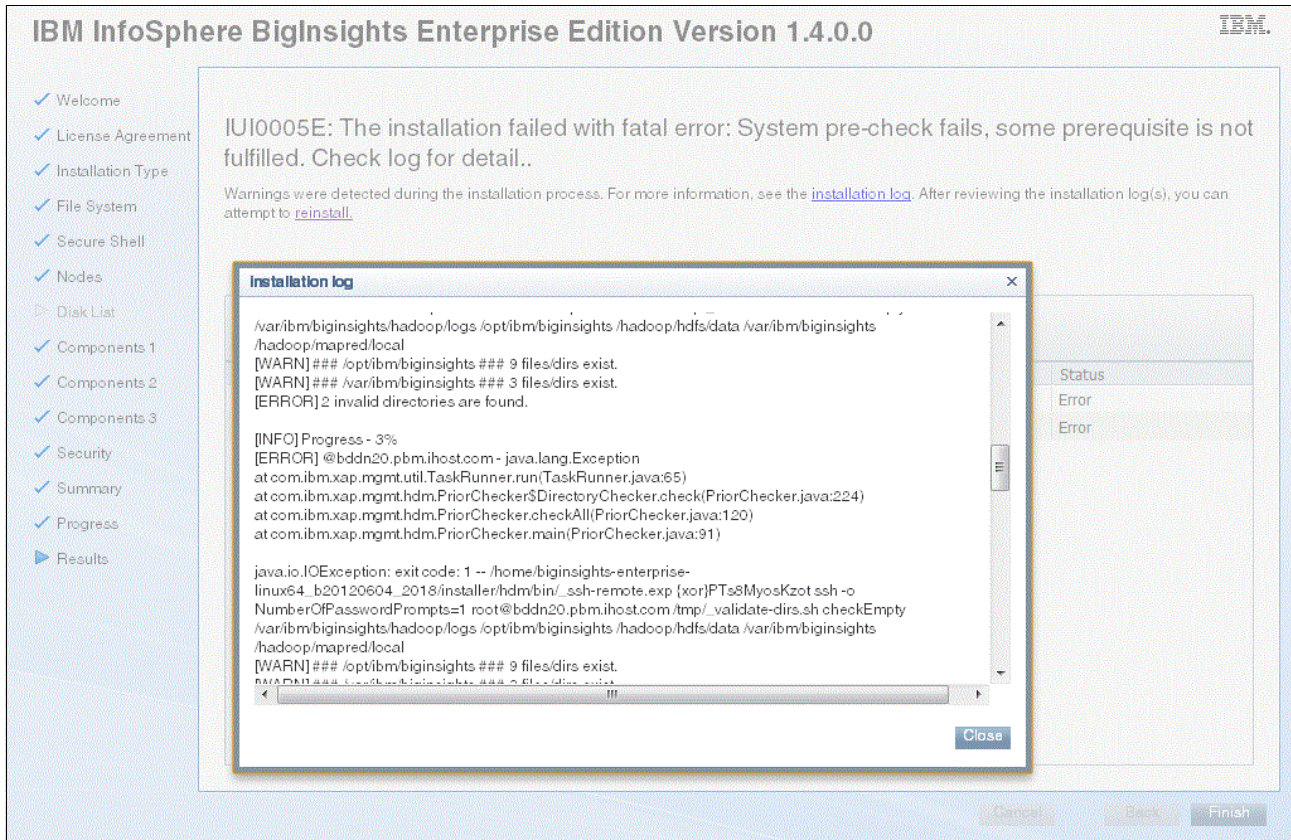


Figure 6-21 Cannot overwrite directory error

Figure 6-22 shows an error that can occur if you try to install BigInsights by using an existing shared directory space and enter the shared POSIX file system root directory incorrectly. To resolve this issue, either ensure that you enter the correct directory on the File System tab or choose to install HDFS instead.

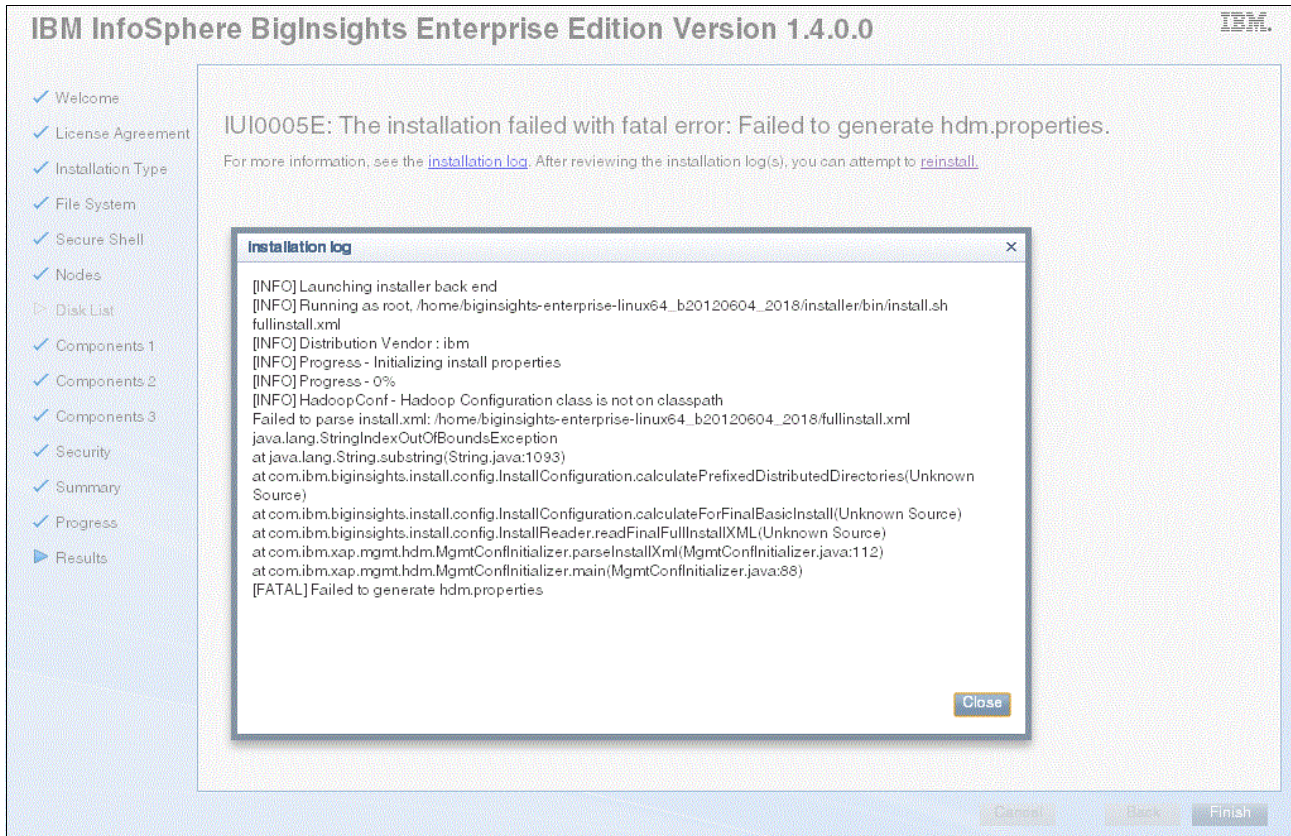


Figure 6-22 Error from using an incorrect root directory

Figure 6-23 shows what to expect if while trying to install, you accidentally select the *Create a response file without performing an installation* option during the Installation Type step. By selecting this option, you are instructing BigInsights not to install. To resolve this issue and do the installation, clear this option.

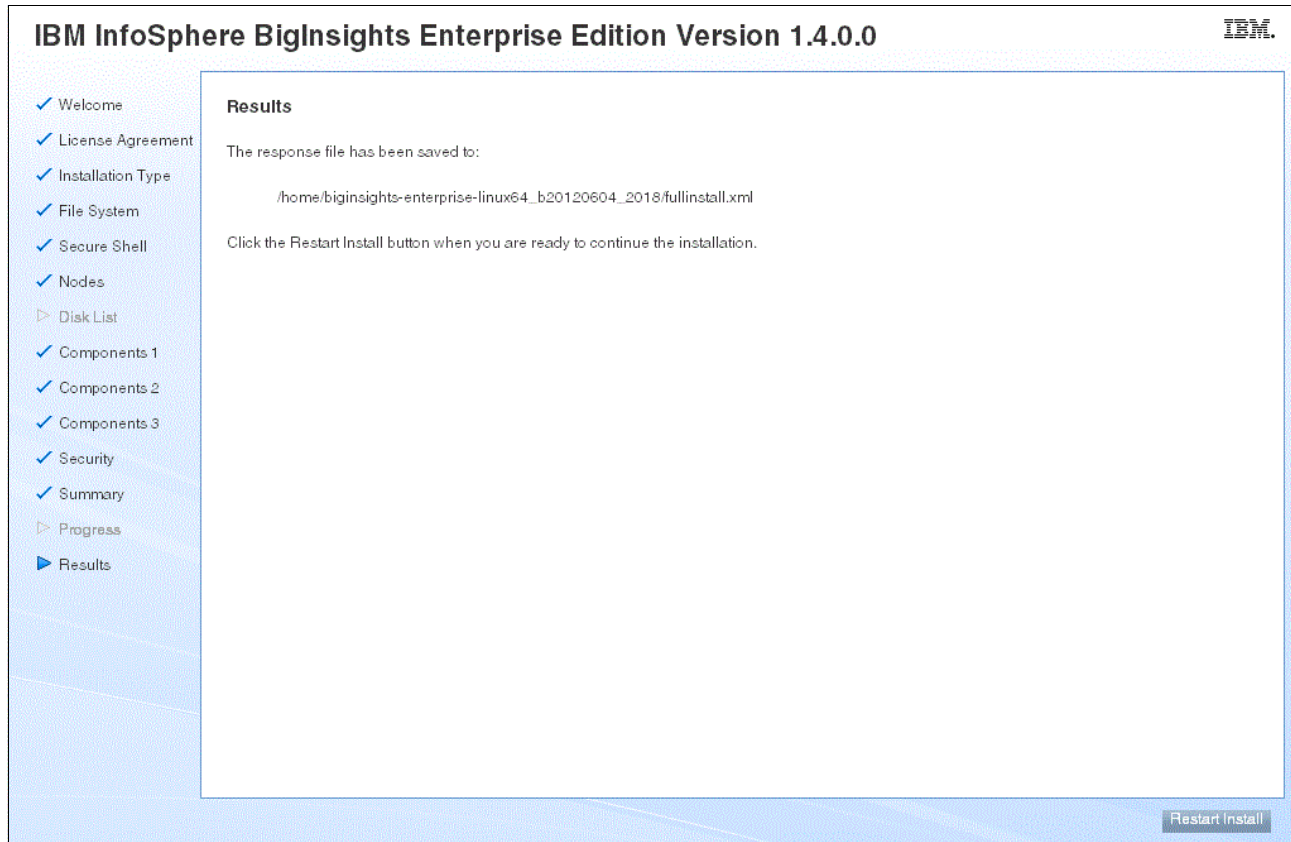


Figure 6-23 Response file only results

Figure 6-24 shows the error that you can expect if you try to install BigInsights while it is already installed and running. If you are sure that the product is not running, the required ports might be occupied by Java modules from a previous installation. These ports must be freed before you attempt a reinstallation.

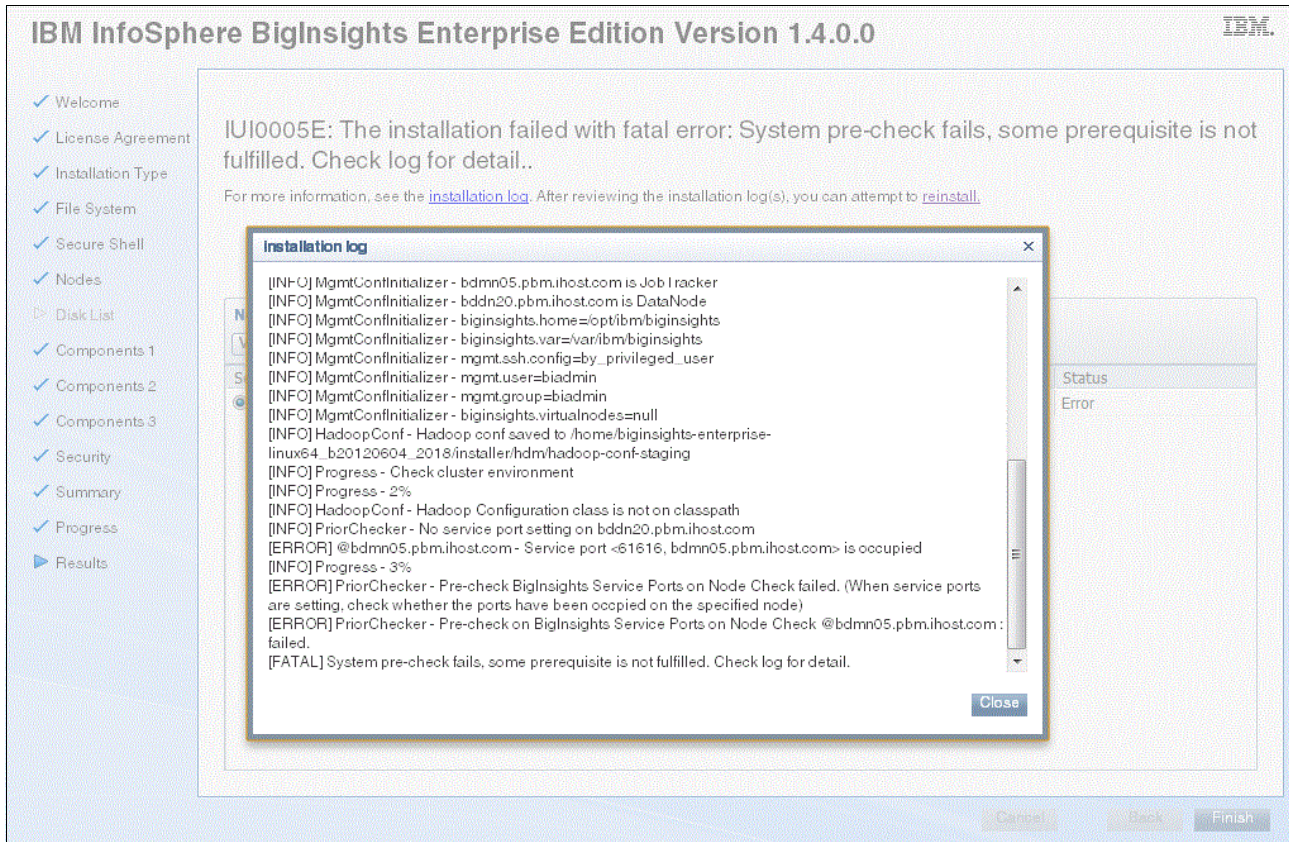


Figure 6-24 Installation error when attempting to install while BigInsights is running



Cluster validation

With an installed and configured BigInsights cluster, we took steps to validate that our cluster is functioning correctly. We now share that insight with you. There are several simple applications that you can run to help verify that your cluster is functioning properly. We also provide a quick, easy calculation method to gauge cluster performance.

7.1 Cluster validation

Up to this point, there has been a focus on the hardware and software components of the architecture. After you decide on the components and install the software, we have a few suggestions to help you verify that your cluster is working correctly.

7.1.1 Initial validation

There are initial validation methods available to ensure that your cluster is working properly. The BigInsights web console provides a quick view into the status of the cluster. When you access the web console and select the **Cluster Status** tab, the cluster status view that is shown in Figure 7-1, is displayed.

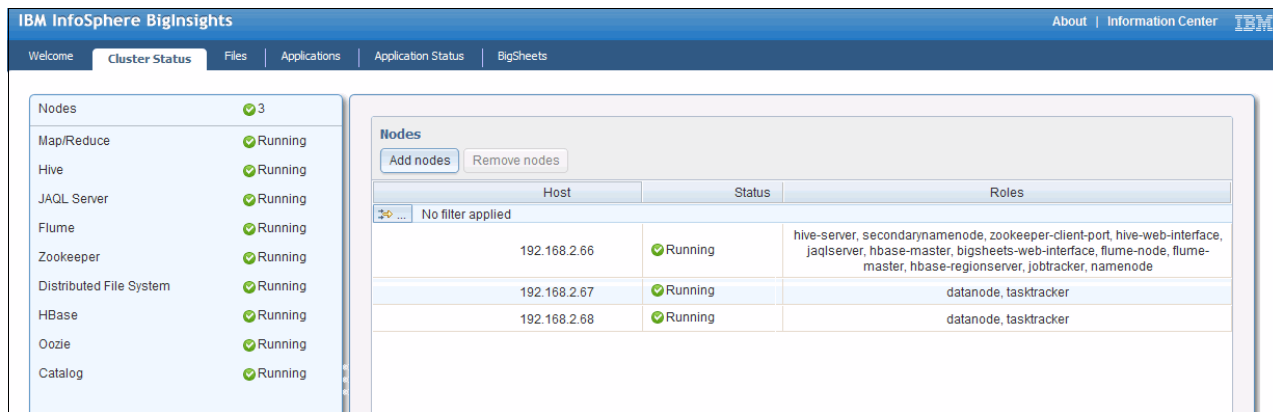


Figure 7-1 Cluster status view from the BigInsights web console

This is a three node cluster where the management node is at IP address 192.168.2.66 and the remaining two are data nodes. As we can see in Figure 7-1, all three nodes in the cluster are running.

Verifying components: The left side of the web console allows for verification of individual components within the BigInsights cluster (that is, Hive, Flume, and so on) are running as well. If you only wanted to focus on an individual component, you can click its name and the console shows you the status of the selected component.

7.1.2 Running the built-in health check utility

Another verification option is to run the health check utility that comes with BigInsights. By default, this tool runs through each installed component and checks to see if it is working properly on each node in your cluster. The tool runs from the command line of the management node and is in the target installation directory that you specified during your BigInsights installation.

Scripts: There are more scripts in this directory. These scripts are described in 7.4, “Other useful scripts” on page 107.

Example 7-1 Location of healthcheck.sh script

```
[biadmin@bddn26 bin]$ pwd
/opt/ibm/biginsights/bin
```

```
[biadmin@bddn26 bin]$ ls
addnode.sh credstore.sh listnode.sh rollbackUpgrade.sh status.sh syncconf.sh
biginsightslevel finalizeUpgrade.sh PMRStamping.sh start-all.sh stop-all.sh
uninstall.sh createosusers.sh healthcheck.sh removenode.sh start.sh
stop.sh upgradeHDFS.sh
```

The **healthcheck.sh** command is used to run the health check utility, which runs a set of tests to verify that each component is functioning appropriately on each node within the cluster. If you provide the **healthcheck.sh** command with one or more component names, it runs the health check tests on only those components. For example, the **healthcheck.sh flume** command only tests the flume component.

An example output can be seen in Example 7-2.

Example 7-2 Example output from healthcheck.sh

```
[INFO] Progress - Health check guardiumproxy
[INFO] Progress - 7%
[INFO] Progress - Health check zookeeper
[INFO] @192.168.2.66 - Zookeeper is running with pid = 21656
[INFO] @192.168.2.66 - zookeeper is healthy
[INFO] Deployer - zookeeper service is healthy
[INFO] Progress - 13%
[INFO] Progress - Health check hadoop
[INFO] Deployer - Running Hadoop word count example
[INFO] Deployer - hadoop service is healthy
[INFO] Progress - 20%
[INFO] Progress - Health check derby
[INFO] @192.168.2.66 - derby already running, pid 22459
[INFO] Progress - 21%
[INFO] @192.168.2.66 - derby is healthy
[INFO] Progress - 27%
[INFO] Progress - Health check jaql
[INFO] @192.168.2.66 - jaql is healthy
[INFO] @192.168.2.67 - jaql is healthy
[INFO] @192.168.2.68 - jaql is healthy
[INFO] Progress - 33%
[INFO] Progress - Health check hive
[INFO] @192.168.2.66 - hive-web-interface already running, pid 22581
[INFO] @192.168.2.66 - hive-server already running, pid 22893
[INFO] Progress - 35%
[INFO] @192.168.2.66 - hive is healthy
[INFO] @192.168.2.67 - hive is healthy
[INFO] @192.168.2.68 - hive is healthy
[INFO] Progress - 40%
[INFO] Progress - Health check pig
[INFO] @192.168.2.66 - pig is healthy
[INFO] @192.168.2.67 - pig is healthy
[INFO] @192.168.2.68 - pig is healthy
[INFO] Progress - 47%
[INFO] Progress - Health check hbase
[INFO] Deployer - hbase service is healthy
[INFO] Progress - 53%
[INFO] Progress - Health check flume
[INFO] @192.168.2.66 - flume-master started, pid 23678
[INFO] @192.168.2.66 - flume-node started, pid 24030
[INFO] @192.168.2.66 - flume-master is healthy
[INFO] Deployer - flume service is healthy
[INFO] Progress - 60%
```

```
[INFO] Progress - Health check text-analytics
[INFO] Progress - 67%
[INFO] Progress - Health check oozie
[INFO] @192.168.2.66 - oozie is healthy
[INFO] Progress - 73%
[INFO] Progress - Health check orchestrator
[INFO] @192.168.2.66 - orchestrator is healthy
[INFO] Progress - 80%
[INFO] Progress - Health check jaqlserver
[INFO] Deployer - jaqlserver service is healthy
[INFO] Progress - 87%
[INFO] Progress - Health check console
[INFO] Progress - 93%
[INFO] Progress - Health check sheets
[INFO] Progress - 100%
[INFO] DeployManager - Health check; SUCCEEDED components: [guardiumproxy, zookeeper,
hadoop, derby, jaql, hive, pig, hbase, flume, text-analytics, oozie, orchestrator,
jaqlserver, console, sheets]; FAILED components: []
```

7.1.3 Simple applications to run

Complementary to the simple validation techniques in 7.1.1, “Initial validation” on page 98, and 7.1.2, “Running the built-in health check utility” on page 98, are validations that require the execution of some simple applications to verify that your cluster is functioning properly. The BigInsights web console has an Applications tab, which is shown in Figure 7-2 on page 101, where users configure and run applications. Common applications that are used for cluster validation are WordCount, TeraGen, and TeraSort.

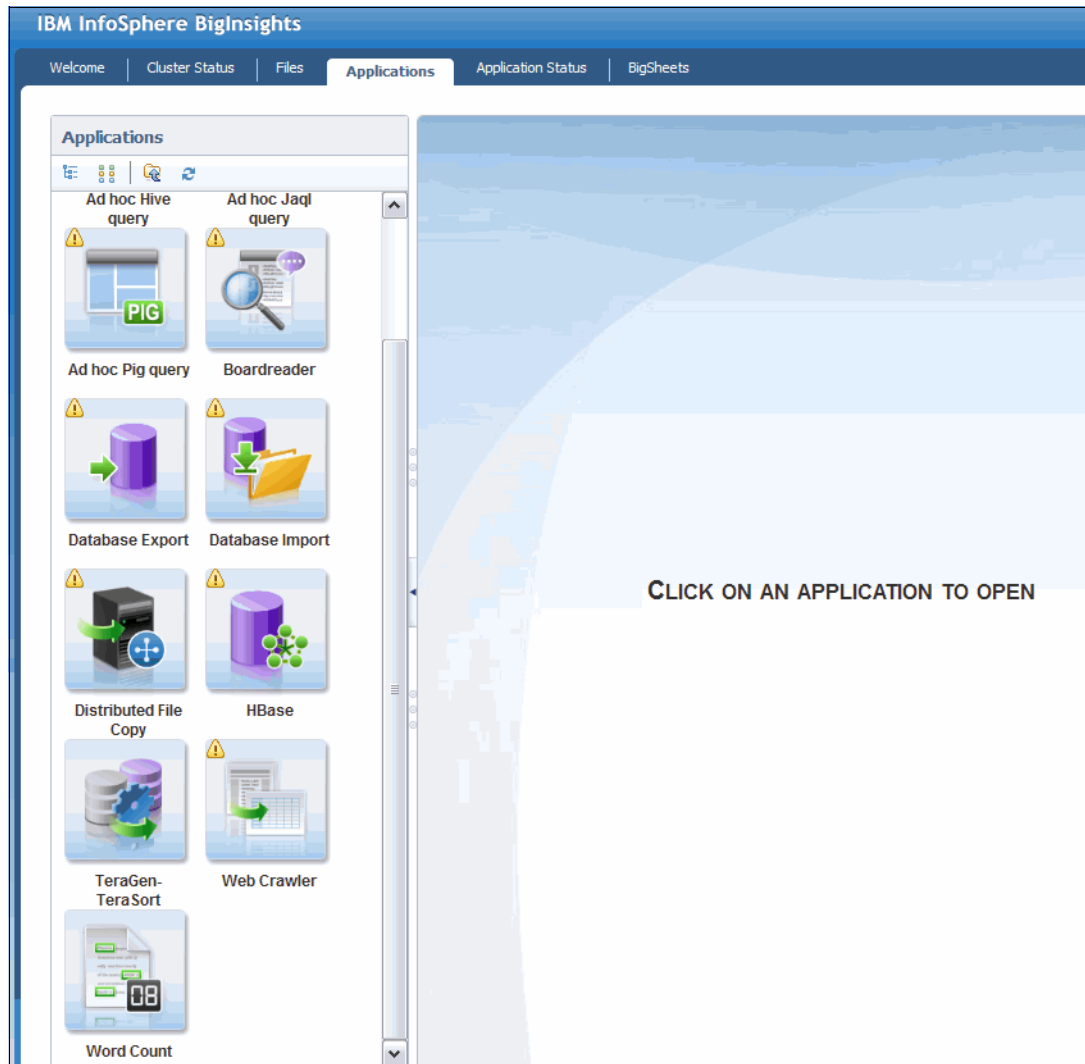


Figure 7-2 Applications tab in the BigInsights web console

Word Count outputs a total number of times each word is displayed in a specified data set. *TeraGen* generates a pre-specified row count of random text data. Lastly, *TeraSort* sorts the randomly generated data from *TeraGen*.

TeraGen and TeraSort: In BigInsights, TeraGen and TeraSort are combined as one application.

As an example, we walk through the process of running the TeraGen-TeraSort application. From the Applications tab in the web console, click the TeraGen-TeraSort application icon. The system prompts you to deploy the application to the cluster. Deploy the application to the cluster and then alter the details of the run as you want.

An example of the steps and settings we used on our cluster are shown in Figure 7-3.

Configuration settings to run the job

- ▶ Can name the runs each time that the application gets executed
- ▶ Select the number of rows to generate and output location for results
- ▶ Can select number of mappers and reducers (-1 denotes default setting)

Name: TeraGen-TeraSort

Description: The TeraGen-TeraSort application runs TeraGen to create random data followed by the TeraSort to sort the data.

Execution: Execution Name: 60M rows

Parameters:

- * Number of rows: 60000000
- * Output Path: /user/biadmin/redbook
- Number of Map Tasks: -1
- Number of Reduce Tasks: -1
- Child JVM size: NOT_SET

Application History

Status	Execution Name	Progress	Start Time	Elapsed Time (sec)	Output	Details
✓	60M rows	100%	Aug 30, 2012 10:35:49 AM	251		
✓	40M rows	100%	Aug 30, 2012 10:30:35 AM	182		
✓	20M rows	100%	Aug 30, 2012 10:25:47 AM	83		
✓	10M rows	100%	Aug 30, 2012 10:09:26 AM	82		
✓	1M rows	100%	Aug 30, 2012 10:07:34 AM	50		

Note: Several jobs have run already run

- ▶ 1 M to 60 M row jobs have run
- ▶ Notice elapsed time

Figure 7-3 Configuration and execution of the TeraGen-TeraSort application

Tip: These sample applications also serve as nice benchmarking tools to get an idea of how the cluster is functioning given certain workloads.

7.2 Performance considerations

In the preceding section, we noted that running sample applications can serve as a simple way to do some benchmarking of cluster performance. Although these applications are not likely to accurately simulate your business workload, they can provide indications of the anticipated throughput of the cluster.

We now take a closer look at the execution sequence of the BigInsights architecture as jobs are submitted to the cluster for processing. An example of the BigInsights stack can be seen in Figure 7-4.

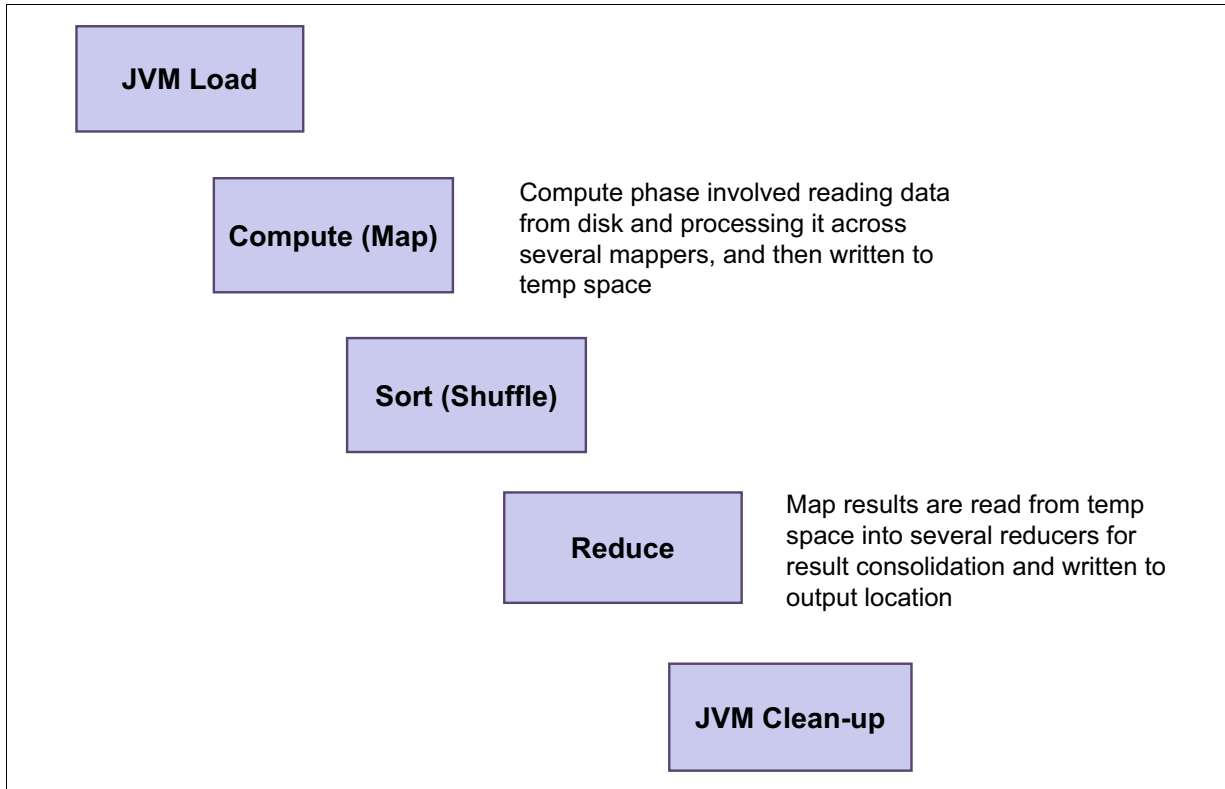


Figure 7-4 BigInsights execution stack

Figure 7-4 illustrates the five core steps that are performed by BigInsights as it runs a job on the cluster. Because BigInsights (based on Hadoop) is a batch process-oriented analytic engine, a significant portion of the processing time is spent reading data from disk. By understanding the specifications of the disks that are used in your cluster, you can theoretically calculate an estimated time that it takes to read the required data that is needed for processing. Continuing that thought process, if you also know how large your wanted result set is, you can also estimate the time that it takes to write the resulting output.

Example 7-3 contains a simple computation that is based on this principle.

Example 7-3 TB disk read example

Disk Read Speed: 75 - 150 MB/sec
 Disk Write Speed: 80 MB/sec
 Number of Disks: 20 (1.5GB/sec read across all disks)
 Amount of Data to Read: 1 TB
 Amount of Data to Write: 20 GB

Time to Read Disk: $1.5(GB/sec) \times (y \text{ secs}) = 1 \text{ TB (or 1000 GB)}$
 $y = 666.7 \text{ secs (11.11 mins)}$
 Time to Write Results: $1.6(GB/sec) \times (y \text{ secs}) = 20 \text{ GB}$
 $y = 12.5 \text{ secs}$

In Example 7-3, the lower limit of the disk read rate was used during the calculations. For example, it takes 11-12 minutes to read and write the data.

The other phases of the execution stack are configurable and are dependent on the amount of hardware resources available in the cluster. Because Hadoop is designed to enable parallel processing of jobs across the cluster, the time that is spent in the map and reduce phases of the stack can vary greatly.

Additionally, the number of reducers directly affects the amount of time that is spent in the shuffle phase. If you have only one reducer, the shuffle process is faster (in some cases) than having 10 reducers, which add a bit more time to the shuffle process but might improve the overall run time, as shown in Figure 7-5.

Notice change in number of reducers

Execution

Execution Name: 60M rows 10 reducers ▶ Run

Parameters

- * Number of rows: 60000000
- * Output Path: /user/biadmin/redbook Browse...
- Number of Map Tasks: -1
- Number of Reduce Tasks: 10
- Child JVM size: NOT_SET

[Schedule and Advanced Settings](#)

Application History

Status	Execution Name	Progress	Start Time	Elapsed Time (sec)	Output	Details
7 of 10 Jobs shown. Clear filter						
✔	60M rows 10 reducers	100%	Aug 30, 2012 1:20:39 PM	143	Output	Details
✔	40M rows 10 reducers	100%	Aug 30, 2012 11:48:12 AM	107	Output	Details
✔	60M rows	100%	Aug 30, 2012 10:35:49 AM	251	Output	Details
✔	40M rows	100%	Aug 30, 2012 10:30:35 AM	182	Output	Details
✔	20M rows	100%	Aug 30, 2012 10:25:47 AM	83	Output	Details
✔	10M rows	100%	Aug 30, 2012 10:09:26 AM	82	Output	Details
✔	1M rows	100%	Aug 30, 2012 10:07:34 AM	50	Output	Details

Notice runtime change from 10 reducers to 1

Figure 7-5 Run time improvements when more reducers were added

Notice the total *Elapsed Time* of the runs of the TeraGen-TeraSort application that were executed using 1- 60 million rows of generated data that were then sorted. The first few were executed using default parameters for the number of mappers, reducers, and Java virtual machine (JVM) parameters. In BigInsights, running applications with the default parameter specified still allows the platform to add as many mappers it feels necessary to increase the speed of processing. The situation for the reduce phase is different, because specifying the default value allows only the cluster to use one reduce process during the reducer processing of the job.

In the last two runs of the application, we increased the number of reducers to 10 instead of one. This yielded 58.8% performance improvement for the 40 million row executions and 57% performance improvement for the 60 million row executions. This improvement comes as no surprise because the performance is expected to increase when more resources are added to aid in the execution process.

7.3 TeraSort scalability and performance test example

To verify the performance and scalability of the BigInsights cluster, we used the built-in Hadoop TeraSort test that uses data sets of 1 and 2 terabytes using three different cluster sizes. See Example 7-4. We then measured the run times. The goal was to see if the run times scaled down linearly as nodes were added to the cluster, and if the run times scaled as we increased the data set size.

Example 7-4 Terasort scalability and performance test outline

DataNode 1 ManagementNode Cluster

Test 1 : Write 1 TB of data to HDFS using teragen, and then run terasort.

Test 2 : Write 2 TB of data to HDFS using teragen, and then run terasort.

6 DataNode 1 ManagementNode Cluster

Test 3 : Write 1 TB of data to HDFS using teragen, and then run terasort.

Test 4 : Write 2 TB of data to HDFS using teragen, and then run terasort.

9 DataNode 1 ManagementNode Cluster

Test 5 : Write 1 TB of data to HDFS using teragen, and then run terasort.

Test 6 : Write 2 TB of data to HDFS using teragen, and then run terasort.

We used the Hadoop command-line interface to start the TeraGen and TeraSort runs. After each TeraSort run was completed, the original source and destination files were deleted from Hadoop Distributed File System (HDFS). As the number of data nodes was increased, the requested number of reducers was scaled accordingly to use the parallelism available during the reduce phase of the sort. Example 7-5 has both the **TeraGen** and **TeraSort** commands that were run to produce these results.

Example 7-5 Hadoop TeraGen and TeraSort command sequence

Commands run in bash shell

```
ONETB=10000000000 # 1TB 10 Billion 100 Byte rows
TWOBTB=20000000000 # 2TB 20 Billion 100 Byte rows
HADOOP_EXAMPLE_PATH=/opt/ibm/biginsights/hdm/IHC/hadoop-examples-1.0.0.jar
```

```
# 1 TB Run, 3 DataNode
```

```
hadoop jar $HADOOP_EXAMPLE_PATH teragen -Dmapred.map.tasks=216 $ONETB
/user/biadmin/tgen.128MBblk.1TB
```

```
hadoop jar $HADOOP_EXAMPLE_PATH terasort -Dmapred.reduce.tasks=36
/user/biadmin/tgen.128MBblk.1TB /user/biadmin/tgen.128MBblk.1TB.tsort
```

```
hadoop fs -rmr /user/biadmin/tgen.128MBblk.1TB
```

```
hadoop fs -rmr /user/biadmin/tgen.128MBblk.1TB.tsort
```

```
# 2 TB Run, 3 DataNode
```

```
hadoop jar $HADOOP_EXAMPLE_PATH teragen -Dmapred.map.tasks=216 $TWOBTB
/user/biadmin/tgen.128MBblk.2TB
```

```
hadoop jar $HADOOP_EXAMPLE_PATH terasort -Dmapred.reduce.tasks=36
/user/biadmin/tgen.128MBblk.2TB /user/biadmin/tgen.128MBblk.2TB.tsort
```

```
hadoop fs -rmr /user/biadmin/tgen.128MBblk.2TB
```

```
hadoop fs -rmr /user/biadmin/tgen.128MBblk.2TB.tsort
```

```
## Add 3 nodes to cluster
```

```
# 1 TB Run, 6 DataNode
```

```

hadoop jar $HADOOP_EXAMPLE_PATH teragen -Dmapred.map.tasks=216 $ONETB
/user/biadmin/tgen.128MBblk.1TB
hadoop jar $HADOOP_EXAMPLE_PATH terasort -Dmapred.reduce.tasks=72
/user/biadmin/tgen.128MBblk.1TB /user/biadmin/tgen.128MBblk.1TB.tsort
hadoop fs -rmr /user/biadmin/tgen.128MBblk.1TB
hadoop fs -rmr /user/biadmin/tgen.128MBblk.1TB.tsort
# 2 TB Run, 6 DataNode
hadoop jar $HADOOP_EXAMPLE_PATH teragen -Dmapred.map.tasks=216 $TWO2TB
/user/biadmin/tgen.128MBblk.2TB
hadoop jar $HADOOP_EXAMPLE_PATH terasort -Dmapred.reduce.tasks=72
/user/biadmin/tgen.128MBblk.2TB /user/biadmin/tgen.128MBblk.2TB.tsort
hadoop fs -rmr /user/biadmin/tgen.128MBblk.2TB
hadoop fs -rmr /user/biadmin/tgen.128MBblk.2TB.tsort

## Add 3 nodes to cluster

# 1 TB Run, 9 DataNode
hadoop jar $HADOOP_EXAMPLE_PATH teragen -Dmapred.map.tasks=216 $ONETB
/user/biadmin/tgen.128MBblk.1TB
hadoop jar $HADOOP_EXAMPLE_PATH terasort -Dmapred.reduce.tasks=108
/user/biadmin/tgen.128MBblk.1TB /user/biadmin/tgen.128MBblk.1TB.tsort
hadoop fs -rmr /user/biadmin/tgen.128MBblk.1TB
hadoop fs -rmr /user/biadmin/tgen.128MBblk.1TB.tsort
# 2 TB Run, 9 DataNode
hadoop jar $HADOOP_EXAMPLE_PATH teragen -Dmapred.map.tasks=216 $TWO2TB
/user/biadmin/tgen.128MBblk.2TB
hadoop jar $HADOOP_EXAMPLE_PATH terasort -Dmapred.reduce.tasks=108
/user/biadmin/tgen.128MBblk.2TB /user/biadmin/tgen.128MBblk.2TB.tsort
hadoop fs -rmr /user/biadmin/tgen.128MBblk.2TB
hadoop fs -rmr /user/biadmin/tgen.128MBblk.2TB.tsort

```

We selected the three data node 1 TB TeraSort run as the baseline, and all the other run times are normalized to that time. As shown in Figure 7-6 on page 107, the BigInsights cluster scales for both the data set sizes and is based on the number of data nodes running in parallel. Although not exactly perfect scaling, they are well within the limits of reasonable expectation. This type of verification is useful when you try to understand the scalability of different types of workloads. It also helps to understand if there are hardware bottlenecks within the system. For example, if the run times did not scale linearly when you double the number of data nodes in the system, this might suggest an issue with code or network throughput.

Figure 7-6 shows Hadoop scalability and performance test results.

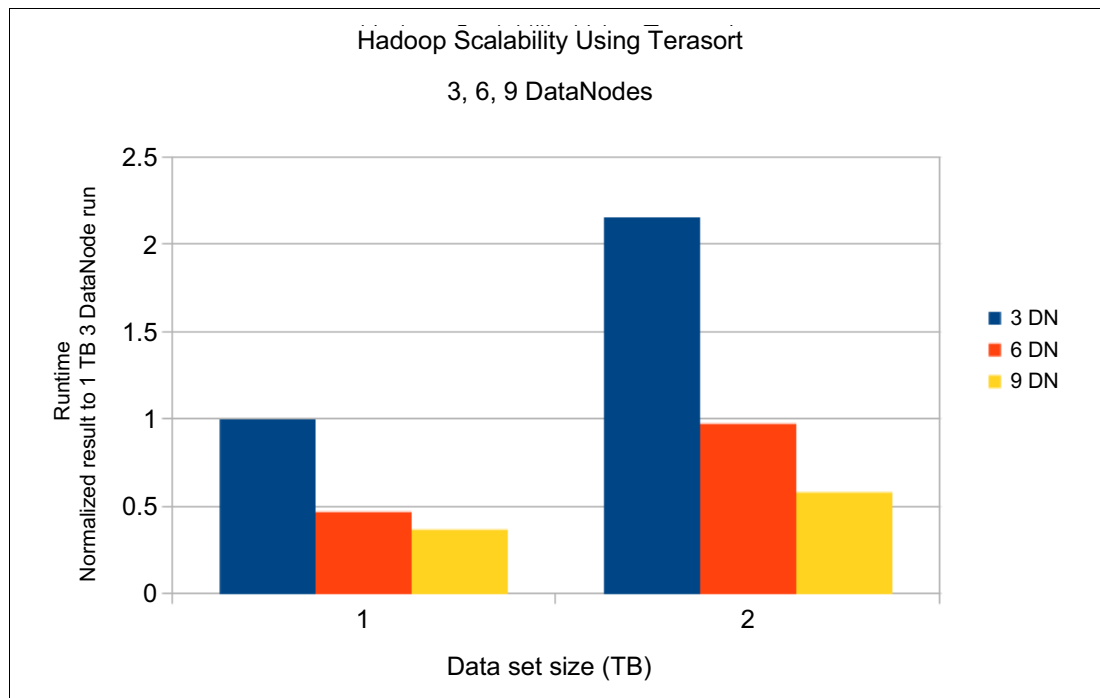


Figure 7-6 Hadoop scalability and performance test results

7.4 Other useful scripts

The `/opt/ibm/biginsights/bin` directory currently contains 18 scripts, one of which is seen. In 7.1, “Cluster validation” on page 98, we covered the `healthcheck.sh` script. The other scripts can be seen in Example 7-6.

Example 7-6 /opt/ibm/biginsights/bin scripts

```

-rwxr-xr-x 1 biadmin biadmin 613 Aug 31 13:41 addnode.sh
-r-xr-xr-x 1 biadmin biadmin 1319 Aug 31 13:41 biginsightslevel
-rwxr-xr-x 1 biadmin biadmin 619 Aug 31 13:41 createosusers.sh
-rwxr-xr-x 1 biadmin biadmin 3059 Aug 31 13:41 credstore.sh
-rwxr-xr-x 1 biadmin biadmin 620 Aug 31 13:41 finalizeUpgrade.sh
-rwxr-xr-x 1 biadmin biadmin 617 Aug 31 13:41 healthcheck.sh
-rwxr-xr-x 1 biadmin biadmin 614 Aug 31 13:41 listnode.sh
-rwxr-xr-x 1 biadmin biadmin 2139 Aug 31 13:41 PMRStamping.sh
-rwxr-xr-x 1 biadmin biadmin 1355 Aug 31 13:41 removenode.sh
-rwxr-xr-x 1 biadmin biadmin 620 Aug 31 13:41 rollbackUpgrade.sh
-rwxr-xr-x 1 biadmin biadmin 610 Aug 31 13:41 start-all.sh
-rwxr-xr-x 1 biadmin biadmin 626 Sep 5 14:39 start.sh
-rwxr-xr-x 1 biadmin biadmin 612 Aug 31 13:41 status.sh
-rwxr-xr-x 1 biadmin biadmin 609 Aug 31 13:41 stop-all.sh
-rwxr-xr-x 1 biadmin biadmin 610 Aug 31 13:41 stop.sh
-rwxr-xr-x 1 biadmin biadmin 1256 Aug 31 13:41 synconf.sh
-rwxr-xr-x 1 biadmin biadmin 3466 Sep 5 14:45 uninstall.sh
-rwxr-xr-x 1 biadmin biadmin 616 Aug 31 13:41 upgradeHDFS.sh

```

We now look at a few scripts and explain when they might be useful.

7.4.1 addnode.sh

This script is used to add more nodes to the cluster and takes the following form:

```
addnode.sh <component> <node>[,password[,/rack[,disklist]]]...
```

The command can be used to add multiple nodes at the same time. Passwords and rack ID can be specified in the command too. If BigInsights is used to set up password-less Secure Shell (SSH), the password also must be entered here. This script is used by the console.

7.4.2 credstore.sh

This script can be used for three functions:

1. It can load credentials from the users credential store or prop file. It decodes automatically and writes results to a file or stdout.
2. It can store a set of key values pairs from an input file to a users credential store.
3. It can update an existing credential store file with a set of key value pairs.

The syntax of the command is different in each case. Further information can be found at the website link that is provided in sub-section 7.4.5, “status.sh” on page 108.

7.4.3 syncconf.sh

The sync configuration script is used in the following form:

```
syncconf.sh <component>...
```

It can be used to synchronize the configuration of one or more components of BigInsights or all of the components if “all” is specified.

7.4.4 start.sh, stop.sh, start-all.sh, and stop-all.sh

These scripts are used to start or stop BigInsights. The use of the **start.sh** or **stop.sh** scripts followed by one or more components, starts or stops those components. Running the **start-all.sh** and **stop-all.sh** commands, starts or stops all components.

7.4.5 status.sh

Rather than doing a full health check, the **status.sh** command can be used to check the running status of one or more of the installed BigInsights components. The command takes the following structure:

```
status.sh <component>...
```

If “all” is specified as the component, the script checks the status of all of the components.

Shell scripts: For more information about any of the scripts, see this website:
<http://pic.dhe.ibm.com/infocenter/bigins/v1r4/topic/com.ibm.swg.im.infosphere.biginsights.admin.doc/doc/r0057945.html?resultof=%22%73%74%61%72%74%2e%73%68%22%20>



BigInsights capabilities

BigInsights includes several IBM developed technologies to help users to extract added insight from massive volumes of structured and unstructured data.

In addition to providing open source tools, which include Pig, Jaql, and Lucene, BigInsights offers value-added advanced analytic capabilities. The capabilities of BigInsights include text analytics, spreadsheet style data analysis and visualization, advanced indexing, and improved workload management. In addition, BigInsights features enterprise-class integration capabilities to speed up and simplify your deployment of applications for analysis of structured and unstructured data.

This chapter covers these capabilities, focusing on components, functionalities, and benefits that are brought by these additional components. We also demonstrate how to load and manage data within the BigInsights environment.

8.1 Data ingestion

One of the most advantageous attributes of BigInsights is the facility to ingest and analyze massive amounts of data. Currently offered as a download at no charge, InfoSphere BigInsights Basic Edition can manage up to 10 TB of data, though the Enterprise Edition presents no restriction on the quantity of data that can be managed.

It is important to know what type of data that you want to ingest and the type of analysis that you want to do on that before you decide which tool to use to ingest data. Depending on the format of the data to be ingested, there might be a preferable way of ingesting that type of data. In the following sections, we present some methods of ingesting different types of data.

8.1.1 Loading data from files using the web console

Data at rest is any type of data that is inactive, stored physically in a database, archive, data warehouse, and so on. It is possible to ingest this data from a file on your local system, by using the web-browser interface provided by BigInsights as shown in Figure 8-1. Keep in mind, this technique for ingesting data is only useful for files that are small enough to be transferred to your cluster within an http session. Generally, if you think your session might time out before the file gets stored properly within Hadoop Distributed File System (HDFS), use another technique to ingest your file into the cluster.

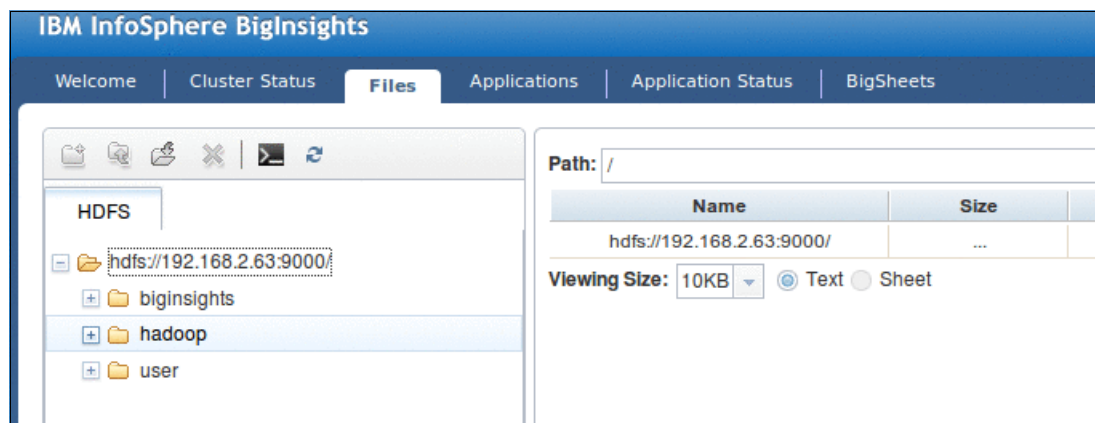


Figure 8-1 Loading files in the web browser

To generate the view that is shown in Figure 8-1, click the **Files** tab. On the left side, you are presented with your Hadoop File System, including a hierarchy of folders and files. You can choose the appropriated folder and then click the toolbar icon for upload to ingest the wanted file. You are prompted to select and ingest a file from your local system. BigInsights does allow you to upload multiple files at one time. Keep in mind the session timeout restrictions when you select the volume of data you want to ingest at one time.

When you are ready to start the upload, click the OK icon, as shown in Figure 8-2 on page 111.

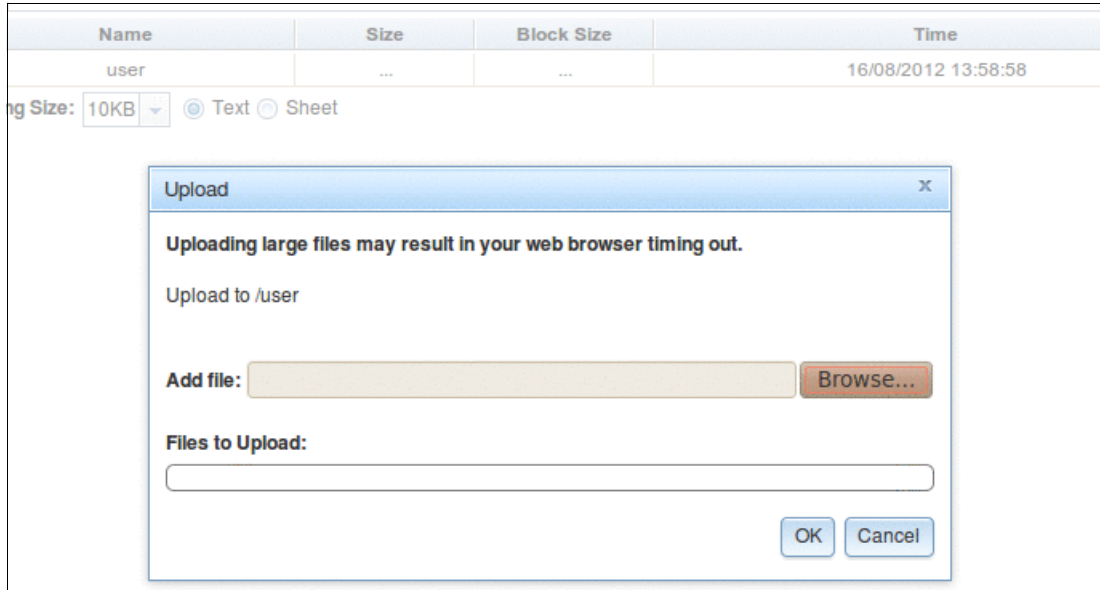


Figure 8-2 BigInsights File upload

Figure 8-3 shows the resulting file that is uploaded.

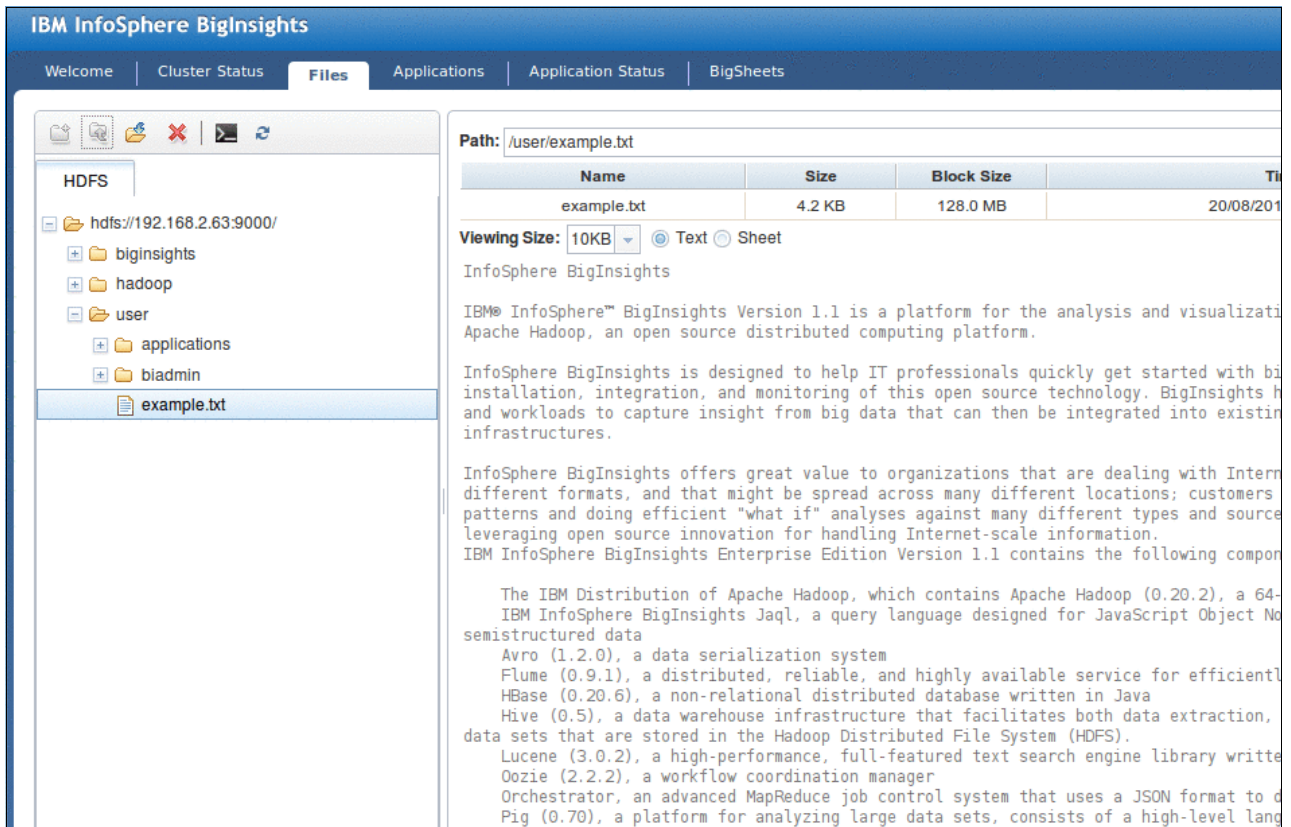


Figure 8-3 Resulting file that is uploaded

8.1.2 Loading files from the command line

Another method of importing data is to use the command line in a terminal session. If the data that you want to ingest is on your local machine, you first import the data through a file transfer command, the `scp` command for example, to the management node in the cluster. Then, run the line commands that are required to import your data into HDFS.

To copy the wanted files from our local system to the cluster management node, we used the following command from the wanted target directory on the management node:

```
$scp user@mylocalmachine:/home/mydir/data
```

If you are running Windows and want to “push” the data to the management node, this transfer can also be done by using FTP, SFTP, or something like WinSCP. Use the file transfer method that you are most familiar with.

Next, import the copied file into HDFS. This process can be done by entering the following command on the management node:

```
$hadoop fs -put source.data destination.data
```

In this example, “destination.data” can be any destination on the file system. It is possible to specify the host and port being copied to, by using `hdfs://host:port/hadoop/directory/file`. For example, to copy the entire directory of a log, you might run the following command:

```
$hadoop fs -put /var/www/ibm/logs hdfs://test.ibm.com:9090/hadoop/data/logs
```

8.1.3 Loading data from a data warehouse

Data from warehouses can be ingested to BigInsights by copying the data into the cluster. If your data is already exported into a file, it can either be copied through the command line or web-browser, as demonstrated in 8.3, “Web console” on page 115 and 8.4, “Text Analytics” on page 115.

It is also possible to use the database import application to extract information from your data warehouse by an `SQL` command, and store the results from this application in an HDFS location. One of the advantages to this approach is that you do not need the intermediate file to be created as we mentioned in the previous section. An example of this application is shown in Figure 8-4 on page 113.

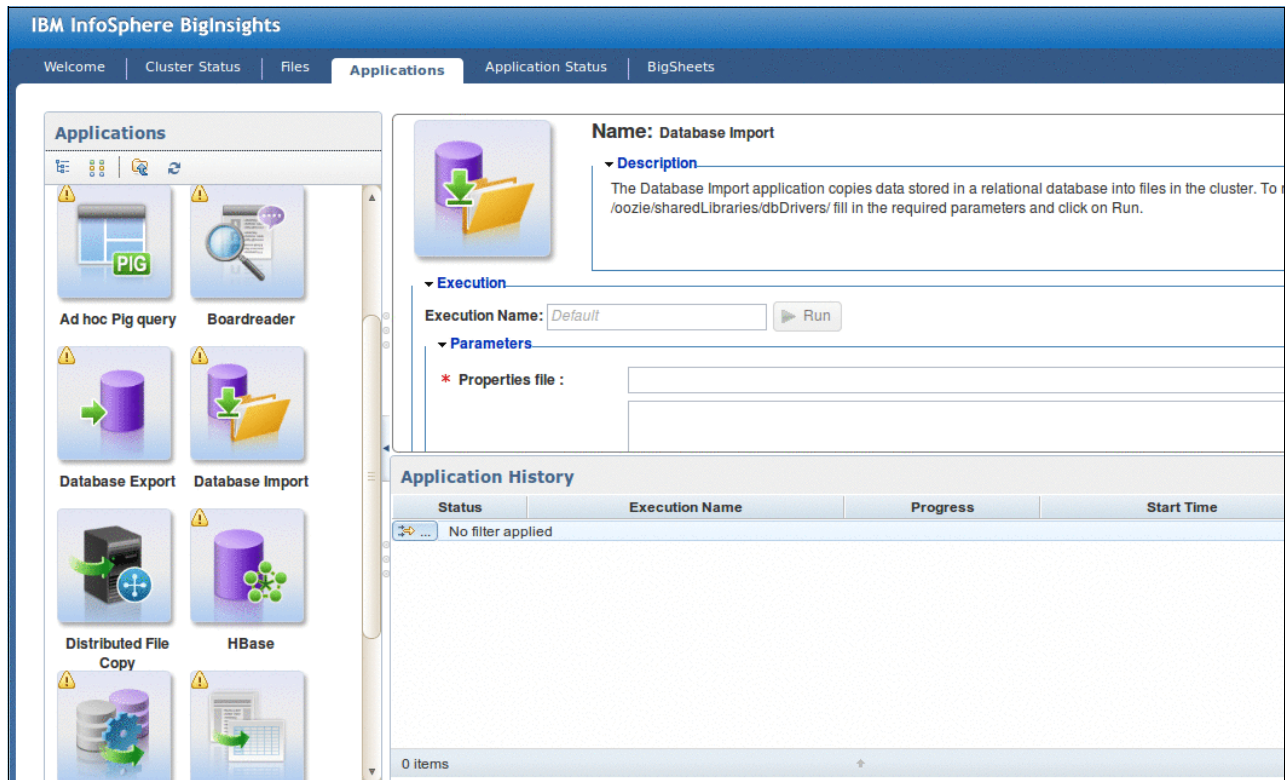


Figure 8-4 BigInsights database Import application

Because this SQL is passed into your data warehouse, you can create any SQL statement that suits your needs and is supported by the source database with which you are communicating. The resulting information is output either in the JavaScript Object Notation (JSON) or CSV format that is based on your selection in the interface. You must also select the wanted output folder.

8.1.4 Loading frequently updated files

Many applications can generate files with dynamic changing content. When you manage this data, it might be a requirement to access the updated data within BigInsights periodically. For this purpose, BigInsights includes the Flume component. *Flume* is an open source framework for collecting large data volumes and moving them around a cluster.

To use Flume from outside the cluster, a client-based runtime toolkit must be installed. It can be downloaded from the web console of BigInsights. The location to download the toolkit is shown in Figure 8-5.

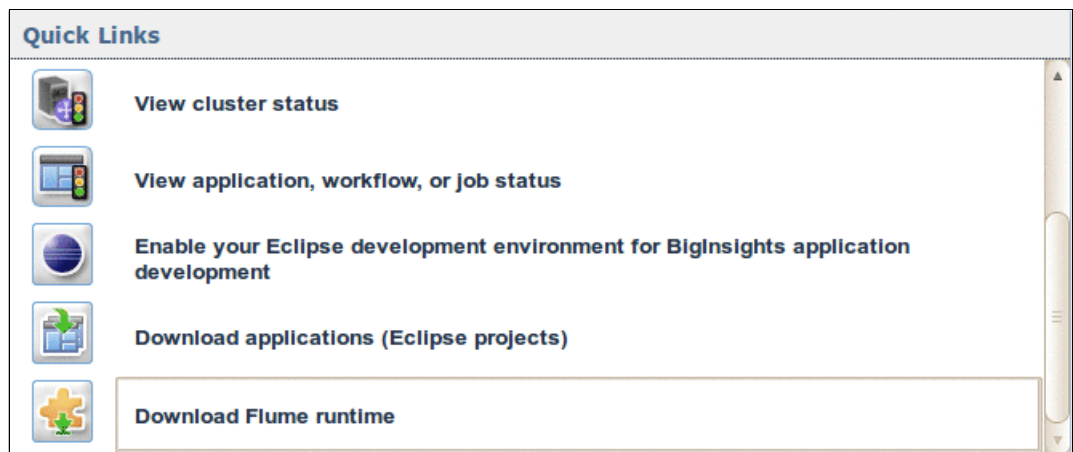


Figure 8-5 Download Flume run time

With this toolkit, you can start an agent outside of your cluster to collect log data.

Flume works with a set of logical nodes. Each logical node has two components, a source and a sink. The source is responsible for collecting data and the sink for sending it to a final destination. Because Flume is a distributed system with processes spread across several systems, it also uses the idea of a master node to control other nodes. Most of this process is already configured for you within BigInsights. To check the health of the Flume component, you can run the following command:

```
$/bin/sh $BIGINSIGHTS_HOME/bin/healthcheck.sh flume
```

The output of this script shows the process ID number for Flume processes and the IP address of the systems where Flume is running. If Flume is not running on your computer, start it by using the following command:

```
$/bin/sh $BIGINSIGHTS_HOME/bin/start.sh flume
```

When you are sure that Flume is running, there is a command to ensure that Flume is collecting data from your source. On our systems, we ran the following command as the user **biadmin**:

```
$ $BIGINSIGHTS_HOME/flume/bin/flume dump 'tail("/home/biadmin/example.log")'
```

This command displays the contents of the `example.log` file to the console and tracks new data that gets added to the `example.log` file. All new content that is inserted in `example.log` is going to be reflected in the console while this command is running.

To upload data into HDFS, we used the following command:

```
$ $BIGINSIGHTS_HOME/flume/bin/flume node_nowatch -l -s -n dump -c 'dump:asciisynth(100,100) | dfs("hdfs://<host:port>/user/path/example.txt");'
```

The `node_nowatch` parameter starts a node/agent to gather data, the `asciisynth` function generates data, and the `dfs` function writes the data to the path specified. In this case, we provided the host and port that matched our HDFS management node within our BigInsights cluster.

8.2 BigSheets

If you want to work with *big data* without writing code or scripts, you want to work with BigSheets. *BigSheets* is a spreadsheet-style tool for business analysts that is provided with the BigInsights Enterprise Edition. For a complete understanding of BigSheets, we provided a link to an article on [ibm.com](http://www.ibm.com/developerworks/data/library/techarticle/dm-1206socialmedia/index.html), in Example 8-1. This article teaches you the basics of using BigSheets to analyze social media and structured data that is collected through sample applications that are provided with BigInsights. You learn how to model this data in BigSheets, manipulate this data using built-in macros and functions, create charts to visualize your work, and export the results of your analysis in one of several output formats.

Example 8-1 URL to the BigSheets article on IBM developerWorks®

<http://www.ibm.com/developerworks/data/library/techarticle/dm-1206socialmedia/index.html>

8.3 Web console

If you are interested in getting off to a quick start with big data projects that involve IBM InfoSphere BigInsights, become familiar with its integrated web console. Through this tool, you can explore the health of your cluster, browse your distributed file system, start IBM supplied sample applications, monitor job and workflow status, and analyze data using a spreadsheet-style tool. The article on [ibm.com](http://www.ibm.com/developerworks/data/library/techarticle/dm-1204infospherebiginsights/) takes you on a tour of the web console, highlighting key capabilities that can help you get up to speed quickly. See Example 8-2 for the website of the article.

Example 8-2 URL to the web console article on developerWorks

<http://www.ibm.com/developerworks/data/library/techarticle/dm-1204infospherebiginsights/>

8.4 Text Analytics

Most of the world's data is unstructured or semi-structured text. The main objective of BigInsights Text Analytics is to extract business value from this type of data.

Text Analytics provides an information extractor, a text processing engine, and a library of annotators that enable developers to identify items of interest from unstructured or semi-structured text-based data. By extracting data of interest from unstructured sources, standard analytics can be run to gain value from data that was hidden before in its unstructured format.

8.4.1 Text analytics architecture

Before we dive into an actual example, a picture of the process might help to clarify the actions that we just described. Figure 8-6 displays the workflow that is typically used to create a Text Analytics application.

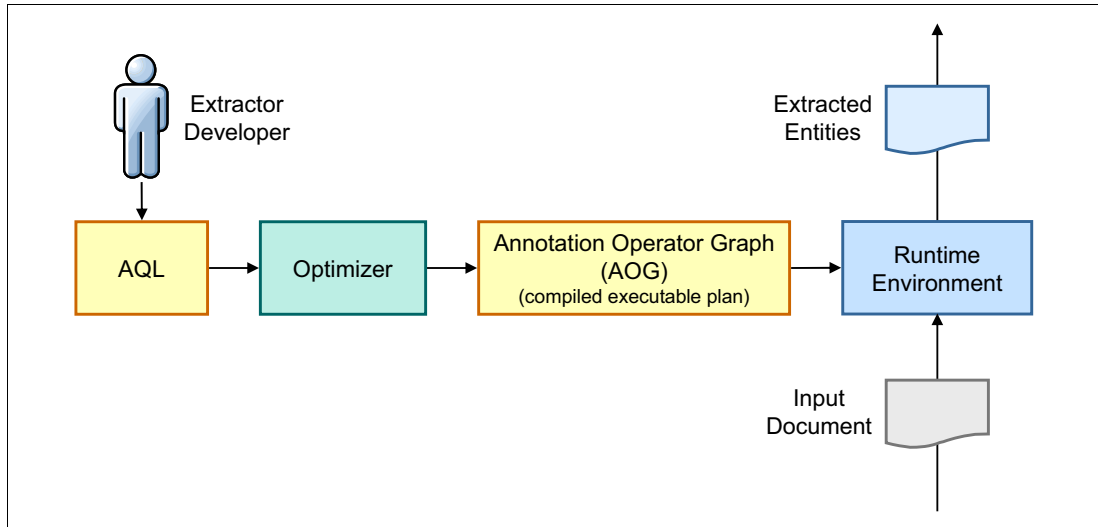


Figure 8-6 Overview of the text analytics architecture

Annotation Query Language (AQL) is the declarative language that is used in Text Analytics for defining extractors. An *extractor* is simply a program that is written in the AQL language. AQL compiles into internal algebra, creating an Annotation Operator Graph (AOG) file. This file consists of compiled, executable extractors for the BigInsights runtime environment.

In the text analysis process, AQL is used to define rules such that the application can extract information deemed important from unstructured text. Next, the optimizer looks at these defined rules and defines the best combination of annotators to be used to process the text most efficiently. If you, as the person writing the AQL, are looking for certain standard types of data, AQL includes built-in extractors. These extractors pull information such as ZIP codes, URL, names, phone numbers, or even financial information.

You might also have custom fields and formats within your existing text that a built-in processor might not detect. For this purpose, the Eclipse framework provides an integrated development environment for you to develop your own extractors. 6.4, “How to install the Eclipse plug-in” on page 88 shows how to download the Eclipse plug-in.

BigInsights provides an open tool, through an Eclipse-based plug-in, to write custom rules for your particular use case. Developers create rules by defining extractors in AQL, test the AQL against sample data within Eclipse, compile the AQL into an AOG file, and then run the AOG against files that are stored within the cluster.

8.4.2 Log file processing example

We now create a Text Analytics project for processing log data. BigInsights provides a step-by-step wizard-like tool to help you create and run the project. Default AQL skeletons are automatically generated, and tools to help you construct regular expressions or pattern matches are also available.

Creating a Text Analytics project in Eclipse

To create a Text Analytics project in Eclipse, click **Create an extractor in a new BigInsights project**, as shown in Figure 8-7.

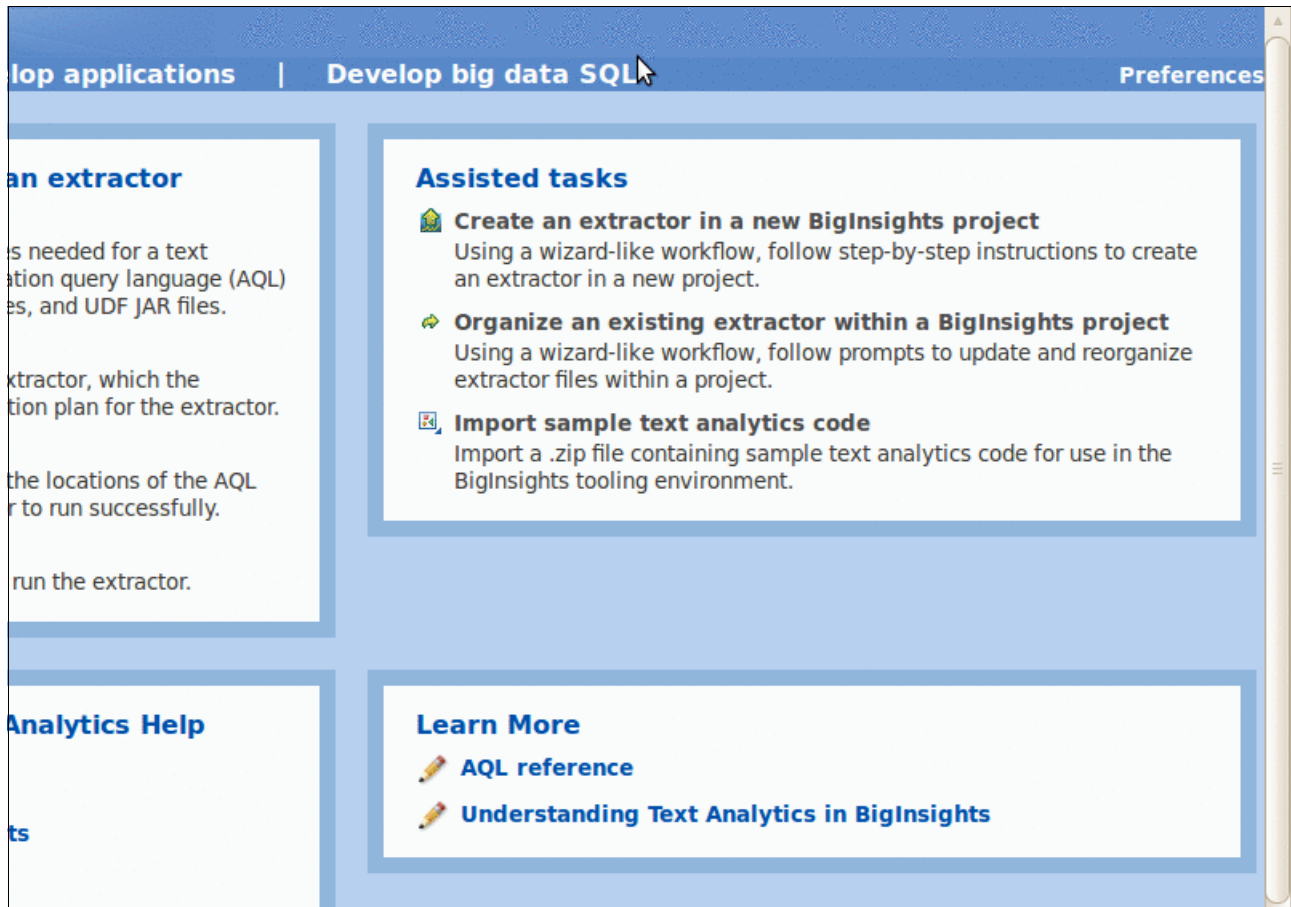


Figure 8-7 Creating Text Analytics project link

After you name the project, you will be presented with the BigInsights perspective. In the left menu, you find the **Extraction Task**, which is a list of tasks and controls to help you build and test your extractor. In the menu on the right side, you find the **Extraction Plan**, a plan to help organize and move through the project and AQL. An example can be seen in Figure 8-8.

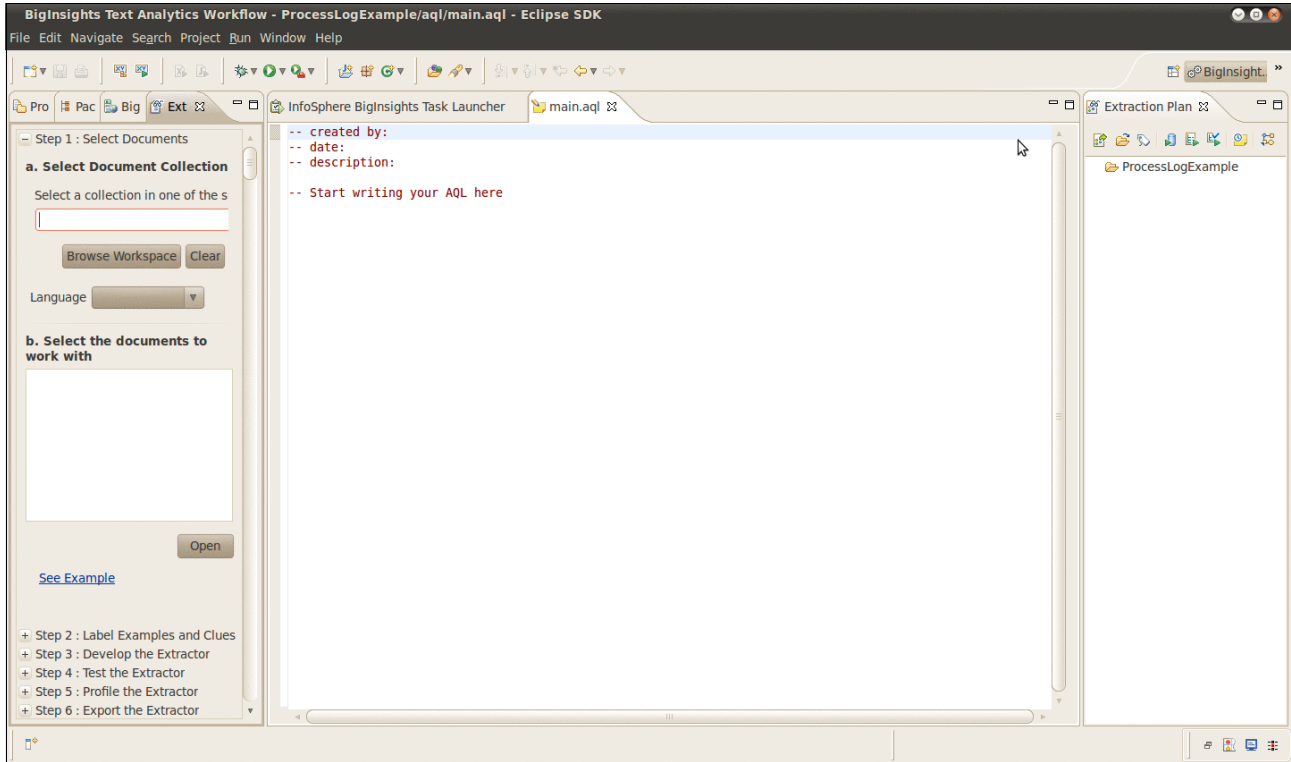


Figure 8-8 BigInsights perspective with left and right menus

The project structure is created automatically, as shown in Figure 8-9.

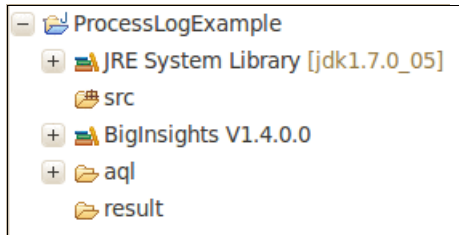


Figure 8-9 Project structure that is automatically created

Before you start the procedure to create a Text Analytics application, upload some text data to be analyzed. You can create a folder and name it, for example, *data* by right clicking the project name and selecting **New -> Folder**. When the input folder is created, you can add data by copying a file, right clicking the input folder, and selecting **Paste**. An example window of this activity is shown in Figure 8-10.

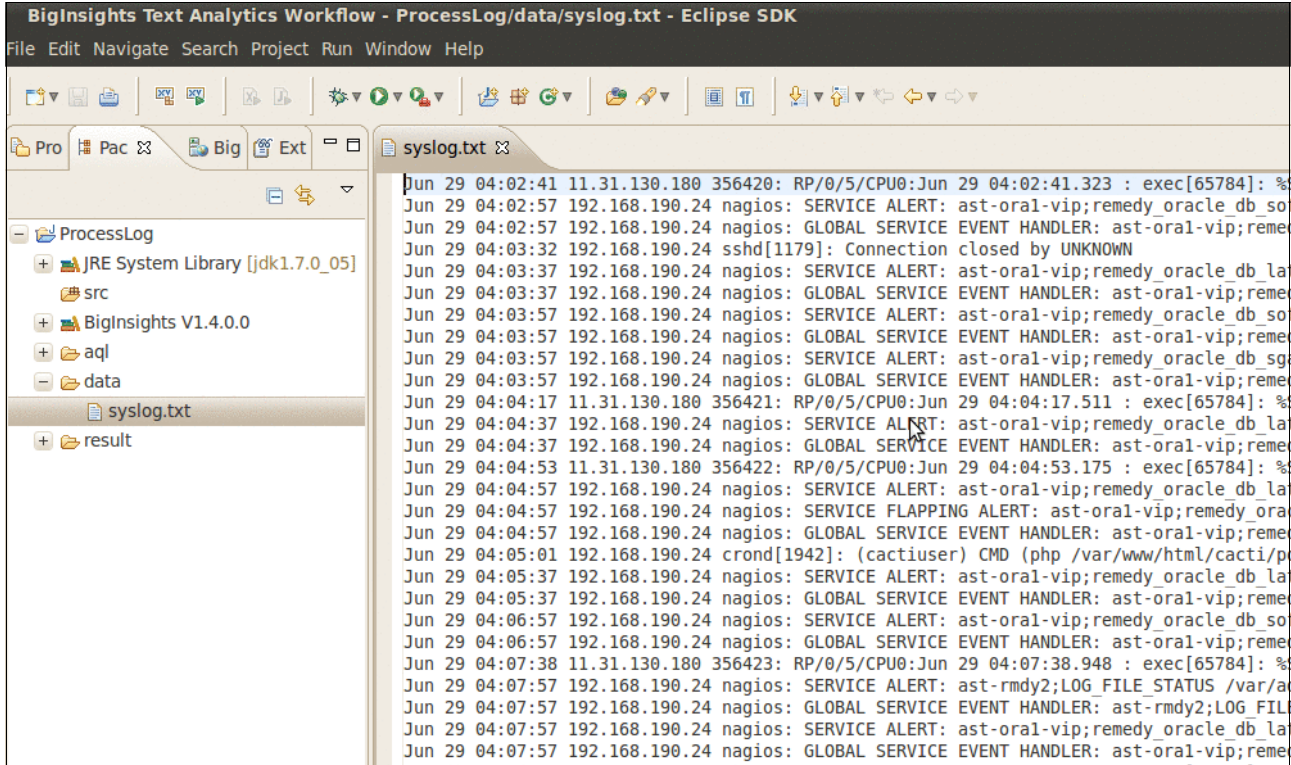


Figure 8-10 Input example

Step-by-step workflow project construction

The BigInsights perspective provides the *Extraction Tasks* pane, which guides you through a six-step process to develop, test, and export extractors for deployment. These six steps are shown in Figure 8-11.

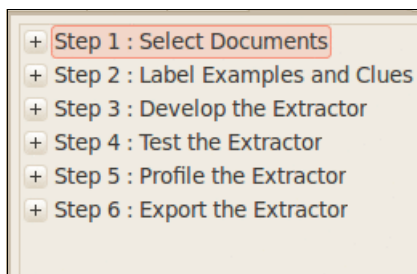


Figure 8-11 Steps to create project

1. Select the document collection that you are interested in. Choose among documents that are already inside the project folder. You can then select a language for the document. An example of this step is shown in Figure 8-12.

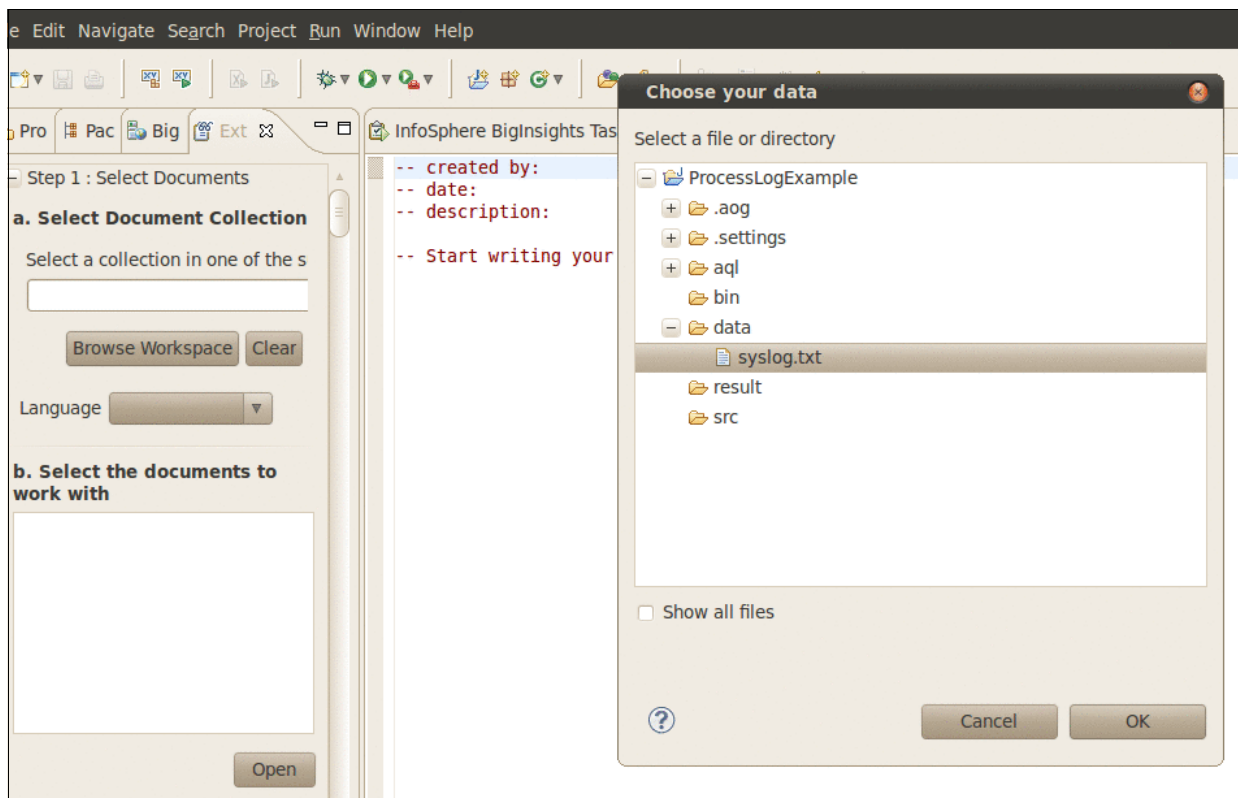


Figure 8-12 Step 1: Loading files

Select the wanted document and click **Open**.

2. Label examples and clues. Here is where you highlight to BigInsights what is your point of interest in the text. In this example, we are processing log files, finding dates, times, IP addresses, and log message details.

To create examples, select the interesting parts for each of these items, right click it and select **Add Example with New Label**. An example of adding a label is shown in Figure 8-13.

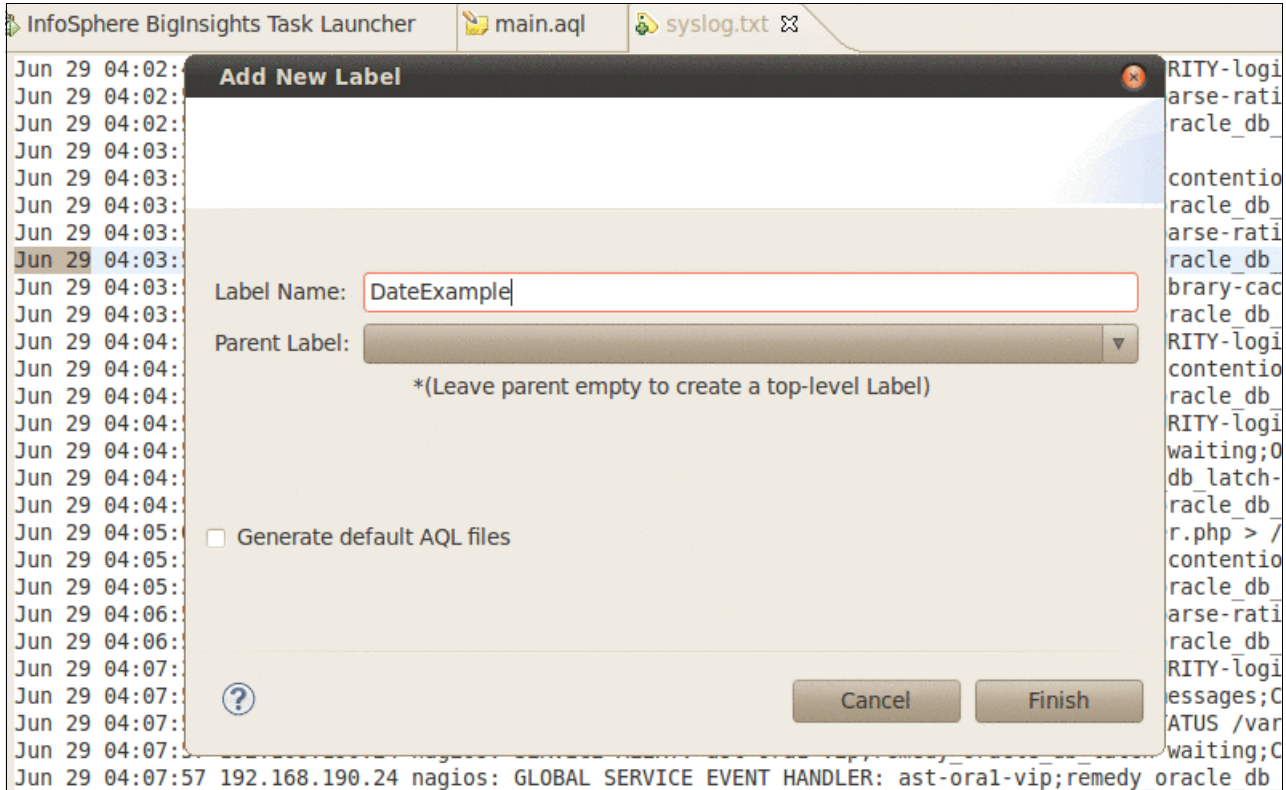


Figure 8-13 Adding example

If you select **Generate default AQL files**, a standard AQL file is automatically created for you.

3. Start developing the extractor. We have several tools to help you do this function. For example, the *Regular Expression Generator* tool and the *Pattern Discovery* tool are useful when you develop new extractors. These tools can both be seen in the **Extraction Tasks** menu.

To build an AQL file, right click the **Basic Feature** component in the **Extraction Plan** menu. An example of building an AQL statement is shown in Figure 8-14.

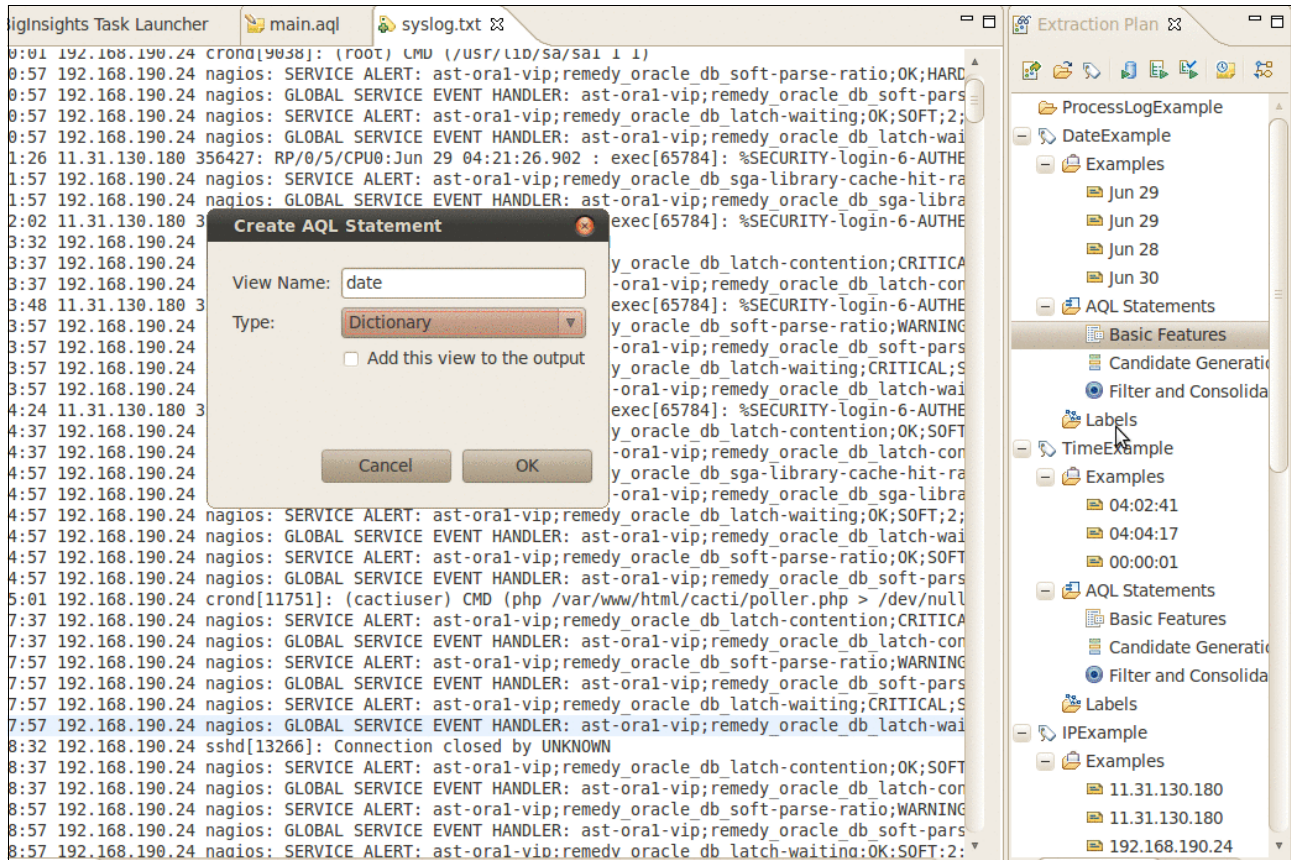


Figure 8-14 Creating an AQL statement

Click **Add AQL Statement** and you are prompted for the standard AQL file that is generated in step two.

Now can construct your rules to extract details from the text file. In Figure 8-15 on page 123, you can see the AQL-file that is used to extract Date, IP address, and the log details information.

```

-- Start writing your AQL here

create dictionary MonthDict
as ('Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec');

create view Month as
extract dictionary 'MonthDict'
  on R.text as match
from Document R;

create dictionary DayDict
as ('1','2','3','4','5','6','7','8','9','10',
    '11','12','13','14','15','16','17','18','19','20',
    '21','22','23','24','25','26','27','28','29','30','31');

create view Day as
extract dictionary 'DayDict'
  on R.text as match
from Document R;

create view Date as
extract pattern <M.match> <D.match>
  return group 0 as match
from Month M, Day D;

-- Simple Regex's to give us time and IP Address
create view Time as
extract regex /\d{2}:\d{2}:\d{2}/
  on R.text as match
from Document R;

create view IPAddress as
extract regex /(?:\d{1,3}\.){3}\d{1,3}/
  on R.text as match
from Document R;

```

Figure 8-15 Step 3: creating AQL files

You might notice that AQL has an SQL-like syntax. In AQL, you create a view for the wanted component, and then either a dictionary, a regular expression, or a pattern from where BigInsights knows what to look for in the text.

Example 8-3 shows the AQL that was created to look for month name abbreviations inside the log file. First, the system created a dictionary with the possibilities for a new item named MonthDict. Then, a view was created, named Month, to find a pattern that matches some occurrence in the MonthDict dictionary.

Example 8-3 Sample dictionary and view statement in AQL

```

create dictionary MonthDict
as ('Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec');

create view Month as
extract dictionary 'MonthDict'
  on R.text as match
from Document R;

```

In Example 8-4, two extractors are combined into one view. We used the AQL for Day and the AQL for Month and combine them into a single view called Date.

Example 8-4 Sample AQL for combining two views

```

create view Date as

```

```

extract pattern <M.match> <D.match>
return group 0 as match
from Month M, Day D;

```

In Example 8-5, we used a regular expression to find the string representing time. In this example, you will see a regular expression as the string shown after the keyword **regex**. Do not worry if you do not know how to work with regular expressions. A Regular Expression Builder tool is provided within BigInsights to assist you with building these expressions. Using AQL, we create a view called Time. This view extracts any combination that matches the pattern nn:nn:nn, where “n” is any number from 0 to 9.

Example 8-5 Sample AQL to extract time from a text file

```

create view Time as
extract regex /\d{2}:\d{2}:\d{2}/
on R.text as match
from Document R;

```

When the AQL is done, we can continue on to step 4 and run the program. In Figure 8-16, we display the results of the extracted log file.

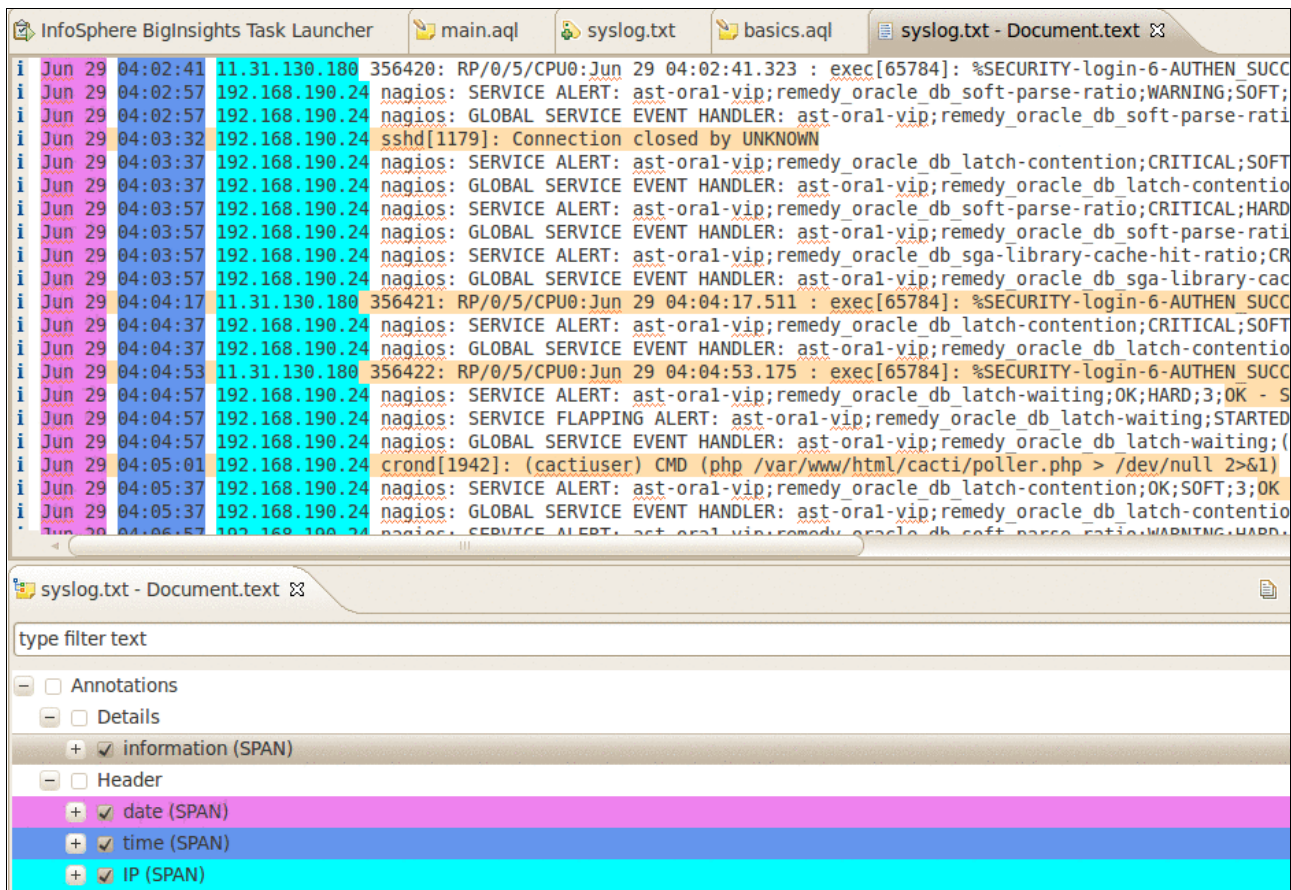


Figure 8-16 Resulting extracted log file

In this example, you see the highlights of only what was important for us to collect. The tool is flexible enough to detect nearly any item worth extracting. Keep in mind, this is just an example of what it is possible within BigInsights today. You can extract data from any other part of the log or from other types of files by using this example approach to text analytics.



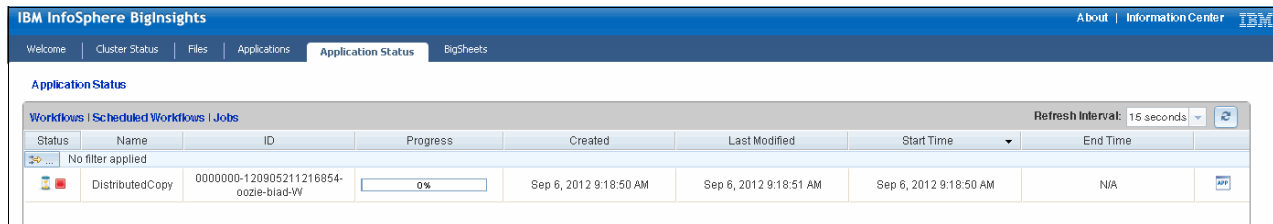
BigInsights hardware monitoring and alerting

One of the more difficult tasks in a clustered environment is understanding the state of the cluster. Hadoop is a dynamic environment with many tunable settings. Verifying your cluster's configuration, utilization, and stability is of paramount significance as environments scale from tens of nodes to hundreds, or even thousands, of nodes. As the size of the cluster increases, it is important to diagnose events as they happen and also verify that the cluster is being used to its maximum performance capabilities. To achieve this performance, administrators can use various different software tools. This chapter covers the following tools:

- ▶ *BigInsights Application Console*. A graphical user interface that is provided by BigInsights to monitor MapReduce jobs and allows users to drill down to various details of the run.
- ▶ *Nigel's monitor (nmon)*. A low-level tool that provides the status of IO, disk, processor, and memory usage.
- ▶ *Ganglia*. An open source software monitoring tool to provide cluster level insight regarding a number of hardware and OS variables called metrics. BigInsights also supports a plug-in into ganglia to monitor a number of hadoop-based metrics.
- ▶ *Nagios*. An open source event monitoring application to alert users and administrators of potential problems in the cluster.
- ▶ *IBM Tivoli® OMNIBus and Network Manager*. Designed to provide a cost-effective Service Assurance solution for real-time network discovery, network monitoring, and event management of IT domains and next-generation network environments.
- ▶ *IBM System Networking Element Manager*. An application for remote monitoring and management of Ethernet switches from IBM.

9.1 BigInsights monitoring

The BigInsights web console provides an interface to monitor MapReduce activities. The web console provides an intuitive interface to view the currently running jobs within the cluster and previously completed jobs. To drill down to job level details, BigInsights provides an intuitive interface to move through job settings and statistics. The web console provides three subcategories under the Application Status tab. The subcategories are workflows, scheduled workflows, and jobs as shown in Figure 9-1.



The screenshot displays the 'Application Status' tab in the IBM InfoSphere BigInsights web console. The page has a blue header with the title 'IBM InfoSphere BigInsights' and navigation links for 'About' and 'Information Center'. Below the header, there are tabs for 'Welcome', 'Cluster Status', 'Files', 'Applications', 'Application Status', and 'BigSheets'. The 'Application Status' tab is active, showing a sub-tab for 'Workflows | Scheduled Workflows | Jobs'. A table lists workflow entries with the following data:

Status	Name	ID	Progress	Created	Last Modified	Start Time	End Time
No filter applied							
	DistributedCopy	0000000-120905211216854-oozie-biad-W	0%	Sep 6, 2012 9:18:50 AM	Sep 6, 2012 9:18:51 AM	Sep 6, 2012 9:18:50 AM	N/A

Figure 9-1 Application Status tab view

9.1.1 Workflows and scheduled workflows

Workflows and scheduled workflow entries are logged when either an application is started from the Applications tab of the BigInsights web console or from the command line. Workflow entries represent oozie-generated workflows. The MapReduce jobs can be tracked by the Jobs link. Scheduled workflows represent the recurring workflows that are configured by using the web console. As these workflows are executed, the entries on the Scheduled workflows tab maintain a list of all the associated workflows. When clicked, it shows the workflows that are associated with the Scheduled workflow.

Figure 9-2 shows an example of a BigInsights distributed copy application that triggers a Scheduled workflow.

The screenshot displays the IBM InfoSphere BigInsights web interface. On the left is a sidebar with application icons: HIVE, JAQL, PIG, Boardreader, Database Export, Database Import, Distributed File Copy, HBase, TeraGen-TeraSort, Web Crawler, and Word Count. The main area shows the configuration for the 'Distributed File Copy' application. It includes a description, an execution name 'distcp_example', and parameters for input and output paths. A 'Schedule and Advanced Settings' section shows a scheduled job with a start date of September 6, 2012, at 09:10, a frequency of 5 minutes, and an until date of September 6, 2012, at 10:00. Below this is an 'Application History' table with the following data:

Status	Execution Name	Progress	Start Time	Elapsed Time (sec)	Details
No filter applied					
	distcp_example	0%	Sep 6, 2012 9:18:50 AM	280	
	distcp_example	100%	N/A	N/A	

Figure 9-2 Example of a Distributed File Copy Application with a scheduled task

When the application is deployed, the task is registered and tracked by using the scheduled workflows view. An example of how to drill down to lower levels of detail is shown in Figure 9-3.

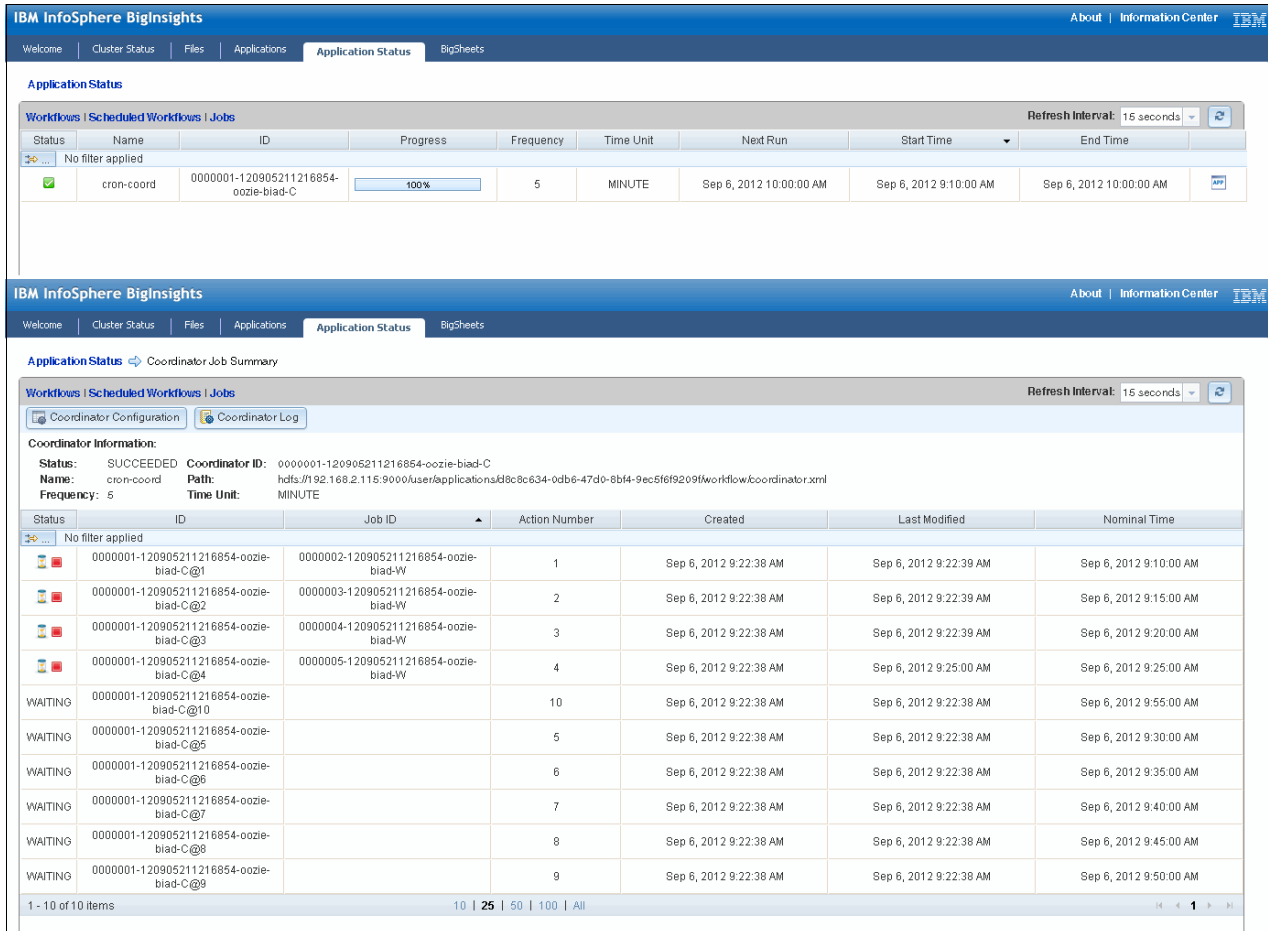


Figure 9-3 Tasks that are logged in the Scheduled Workflow of the Applications Status tab

9.1.2 MapReduce jobs

As described so far, the Workflows and Scheduled Workflows views are useful for viewing oozie-based workflows and tracking scheduled jobs that are created either by a user of the BigInsights Applications tab or a user submitting a job from the command line. To monitor the status of the currently running MapReduce jobs and also review previously completed MapReduce jobs, the **Application Status -> Jobs** view is useful.

The view that is shown in Figure 9-3 allows the user to drill down to see more details about the job. This area of the BigInsights console shows the status of the job, pass and fail information, and detailed task information.

Figure 9-4 shows an example of a typical drill-down, click path to show how to access the details and levels that BigInsights provides.

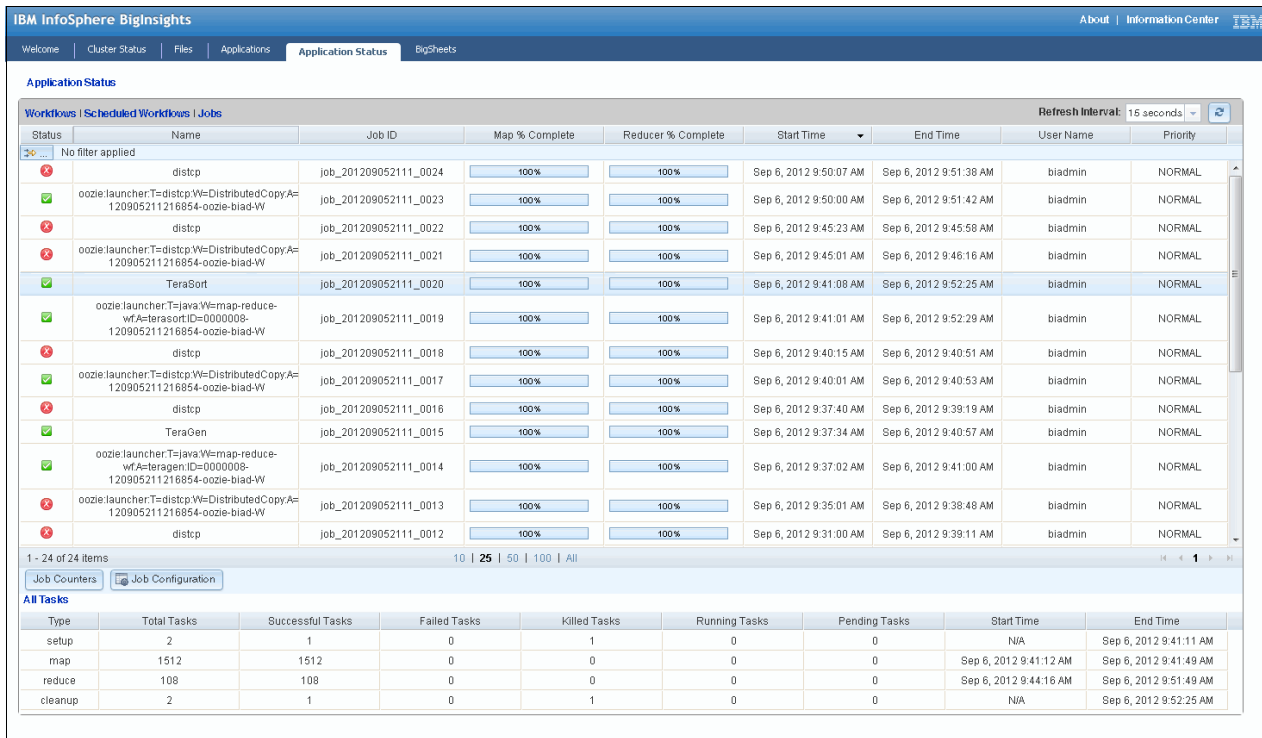


Figure 9-4 MapReduce job results within the BigInsights console

9.1.3 Job and task counters

To debug certain aspects of the job such as Hadoop Distributed File System (HDFS) usage, temporary space usage, cluster utilization, network traffic, and data locality, it is often useful to query the job counters at the end of the execution of a job. This process can be done by clicking the **Job Counters** icon, as shown in Figure 9-4. A few examples of items to verify after a job completes include the following questions:

- ▶ Were the correct number of maps and reduce tasks started?
- ▶ Does the amount of HDFS memory that is read in match expectations?
- ▶ Did the number of input records (key value pairs) match expectations?
- ▶ Did the amount of data that is written to HDFS make sense?

Figure 9-5 shows an example of Hadoop job counters.

Job Counters	
Job Counters	
Name	Value
DATA_LOCAL_MAPS	1509
RACK_LOCAL_MAPS	3
SLOTS_MILLIS_MAPS	99943954
TOTAL_LAUNCHED_MAPS	1512
FALLOW_SLOTS_MILLIS_REDUCE	0
FALLOW_SLOTS_MILLIS_MAPS	0
TOTAL_LAUNCHED_REDUCE	108
SLOTS_MILLIS_REDUCE	51224319
FileSystemCounters	
Name	Value
HDFS_BYTES_READ	200085127976
FILE_BYTES_WRITTEN	612049497445
FILE_BYTES_READ	418056532854
HDFS_BYTES_WRITTEN	200000000000
File Output Format Counters	
Name	Value
BYTES_WRITTEN	200000000000
Map-Reduce Framework	
Name	Value
VIRTUAL_MEMORY_BYTES	553409033420
REDUCE_INPUT_GROUPS	2000000000
COMBINE_OUTPUT_RECORDS	0
MAP_OUTPUT_RECORDS	2000000000
CPU_MILLISECONDS	101650490
MAP_INPUT_RECORDS	2000000000
REDUCE_SHUFFLE_BYTES	20391331490
COMBINE_INPUT_RECORDS	0
SPILLED_RECORDS	6000000000
SPLIT_RAW_BYTES	192024
MAP_OUTPUT_BYTES	200000000000
MAP_INPUT_BYTES	200000000000
REDUCE_INPUT_RECORDS	2000000000
PHYSICAL_MEMORY_BYTES	77261595852
COMMITTED_HEAP_BYTES	63291219763
REDUCE_OUTPUT_RECORDS	2000000000
MAP_OUTPUT_MATERIALIZED_BYTES	20400097977
File Input Format Counters	
Name	Value
BYTES_READ	200084935952

Figure 9-5 Hadoop job counters example from a 200 GB Terasort job

By scanning this table, we can answer a few of the questions that are shown in Example 9-1.

Example 9-1 Sample counter validation for a TeraSort run

DATA_LOCAL_MAPS equals 1509. The entire job had 1512 map tasks. A high degree of locality was achieved with this run.

RACK_LOCAL_MAPS equals 3. There were 3 map tasks that ended up running on nodes where the data was not present and required network transfer.

HDFS_BYTES_READ equals 200 GB . This is the data read in during map phase.

HDFS_BYTES_WRITTEN equals 200 GB . This is the data written in during reduce phase.

Launched map tasks equals 1512. This happens to be 2TB of data divided by the 128MB block size

Launched reduced tasks 108 - expected number based on job submittal parameter
Map input records - 2 billion - expected number of row
Map output records - 2 billion - 1:1 input to output ratio
Spilled records - 2 billion - the number of records written to temp space
Reduce input records - 2 billion - input from map output
Reduce output records - 20 billion - sorted number of rows written to HDFS

9.2 Nigel's monitor

Nigel's monitor (*nmon*) is a general purpose, debug utility to monitor processor utilization, memory usage, disk IO parameters, and so on, when using a nmon-supported operating system (OS). For information about nmon-compatible OSs, and a full list of items that are monitored, refer to the official nmon link in IBM developerWorks:

http://www.ibm.com/developerworks/aix/library/au-analyze_aix/

For our cluster, we elected to obtain the **nmon** binary for Red Hat 6.2. This is a prebuilt binary that was downloaded from <http://pkgs.repoforge.org/nmon>. The following *rpm* was downloaded:

nmon-14g-1.e16.rf.x86_64.rpm.

When the rpm was downloaded to the /tmp directory, we typed the following command from within the /tmp directory as root:

```
rpm -Uvh nmon-14g-1.e16.rf.x86_64.rpm
```

By default, this installs nmon into the /usr/bin directory.

Nmon has two modes of operation. One mode provides a real-time display that is shown within a terminal window. The other mode runs nmon as a background process that samples the system parameters and writes those parameters to an ascii-based file for processing and analysis later. Both modes are useful when probing the OS. The next sections describe how this tool can be used during a Hadoop run to identify potential performance-related bottlenecks.

9.2.1 nmon within a shell terminal

To start nmon, use the **/usr/bin/nmon** command.

When started, the **nmon** command displays an initial window. The following menu items are typically used for system evaluation:

- ▶ Processor menu **option c**
- ▶ Disk performance menu **option d**
- ▶ Network performance menu **option n**
- ▶ Memory usage memory **option m**

Some example windows from a typical nmon foreground session are shown in the following examples.

Figure 9-8 depicts 24 logical processors. User, wait, and system times are all tracked, along with an average utilization being maintained at the bottom of the window. In the IBM System x Reference Architecture, the systems use two, 6-core processors with two threads per core. This yields 24 logical processors, as shown in this view.

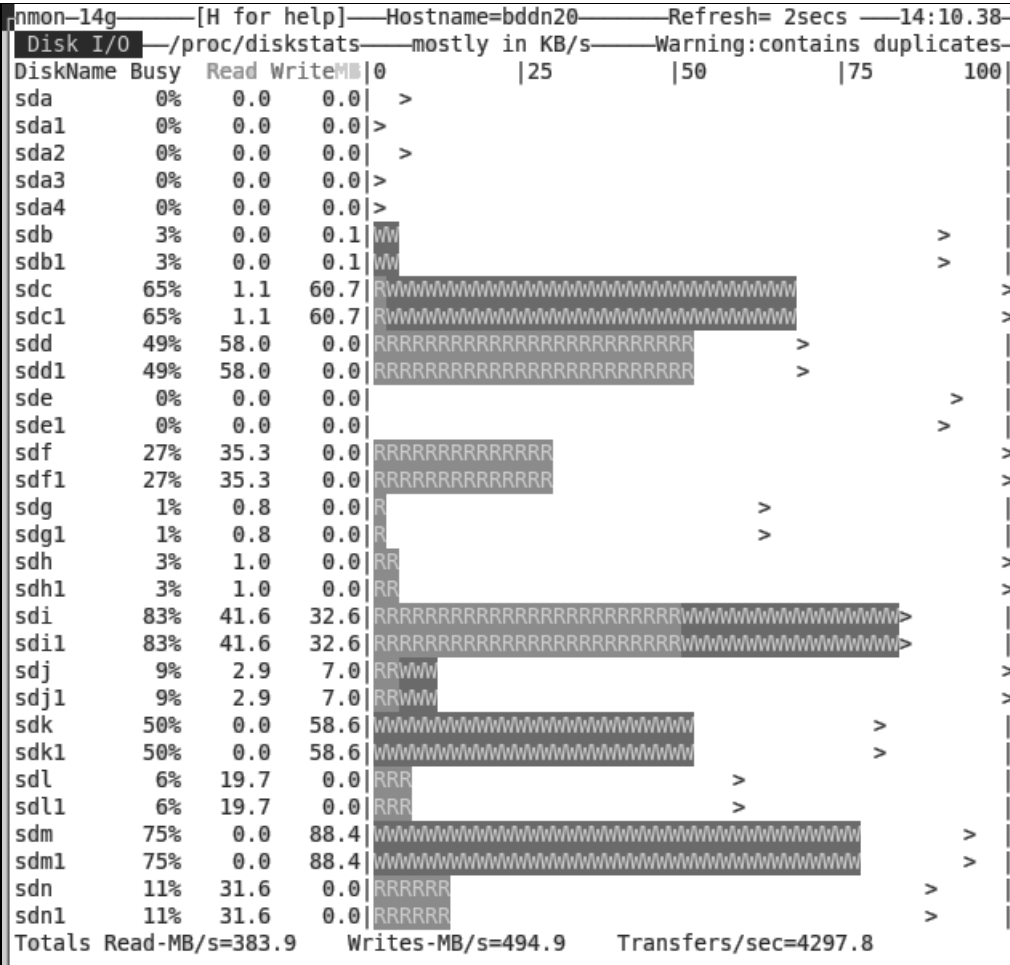


Figure 9-8 Disk utilization example using nmon

For the data node within the cluster that we used for this writing, there are 14, 3 TB drives. The display that is shown in Figure 9-8 shows the devices and device partition activity for both read and write operations. The > character shows the peak value that is achieved during the current monitoring session. An example of a memory utilization report can be seen in Figure 9-9.

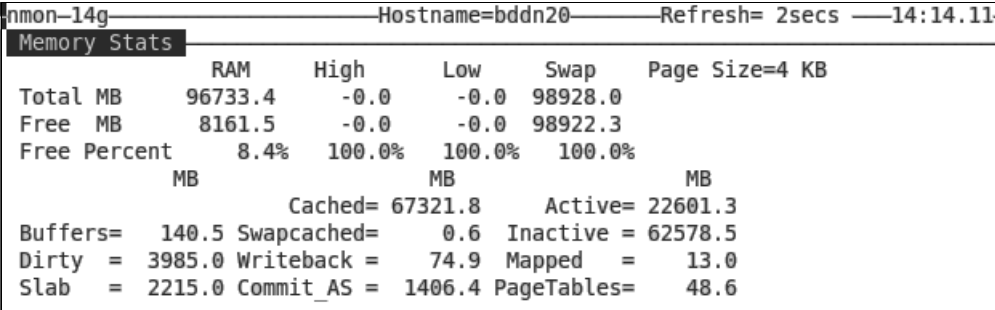


Figure 9-9 Memory utilization example using nmon

For Hadoop-based applications, it is important that the JVMs execute resident in memory. When the OS does an activity that is called *swapping*, the performance of Hadoop is usually affected in a negative way. In Figure 9-9 on page 133, the Free percent for swap should always be 100% for optimal performance (because this percentage indicates that swapping is not occurring). Also, it is a good idea to watch the amount of free memory. One common pitfall is to view only the Free MB row for the RAM column and assume that this number represents the amount of memory that is available to applications. However, you must also take into account other items such as cached pages. In the example in Figure 9-9 on page 133, even though there is 8161 MB of free memory, there is an additional 67321 MB of cached memory that is available to be reclaimed by other applications. As a way to make a rough guess, the real amount of free memory is the sum of the free memory and the cached memory.

free and **vmstat** commands: Other useful OS-level, command-line tools to monitor memory usage and swap are **free** and **vmstat**. These tools are started by issuing the **free** command and the **vmstat** command, in that order.

9.2.2 Saving nmon output to a file

Many times, administrators want to understand the variation of a job's execution over a time. Administrators might also want to see how a number of machines in the cluster are behaving simultaneously. For these types of situations, it is useful to take advantage of the **nmon** write-to-file feature. This function can be done by using a simple command-line flag, **-F**. The following example shows how to start the **nmon** command to be written to file named `outputFile`.

```
nmon -F /tmp/outputFile -c 5 -s 12
```

In this example, the administrator specifies the output directory for the file and also the number of samples (the **-c** flag value), and the number of seconds between samples (the **-s** flag value). After completion, the output file consists of five data samples for 60 seconds total. The output format of the text file is beyond the scope of this document; however, there is a Microsoft Excel spreadsheet template called `NmonAnalyzer` on the **nmon** page in IBM developerWorks that can parse the output of the text file into a formatted spreadsheet with excellent graphs. You can find a copy of the template at the following website:

http://www.ibm.com/developerworks/aix/library/au-nmon_analyser/

The write-to-file feature is useful when trying to monitor a cluster of servers. During the execution of Hadoop-based jobs, it can be useful to run the **nmon** command during times when the cluster is under a workload. You can then process the data when the jobs complete their execution to get a cluster-wide view of the performance-based information.

9.3 Ganglia

Ganglia is an open source tool that can be used to monitor many parameters of the cluster.

Ganglia versus nmon: We used Ganglia when monitoring our cluster during this writing. Unlike **nmon**, Ganglia is not provided by IBM and is part of the open source community of tools. Ganglia is not a required part of the solution and you can decide for yourself if Ganglia should be used on your cluster.

The Ganglia software is designed with scalability in mind. It is also designed to use minimum system resources while monitoring a cluster. Ganglia works with the round robin database (RRD) software. To find more information about Ganglia, see the following website:

<http://ganglia.sourceforge.net/>

An overview diagram of Ganglia is included in Figure 9-10. Ganglia consists of three components: A *monitor daemon*, a *collection daemon*, and a set of *PHP scripts* that interact with an HTTP server to enable web-based browsing of the status of monitored parameters.

The diagram that is shown in Figure 9-10, shows an example Ganglia layout with the *gmond* monitor daemons on all the nodes in our cluster. The *gmetad* collection daemon and PHP scripts are on the management node. Ganglia supports both unicast mode and multicast mode mechanisms for *gmond* to communicate with the *gmetad* daemon. In this implementation, the multicast mode of operation was used.

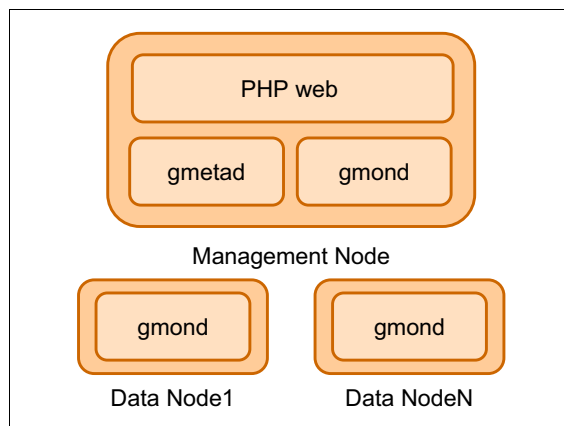


Figure 9-10 Example of the Ganglia layout on our BigInsights cluster

9.3.1 Ganglia installation (optional)

To install Ganglia on Red Hat Enterprise Linux 6.2, the *Yellowdog updaters modified (YUM)* installation utility was used. This software enables users to update software for many useful applications. One of the benefits of the tool is that software dependencies are managed automatically. When a repository is added to one of the YUM configuration files, YUM can automatically search and download supported packages. For Ganglia 3.1.7 on our cluster, the *Extra Packages for Enterprise Linux (EPEL)* repo was used. The EPEL package was added to the YUM repository by running the following command:

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-7.noarch.rpm
```

When the repository is added, the default Ganglia distribution can be downloaded. In this example, the Ganglia distribution that we used was Version 3.1.7. To install Ganglia, each node that requires system monitoring requires the *gmond* daemon to be installed. The following command was what we used to install ganglia *gmond*.

```
yum install ganglia-gmond.x86_64
```

When we installed the *gmetad* and the PHP web-based interface scripts on our management node, we used the following command:

```
yum install ganglia-gmetad.x86_64 # Ganglia Metadata collection daemon
yum install ganglia-web.x86_64 # Ganglia Web Frontend
```

9.3.2 Ganglia configuration (if installed)

When the Ganglia software is installed, the next step is to configure the daemons properly. To customize the configuration for gmond, edit the `gmond.conf` file in the `/etc/ganglia` directory. Ensure that the files are updated similarly across the cluster. The original configuration file was copied to `gmond.317.orig`, and the `gmond.conf` file was edited. When we performed the configuration steps on our cluster, we used the following command:

```
cp /etc/ganglia/gmond.conf /etc/ganglia/gmond.317.orig # perform on each node
```

In Example 9-2, we show an example modification that we performed on our `gmond.conf` file.

Example 9-2 A sample modification required for gmond.conf to set up Ganglia on the cluster

```
/*
 * The cluster attributes specified will be used as part of the <CLUSTER>
 * tag that will wrap all hosts collected by this instance.
 */
cluster {
  name = "hadoop_redbook"
  owner = "redbook"
  latlong = "na"
  url = "na"
}
```

Grouping subsets of nodes: If there are requirements to group subsets of nodes, then modify the `port = settings` in `gmond.conf`. By default, this value is 8649 but can be changed to any available port.

To customize the gmetad configuration, edit the `gmetad.conf` file in the `/etc/gmetad.conf`. For this cluster, the following modifications were made to the configuration files. The original configuration file was copied to `gmetad.317.orig`. We used the following command to do this file-copy process:

```
cp /etc/ganglia/gmetad.conf /etc/ganglia/gmetad.317.orig # management node only
```

There are a number of settings that can be changed in `gmetad.conf`, but to get started, the ones in Example 9-3 are the initial values that we set. For consistency in the `data_source` line, the label should be the same as the cluster name in the `gmond.conf` file. The *number* is the refresh interval, and the *node:port values* are the nodes that gmetad communicates with to gather the metrics data with the *first node:port* being the primary. And, the following `node:port` names are used for backup.

Example 9-3 Minimal configuration file that is needed for gmetad.conf

```
data_source "hadoop_redbook" 5 node0.mydomain.ibm.com:8649 node1.mydomain.ibm.com:8649
case_sensitive_hostnames 1
```

To start Ganglia, an HTTP server must be installed on the node where gmetad is located. Additionally, the Ganglia URL must be enabled by the `/etc/httpd/conf.d/ganglia.conf` file. If you choose to run Ganglia, it must be configured to comply with the HTTP server policies of your company.

To start Ganglia explicitly, we ran the following commands on each node where gmond was installed within our cluster (this assumes that the files were copied to the new names):

```
/usr/sbin/gmond -c /etc/ganglia/gmond.conf # perform on each monitor node
/usr/sbin/gmetad -c /etc/ganglia/gmetad.conf # perform on mgmt node only
```

To automate the Ganglia startup so that Ganglia starts upon restart, we used the command sequence, which is shown in Example 9-4. This sequence registered gmond and gmetad with the autostart mechanism and set the correct run level.

Example 9-4 Sample configuration to autostart Ganglia on Red Hat 6.2

```
#For each node where gmond is installed
chkconfig --add gmond
chkconfig --level 2345 gmond on
chkconfig --list gmond
==> gmond 0:off1:off2:on3:on4:on5:on6:off

#For the node where gmeta is installed
chkconfig --add gmetad
chkconfig --level 2345 gmetad on
chkconfig --list gmetad
==> gmetad          0:off1:off2:on3:on4:on5:on6:off
```

When these steps were performed, and the HTTP server was configured, the following example URL, <http://<management node IP address>/ganglia> produced a web page on our cluster that is similar to the window shown in Figure 9-11.



Figure 9-11 Example of our Ganglia default web interface

9.3.3 Multicast versus unicast

For quick deployments, *multicast* is the preferred choice. Also, for systems that have both administrator and private networks, it is a good idea to use the administrator network for the Ganglia packets because it does not affect the operation of the private network. Ensure that the default route for multicast packets is routed to the administrator network. One consideration for configuring Ganglia is the use of multicast versus unicast operating modes. Common issues involve the fact that some networks do not allow multicast packets because they can tend to use up more network bandwidth when compared to unicast packets. The unicast packets are broadcast to all the nodes within a certain configurable number of hops. Also, it is possible that corporate networks might disable the multicast capability. In this scenario, *unicast* mode must be enabled.

9.3.4 Large cluster considerations

The drawback with using multicast for very large clusters is that multicast can add extra network usage which might negatively affect network performance. In this scenario, a federated Ganglia setup would be used. Here, a subset of nodes is grouped to form a Ganglia subcluster; each having the gmond daemons reporting to the gmetad daemon in that subcluster. From here, each gmetad daemon can be configured to report to an aggregator gmetad daemon. In this way, a tree structure can be formed to have a single, top-level gmetad daemon that can be used to communicate the complete cluster information to the web-based interface. The following section focuses on the customization of a BigInsights cluster that is integrated into a Ganglia environment.

9.3.5 BigInsights 1.4 configuration to enable Hadoop metrics with Ganglia

To enable Hadoop metric reporting by using Ganglia, the `hadoop-metrics2.properties` file must be configured to enable the Hadoop Java daemons to emit metrics. To update the configuration file and broadcast the change through the cluster, the BigInsights cluster settings `synchronization` command must be used. The `synchronization` script publishes edits that are performed in a staging directory on the management node and updates the other nodes in the cluster. This function is useful because it saves the user time when managing and updating Hadoop configuration files across the cluster. The method that we used when configuring our cluster is described in the shaded box.

IP address: The IP address that is used in the `hadoop-metrics2.properties` configuration in Example 9-5 should be the same as the multicast address used in the `gmond` configuration file. It is not the IP address of the `gmeta` node.

To enable metrics for Ganglia, we made the changes that are shown in Example 9-5.

Example 9-5 Sample `hadoop-metrics2.properties` file to enable BigInsights Ganglia metrics

```
# syntax: [prefix].[source|sink|jmx].[instance].[options]

# Below are for sending metrics to Ganglia
# for Ganglia 3.0 support
# *.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink30
# for Ganglia 3.1 support
*.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink31
*.sink.ganglia.period=10
# default for supportsparse is false
*.sink.ganglia.supportsparse=true
*.sink.ganglia.slope=jvm.metrics.gcCount=zero,jvm.metrics.memHeapUsedM=both
*.sink.ganglia.dmax=jvm.metrics.threadsBlocked=70,jvm.metrics.memHeapUsedM=40
namenode.sink.ganglia.servers=239.2.11.80:8656
datanode.sink.ganglia.servers=239.2.11.80:8656
jobtracker.sink.ganglia.servers=239.2.11.80:8656
tasktracker.sink.ganglia.servers=239.2.11.80:8656
maptask.sink.ganglia.servers=239.2.11.80:8656
reducesetask.sink.ganglia.servers=239.2.11.80:8656
```

When the configuration file is modified, the Hadoop cluster must to be resynchronized and restarted to enable the metrics to be sent to the `gmond` daemon. We issued the following set of commands on our cluster to achieve the wanted results:

Note: `BIGINSIGHTS_HOME=/opt/ibm/biginsights`

```
$BIGINSIGHTS_HOME/bin/syncconf.sh hadoop force;.  
$BIGINSIGHTS_HOME/bin/stop.sh hadoop;.  
$BIGINSIGHTS_HOME/bin/start.sh hadoop
```

After that step was completed, the Ganglia configuration was verified by using the Ganglia web browser interface. The example URL for this example is the same one displayed earlier: <http://<management node IP address>/ganglia>.

The Hadoop metrics are organized under four different name spaces. These names are *jvm*, *dfs*, *mapred*, and *rpc*. Under each name space, there are many different values to query. The example lists, which are shown in “Ganglia monitoring options” on page 182, contain the parameters that can be viewed with the BigInsights V1.4 release.

9.4 Nagios

According to the following web page, *Nagios Core* is an open source system and network monitoring application:

<http://nagios.sourceforge.net/docs/nagioscore/3/en/about.html#whatis>

Nagios is an open source tool: We used Nagios when monitoring our cluster while writing this book. Unlike nmon, Nagios is not provided by IBM and is part of the open source community of tools. Nagios is not a required part of the solution. You can decide for yourself if Nagios should be used on your cluster.

Nagios watches hosts and services that you specify, alerting you when things go bad and when they get better.

Nagios Core was originally designed to run under Linux, although it works under most other UNIX versions as well.

Nagios Core offers the following features:

- ▶ Monitoring of network services (SMTP, POP3, HTTP, NNTP, PING, and so on)
- ▶ Monitoring of host resources (processor load, disk usage, and so on)
- ▶ Simple plug-in design that allows users to easily develop their own service checks
- ▶ Parallelized service checks
- ▶ Ability to define network host hierarchy using parent hosts, allowing detection of and distinction between hosts that are down and those that are unreachable
- ▶ Contact notifications when service or host problems occur and get resolved (using email, pager, or user-defined method)
- ▶ Ability to define event handlers to be run during service or host events for proactive problem resolution
- ▶ Automatic log file rotation
- ▶ Support for implementing redundant monitoring hosts
- ▶ Optional web interface for viewing current network status, notification and problem history, log file, and so on

For more information about Nagios Core and for example system requirements, see this website: <http://nagios.sourceforge.net/docs/nagioscore/3/en/about.html>

From our experience with the tool, we set it up to monitor critical areas within the cluster. Some examples include watching the remaining free space within HDFS and alerting the administrators to conditions that we configured for wanted warning levels. We found the approach to installing, configuring, and monitoring hosts and services to be helpful.

Writing about the process to install and configure *Nagios* is beyond the scope of this book. To learn more about Nagios and the available plug-ins, see the Nagios self-paced training page at this website: <http://library.nagios.com/training/selfpaced/topics/>

9.5 IBM Tivoli OMNibus and Network Manager

IBM Tivoli OMNibus and Network Manager V9.1 is designed to provide a cost-effective Service Assurance solution for real-time network discovery, network monitoring, and event management of IT domains and next-generation network environments. The customizable web-based user interface, enabled through the Tivoli Integrated Portal infrastructure, allows you to achieve end-to-end visualization, navigation, security, and reporting (real time and historical) across Tivoli and third-party management tools.

IBM Tivoli OMNibus and Network Manager V9.1 delivers event, network, and configuration management enabling network device security and control in a single offering. It gives you the opportunity to capitalize on the tight integration between *Tivoli Netcool/OMNibus*, *Tivoli Network Manager IP Edition*, and *IBM Tivoli Netcool® Configuration Manager*. These three products, which can be ordered separately, provide unrivalled visibility and control of the managed domain. As a joint offering, these three products provide the depth and breadth of management capability that is required, almost regardless of size.

Designed for use in both the smallest to some of the largest deployments, this fast-to-deploy solution offers around-the-clock event and network management with high automation to help you deliver continuous uptime of business, IT, and network services.

IBM Tivoli Netcool Configuration Manager V6.3 is the next generation of intelligent networking solutions for network-driven organizations so that they can control, manage, and scale their networks. Tivoli Netcool Configuration Manager V6.3 provides a best-in-class network automation solution for multivendor devices; for example, routers, switches, hubs, and firewalls. This powerful solution includes network configuration and change management, policy-based Compliance Management, and software upgrades.

9.5.1 Tivoli Netcool Configuration Manager

Tivoli Netcool Configuration Manager V6.3 helps you to enforce granular access control and security, automate configuration changes, enable network compliance, and accurately provision devices, virtually irrespective of vendor, type, model, or OS. Using IBM DB2 technology, Tivoli Netcool Configuration Manager is designed to manage data more effectively and efficiently. Greater availability is delivered through enhancements such as online, automated database reorganization. In addition, the increased scalability and the ability to use the latest in server technology helps deliver increased performance of backup and recovery processes.

With Tivoli Netcool Configuration Manager V6.3, you get a new pricing structure and three features to improve time to value, and improve the robustness and reliability of multivendor, critical networks.

9.5.2 Highlights of Tivoli Netcool Configuration Manager

- ▶ Launch key Tivoli Netcool Configuration Manager UI components within Tivoli Integrated Portal (for example, apply command set, revert config, policy check) to drive key operational workflows seamlessly within Tivoli Integrated Portal. This drives strong integrated workflows at the visualization layer across event, network management, configuration, and Compliance Management.

Supporting launch-in-context of Tivoli Netcool Configuration Manager UIs from OMNIBus and Network Manager to drive more advanced operational and administration-use cases.

- ▶ *Activity Viewer*. This tool allows you to visualize Tivoli Netcool Configuration Manager activities in a timeline viewer portlet to clearly see key Tivoli Netcool Configuration Manager activities.
- ▶ Ready-to-use compliance reports using Tivoli Common Reporting.
- ▶ IBM AIX® and SUSE platform support.
- ▶ Supports more databases, including DB2.
- ▶ Uses federal market enabler: FIPS 140-2.
- ▶ Virtualization
 - Network proxy and gateway support to enable management of networking components within proxy and gateway (for example, management of a vSwitch within a hypervisor or devices being managed by an Electronic Management System (EMS)).
 - Improved visualization of virtual devices.
 - Drivers for networking components within a virtualized solution (for example, configuration of vSwitches).
- ▶ Service activation

Service management interface (SMI) includes enhanced application programming interfaces (APIs) for network configuration management to support activation that is driven by external systems.
- ▶ Enhanced policy definition by enabling programmatic constructs that are supported by the introduction of Java scripting within compliance policies definitions.

Provides the ability to include external calls as a part of Java script policy definitions.

9.5.3 IBM Tivoli Netcool/OMNIBus

IBM Tivoli Netcool/OMNIBus V7.3.1 is designed to scale from the smallest to some of the largest, most complex environments across business applications, virtualized servers, network devices and protocols, Internet protocols, and security and storage devices. Breadth of coverage, rapid deployment, ease of use, high resilience, and exceptional scalability and performance are just some of the reasons why leading global organizations are using Tivoli Netcool/OMNIBus to manage some of the world's largest, most complex environments.

9.5.4 IBM Tivoli Network Manager IP

IBM Tivoli Network Manager IP Edition V3.9 helps an organization visualize and understand the layout of complex networks and the affect of events upon them. Its root-cause analysis allows network operations center (NOC) operators to work more efficiently by focusing time and attention on root-cause events and identifying symptom events that can be filtered into a separate view.

9.6 IBM System Networking Element Manager

Highlights of the *IBM System Networking Element Manager* include the following benefits:

- ▶ Improve network visibility and drive reliability and performance
- ▶ Increase the availability and performance of critical business services with advanced correlation, event de-duplication, automated diagnostics, and root-cause analysis
- ▶ Simplify management of large groups of switches with automatic discovery of switches on the network
- ▶ Automate and integrate management, deployment, and monitoring
- ▶ Simple Network Management Protocol (SNMP)-based configuration and management
- ▶ Support of network policies for virtualization
- ▶ Authentication and authorization
- ▶ Real-time root cause analysis and problem resolution
- ▶ Integration with IBM Systems Director and VMWare Virtual Center and vSphere clients

IBM System Networking Element Manager is an application for remote monitoring and management of Ethernet switches from IBM. It is designed to simplify and centralize the management of your IBM BladeCenter® or blade server and top-of-rack Ethernet switches.

9.6.1 Product features

- ▶ Easy to use
- ▶ Multiple licensing tiers
- ▶ Automatic switch discovery
- ▶ Multiple operating systems
- ▶ Wide variety of integration options

9.6.2 Software summary

- ▶ SNMP-based configuration and management of IBM top-of-rack and embedded switches
- ▶ Automatic discovery of all switches in network
- ▶ Supports network policies for virtual servers
- ▶ Integration with HP Systems Insight Manager 5.1+, IBM Systems Director 5.20+ and 6.x+, and VMware Virtual Center and vSphere client



BigInsights security design

IBM InfoSphere BigInsights security architecture includes web console authentication, roles and authorization levels, and HTTPS support for the BigInsights web console.

This chapter describes the security aspects of BigInsights. We cover roles and groups that are defined to provide authorization. We also include authentication features that are provided by BigInsights when configured to work with *Lightweight Directory Access Protocol* (LDAP), *Pluggable Authentication Modules* (PAMs), and *flat-file based authentication*.

10.1 BigInsights security overview

In the BigInsights cluster, the management node and data nodes interact with each other through a network. Usually, the management node, running the BigInsights web console, is accessed remotely from another system. It can either be a local area network within a corporate intranet, or an extranet that links-in partners, or the worldwide Internet. With business information traveling around the cluster, it is essential to verify the authenticity of the source and destination of data. Information might be shared among applications, but it must also be protected from unauthorized modifications. It is also necessary to prevent disclosure of private information.

To enhance security, BigInsights provides built-in features that can be configured during the installation process.

Security aspects: There are two security aspects to consider. The first is authentication, which makes sure a user is who they claim to be. The second aspect is authorization, determining what actions the user is authorized to perform within the system.

For certain ways to perform authentication in your BigInsights cluster, you might want to configure LDAP, flat-file, or PAM authentication. When configured, users that access the web console are prompted for a user ID and password. HTTPS is also provided to enhance security between the user's browser and the BigInsights web server.

Figure 10-1 shows the BigInsights security architecture. As described in 3.3, “Networking zones” on page 32, consider connecting all systems in your cluster over a private network because they have unrestricted communication with each other.

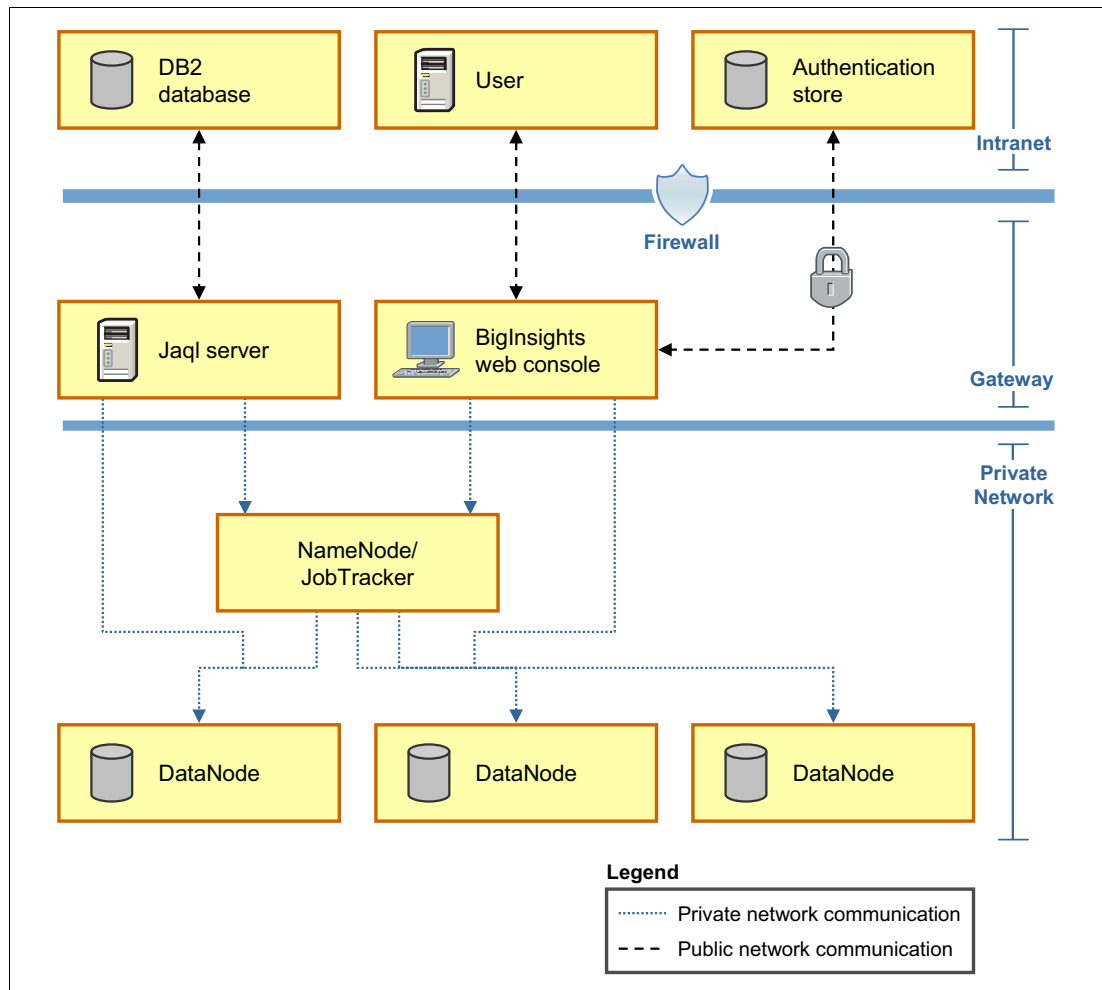


Figure 10-1 BigInsights security architecture overview

Furthermore, the integrity of data within the cluster can be potentially kept safe from unwanted actions or access through the roles and groups that are defined by BigInsights.

10.2 Authorization

Authorization is supported in BigInsights by defining roles. A role allows us to define what actions a specific user can or cannot perform. The web console content, for example, is dynamic and what a user sees is dependent on their role.

10.2.1 Roles

BigInsights supports four predefined roles. For each role, there is a set of actions that can be performed in the cluster. During the installation process, it is possible to map users and groups to these roles. They are:

- ▶ **BigInsightsSystemAdministrator.** Performs all system administration tasks. For example, a user in this role can perform monitoring of the cluster's health, and adding, removing, starting, and stopping nodes.
- ▶ **BigInsightsDataAdministrator.** Performs all data administration tasks. For example, these users create directories, run Hadoop file system commands, and upload, delete, download, and view files.
- ▶ **BigInsightsApplicationAdministrator.** Performs all application administration tasks, for example publishing and unpublishing (deleting) an application, deploying and removing an application to the cluster, configuring the application icons, applying application descriptions, changing the runtime libraries and categories of an application, and assigning permissions of an application to a group.
- ▶ **BigInsightsUser:** Runs applications that the user is given permission to run and views the results, data, and cluster health. This role is typically the most commonly granted role to cluster users who perform non-administrative tasks.

The `geronimo-web.xml` file maintains the mapping of roles to groups or users. In the `role-name` tag, the role is specified and the `name` attribute contains the group or user that is mapped to this role. The `geronimo-web.xml` should be in `$BIGINSIGHTS_HOME/console/conf/` directory. An example of its contents is shown in Example 10-1.

Example 10-1 Example of `geronimo-web.xml`

```
<sec:security>
<sec:role-mappings>
<sec:role role-name="BigInsightsSystemAdministrator">
<sec:principal class="" name="supergroup"/>
<sec:principal class="" name="sysAdmins"/>
</sec:role>
<sec:role role-name="BigInsightsDataAdministrator">
<sec:principal class="" name="supergroup"/>
<sec:principal class="" name="dataAdmins"/>
</sec:role>
<sec:role role-name="BigInsightsApplicationAdministrator">
<sec:principal class="" name="supergroup"/>
<sec:principal class="" name="appAdmins"/>
</sec:role>
<sec:role role-name="BigInsightsUser">
<sec:principal class="" name="supergroup"/>
<sec:principal class="" name="users"/>
</sec:role>
</sec:role-mappings>
</sec:security>
```

You can verify user groups and roles by typing the following command in the web console:

`http://<host>:<port>/data/controller/AuthenticationAction?actiontype=getUserInfo`

The `BigInsightsSystemAdministrator` is able to change permissions in files to allow or restrict the access of users. `BigInsights` follows the standard Access Control Lists standards that are used within POSIX file systems. These permissions can be seen if you run the `ls` command, as shown in Example 10-2.

Example 10-2 Viewing hadoop filesystem permissions

```
$hadoop fs -ls
Found 2 items
```

```
drwx----- - biadmin supergroup <rep count> <date> <time> <some directory name>
drwxr-xr-x biadmin supergroup <rep_count> <date> <time> <another directory name>
```

As an administrator of the cluster, you are able to change permissions and ownership with **chmod**, **chown**, or **chgrp** commands. Example 10-3 shows an example of the Hadoop commands that are used to perform these actions within HDFS.

Example 10-3 Examples of changes to permissions and ownership within HDFS

```
$ hadoop fs -chmod 755 myexample.txt
$ hadoop fs -chown admin:admingroup
```

10.3 Authentication

InfoSphere BigInsights provides four options for authentication. You can choose to have no authentication (the installation default that is shown in Figure 10-2, or chose between LDAP, flat-file, or PAM authentication.

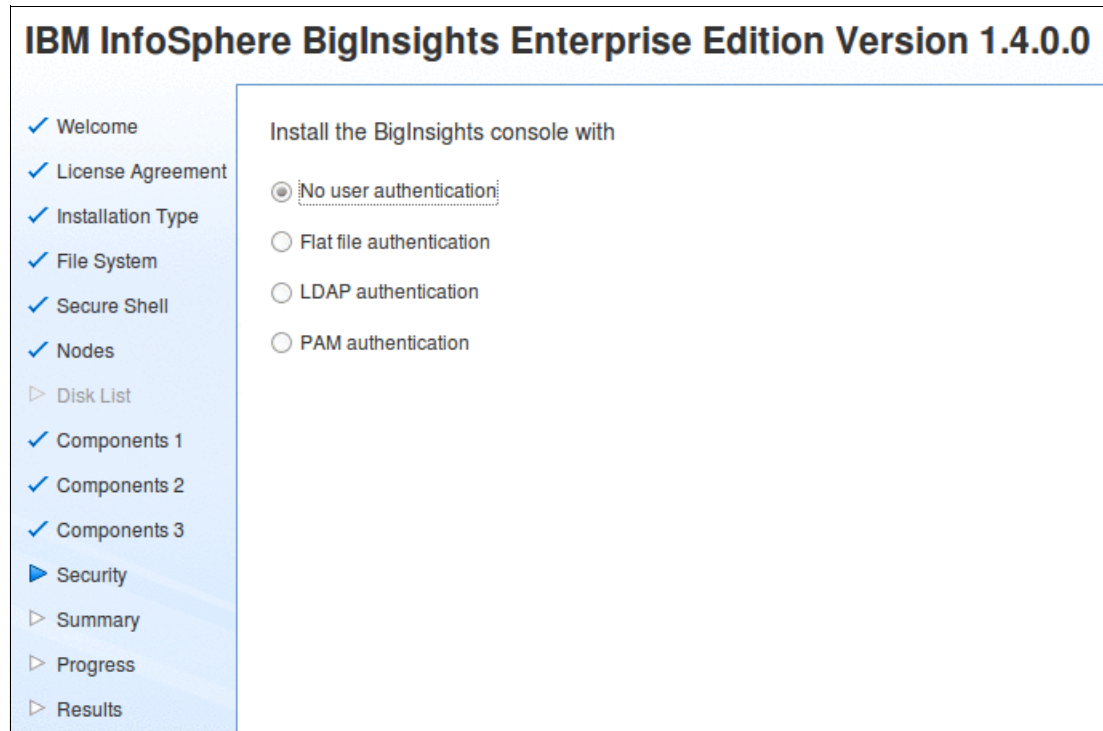


Figure 10-2 Authentication options

10.3.1 Flat file

Flat-file authentication uses a flat file to store authentication settings, such as user IDs and passwords. This authentication method allows you to select a default configuration or specify custom settings. The default configuration is not recommended in a production environment because your users' passwords and groups assignments are saved in a common file with no encryption. In this scenario, you can update users, passwords, and groups configuration before you install the product in the file and settings that are shown in Example 10-4 on page 150.

Example 10-4 Location of the default flat file for authentication pre-installation

```
/home/biginsights-enterprise-linux64_b20120604_2018/artifacts/security/flatfile/biginsights_user.properties  
/home/biginsights-enterprise-linux64_b20120604_2018/artifacts/security/flatfile/biginsights_group.properties
```

After the product is installed, you can manage these same authentication settings in the files that are shown in Example 10-5.

Example 10-5 Location of the default flat file for authentication post-installation

```
/opt/ibm/biginsights/console/conf/security/biginsights_user.properties  
/opt/ibm/biginsights/console/conf/security/biginsights_group.properties
```

It is also possible to customize the authentication settings by specifying where the users, groups, and passwords directory is going to be placed. And most importantly, you can also specify a cryptography method for storing the passwords. You can choose between the MD5 or SHA1 digest algorithms.

Customize flat file authentication settings
Please fill in the customized flat file for the groups and users, as well as the groups/users to role mappings. You can edit these files directly at any time.

Flat file authentication settings

Specify flat files that defines the groups and users for user authentication and access control.

* Users File URI:
The full path of a flat file that holds the user/password information. The format of each line should be username:password

* Groups File URI:
The full path of a flat file that holds the group information. The format of each line should be group=user,username

Digest Algorithm:
Message Digest algorithm (e.g. MD5, SHA1, etc.) used on the passwords. Select 'None' if no digest algorithm is specified.

Digest Encoding:
Encoding to use for digests (e.g. hex, base64). This is used only if a Message Digest algorithm is specified.

Figure 10-3 Flat file authentication installation options

After you specify the location of the configuration file, describe the mappings of the groups and users to the four predefined BigInsights roles. This step, as shown in Figure 10-4, is included within any authentication method you choose.

The screenshot shows a configuration window titled "Groups and Users". It contains four sections, each for a different role. Each section includes a description of the role, a "Groups:" field with a text input, and a "Users:" field with a text input.

- BigInsights System Administrator**: Users with this role perform all system administration tasks. The "Groups:" field contains "supergroup, sysAdmins".
- BigInsights Data Administrator**: Users with this role perform all data administration tasks. The "Groups:" field contains "supergroup, dataAdmins".
- BigInsights Application Administrator**: Users with this role perform all job management tasks. The "Groups:" field contains "supergroup, appAdmins".
- BigInsights User**: A non-administrative role for executing jobs and viewing results. The "Groups:" field contains "supergroup, users".

Figure 10-4 Example of user-to-group mappings within the install interface

As mentioned, these mapping settings are stored in the `geronimo-web.xml` file

10.3.2 Lightweight Directory Access Protocol

The Lightweight Directory Access Protocol (LDAP) defines a standard method for accessing and updating information in a directory. It provides a single sign-on facility where one password for a user can be shared between many services. This is a commonly used authentication protocol because it enables users to sign on to the cluster by using corporate user IDs and passwords.

LDAP authentication is initiated when a client starts an LDAP session by connecting to an LDAP server. This is known as *binding* to the server. The client continues by making a request (for example, by making a search, modify, or delete request). An LDAP credential store server should exist and be reachable through TCP/IP to continue the authorization process. The LDAP server then sends a response in return. When the client is done making requests, it then closes the session with the server.

The LDAP process is depicted in Figure 10-5.

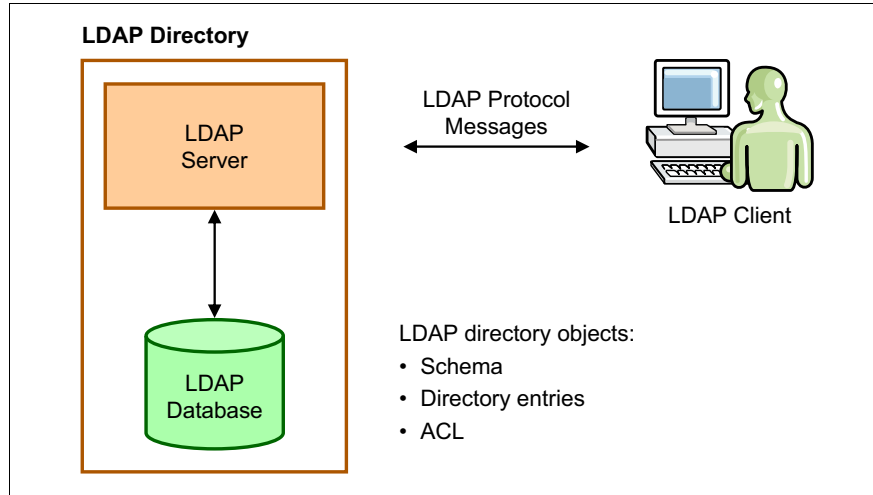


Figure 10-5 LDAP overview

In BigInsights, LDAP authentication is configured during the installation process. You must configure it to communicate with an LDAP credential store for users, groups, and user-group mapping. The settings of the LDAP installation depend on your LDAP server credentials. Figure 10-6 shows how we configured LDAP during our cluster installation.

The screenshot shows the "LDAP authentication" configuration window. The "Connection information" section is expanded. The "Initial context factory" field has two radio buttons: "Use default:" (selected) and "Specify the fully qualified class name:" (unselected). The "Specify the fully qualified class name" field contains the text "com.sun.jndi ldap.LdapCtxFactory". The "Connection URL" field contains "ldap://example:389" and has an example below it: "Example: ldap://ldap-server-hostname:389". The "Connect user ID" field contains "cn=admin,dc=itso,dc=ibm,dc=com". The "Connect password" and "Confirm password" fields are masked with dots. The "Connect protocol" field is a dropdown menu set to "Default". The "Authentication level" field is a dropdown menu set to "Simple".

Figure 10-6 LDAP configuration example, window 1

Figure 10-7 shows another example of how we configured LDAP during our cluster installation.

User lookup	
* User base:	<input type="text" value="ou=cambridge_L,dc=itso,dc=ibm,dc=com"/> The base LDAP context (location) to search for users.
* User search matching:	<input type="text" value="(uid={0})"/> The LDAP attribute search string used to find the user. For example: (uid={0})
* User search sub-tree:	<input type="button" value="Yes"/> If set to Yes, sub-trees under the "User base" will be searched for users.
Role lookup	
* Role base:	<input type="text" value="ou=cambridge_L,dc=itso,dc=ibm,dc=com"/> The base LDAP context (location) to search for roles.
* Role name:	<input type="text" value="cn"/> The LDAP attribute type that corresponds to the role name. For example: cn
* Role search sub-tree:	<input type="button" value="Yes"/> If set to Yes, sub-trees under the "Role base" will be searched for roles.
* User role entry attribute:	<input checked="" type="radio"/> Use user entry attribute for roles <input type="radio"/> Use role entry attribute for users
* User role search string:	<input type="text" value="(memberUID={0})"/> Example role attribute: (memberUID={0}) Example user attribute: (memberOf={0})

Figure 10-7 LDAP configuration example, window 2

After the installation process, you must create a configuration file with a name such as `User-group.properties`. The `ldap-user_group.properties` are specified as shown in Example 10-6, where the group and user are the ones they have on LDAP/ADS.

Example 10-6 `$BIGINSIGHTS_HOME/console/conf/security/ldap-user_group.properties`

```
group1=user1,user2
group2=user3,user4
group3=user1,user5
```

When you create the file, you can run the `$BIGINSIGHTS_HOME/bin/createosusers.sh` script to create the OS users. The resulting output on our system looked like the example that is shown in Example 10-7.

Example 10-7 `createosusers.sh` example output

```
[biadmin@bddn26 biginsights]$ bin/createosusers.sh
console/conf/security/ldap-user_group.properties
[INFO] Cluster - Setup users/groups for hadoop proxy users
[INFO] @example - root@example's password:
Added new group BigInsightsSystemAdministrator
Added new user user1
[INFO] Progress - 33%
[INFO] @example - root@example's password:
Added new group BigInsightsSystemAdministrator
Added new user user1
[INFO] Progress - 67%
[INFO] @example - root@example's password:
Added new group BigInsightsSystemAdministrator
Added new user user1
```

[INFO] Progress - 100%
[INFO] DeployManager - ; SUCCEEDED components: [CreateOSUsers]; FAILED components: []

10.3.3 Pluggable Authentication Module

Pluggable Authentication Module (PAM) authentication provides a layer that exists between applications and authentication. It is an API that takes care of authenticating a user to a service. The principal feature of PAM is the dynamic configuration of authentication through either */etc/pam.d* or */etc/pam.conf* file.

To install PAM, you select the PAM authentication method during the installation process and describe the user and group mapping to roles. An example is described in section 10.2.1, “Roles” on page 147. The window that is shown for the PAM settings during the installation is shown in Figure 10-8.

Specify the groups and users (separated by comma) defined in PAM that will have access to the secured BigInsights Console. For example

BigInsights System Administrator - Users with the BigInsights System Administrator role perform all system administration tasks including n

* Groups:

Users:

BigInsights Data Administrator - Users with the BigInsights Data Administrator role perform all data administration tasks including file struct

* Groups:

Users:

BigInsights Application Administrator - Users with the BigInsights Application Administrator role perform all job management tasks includin
to enable efficiency in performing above functions.

* Groups:

Users:

BigInsights User - The BigInsights User role is a non-administrative role. Users with this role can execute jobs, view results, view data and

* Groups:

Figure 10-8 PAM authentication installation example window

If you choose PAM authentication during your BigInsights installation, PAM is automatically configured to authenticate with a local, server, shadow password file (*flat-file*), as shown in Example 10-8.

Example 10-8 Default content of /etc/pam.d/net-sf-jpam

auth	required	/lib64/security/pam_unix_auth.so
account	required	/lib64/security/pam_unix_acct.so
password	required	/lib64/security/pam_unix_passwd.so
session	required	/lib64/security/pam_unix_session.so

You can configure PAM to authenticate with LDAP by modifying `/etc/pam.d/net-sf-jpam`, as shown in Example 10-9.

Example 10-9 Content of `/etc/pam.d/net-sf-jpam` to configure PAM with LDAP

```
auth      required /lib64/security/pam_ldap.so
account   required /lib64/security/pam_ldap.so
password  required /lib64/security/pam_ldap.so
session   required /lib64/security/pam_ldap.so
```

After you replace the `/etc/pam.d/net-sf-jpam` content, you must restart the InfoSphere BigInsights cluster for the changes to take effect.

10.4 Secure browser support

The InfoSphere BigInsights installer provides the option to configure HTTPS to potentially provide more security when a user connects to the BigInsights web console. If HTTPS is selected, the Secure Sockets Layer and Transport Layer Security (SSL/TLS) protocol provides security for all communication between the browser and the web server. The InfoSphere BigInsights installer generates a self-signed certificate with a validity period that is suitable for easily deploying and testing HTTPS. An example of the HTTPS installation settings is shown in Figure 10-9.

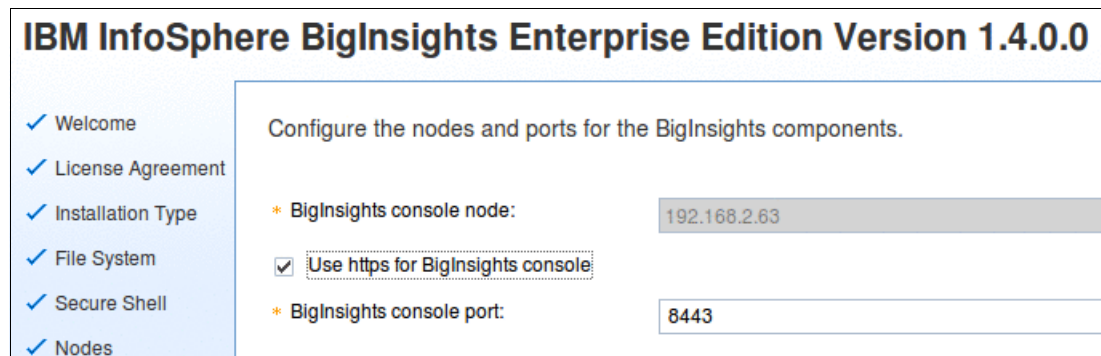


Figure 10-9 Configure HTTPS during the installation process

The console port (8080 by default) on the system that hosts the web console (typically the management node) must be the only open HTTP port for non-SSL traffic. To allow users to connect by using HTTPS, the port you specify during the installation (set to port 8443 by default) must also be an open port within your firewall between your users and the server where you are running the BigInsights web console service.

To provide secure browser access, enable an authentication method during the installation process (either PAM, Flat file or LDAP). Through the authentication method you select at installation time, users must identify themselves using a login page, which is shown in Figure 10-10 on page 156, before being allowed to access the BigInsights web console content. After successful authentication, the content that is displayed in the web console differs from user to user, which is based on their specific group assignments.

Figure 10-10 shows a BigInsights login window.

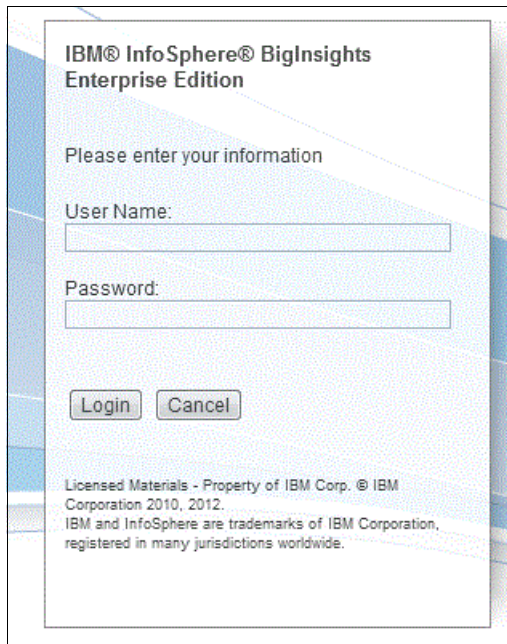


Figure 10-10 *BigInsights login window*

The BigInsightsSystemAdministrator can manage the cluster from the Cluster Status tab. With this tab, you can add and remove nodes, stop jobs that are running, or start services within the cluster.

The BigInsightsSystemAdministrator role also can download the Eclipse plug-in and the example applications that are provided with the software to work within the Eclipse environment.

In the Applications tab, users in the BigInsightsSystemAdministrator and BigInsightsApplicationAdministrator groups have access to several applications provided by BigInsights. Whereas, a user in the BigInsightsUser group sees only applications and data that they were given permission to access.

A comparison of the different appearance of the Welcome tab within the BigInsights console can be seen in Figure 10-11 and Figure 10-12.

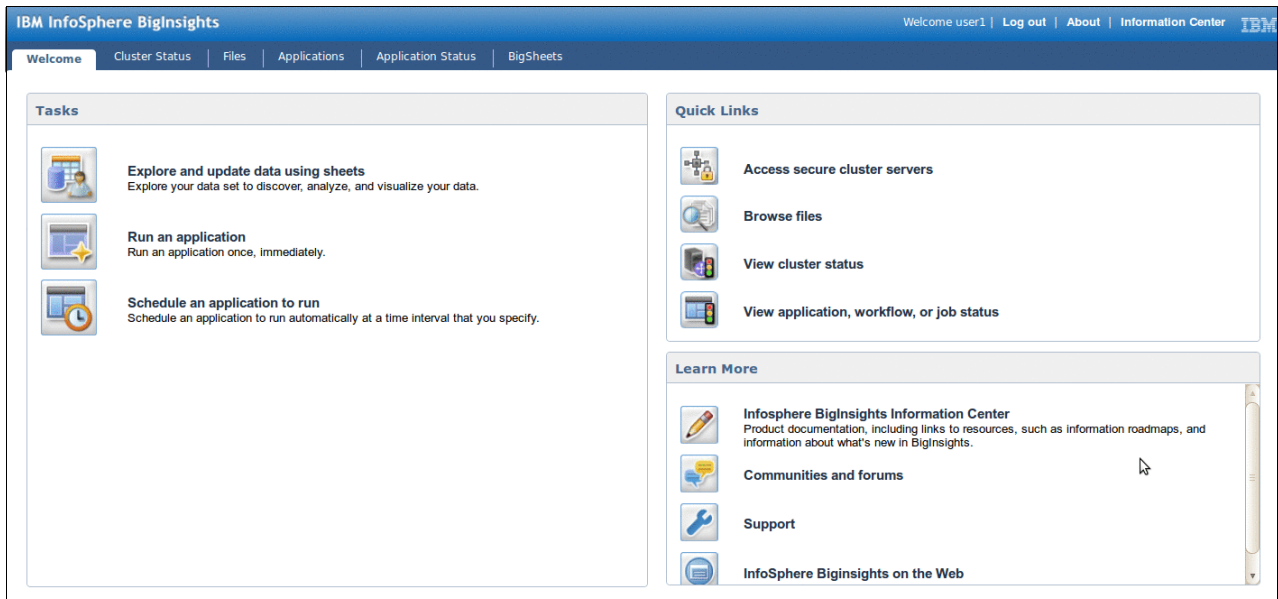


Figure 10-11 The default set of icons for users in the BigInsightsUser group

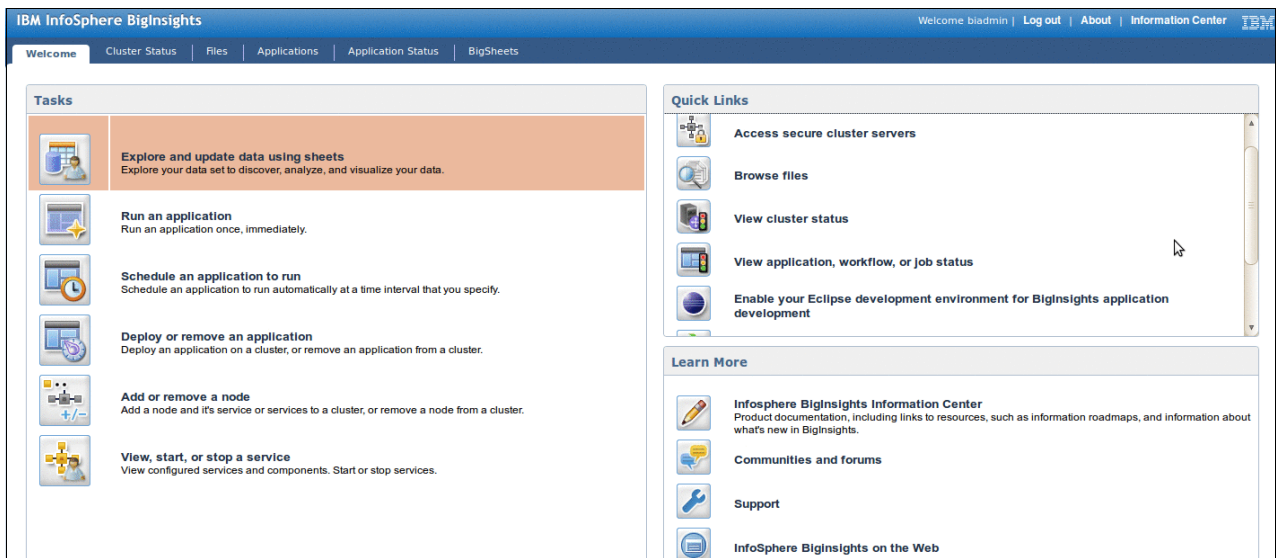


Figure 10-12 The default set of icons for users in the BigInsightsSystemAdministrator group

Hopefully, this chapter provided a helpful summary of the different security options that are provided by BigInsights.

BigInsights security: For more information about InfoSphere BigInsights security, see this website:

<http://pic.dhe.ibm.com/infocenter/bigins/v1r4/index.jsp?topic=%2Fcom.ibm.swg.im.infosphere.biginsights.admin.doc%2Fdoc%2F0057888.html&resultof%3D%2522%2573%2565%2563%2575%2572%2569%2574%2579%2522%2520%2522%2573%2565%2563%2575%2572%2522%2520>



IBM Platform Symphony

This chapter introduces IBM Platform Symphony, which brings low-latency scheduling and multi-tenancy to IBM InfoSphere BigInsights. The changing nature of distributed computing is explored, followed by explanations of how Platform Symphony operates, how InfoSphere BigInsights works with Platform Symphony, and the performance benefit of Platform Symphony. Sections highlighting the applications supported and the BigInsights versions supported are also provided.

11.1 Overview

IBM InfoSphere BigInsights is built on Apache Hadoop, an open source software framework that supports data-intensive, distributed applications. By leveraging open source Hadoop, and extending it with advanced analytic tools and other value-added capabilities, BigInsights helps organizations of all sizes more efficiently manage the vast amounts of data that consumers and businesses create every day.

At its core, Hadoop is a distributed computing environment that manages the execution of distributed jobs and tasks on a cluster. As with any distributed computing environment, the Hadoop software needs to provide facilities for resource management, scheduling, remote execution, and exception handling. Although Hadoop provides basic capabilities in these areas, IBM Platform Computing has been working on and improving facilities for these areas for twenty years.

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

IBM Platform Symphony is a low-latency scheduling solution that supports true multi-tenancy and sophisticated workload management capabilities. In the sections that follow, we provide an overview of IBM Platform Symphony and its architecture. Further, we explain why IBM Platform Symphony is uniquely suited to scheduling and managing MapReduce and other grid workloads. We also describe specifically how Platform Symphony complements IBM InfoSphere BigInsights, helping to deliver better performance at a lower cost for a variety of big data workloads.

11.2 The changing nature of distributed computing

Workload managers have 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 workflows
- ▶ Parametric sweeps and session-oriented workloads
- ▶ Parallel job scheduling with back-fill scheduling optimizations
- ▶ Various types of preemptive scheduling policies ensuring service levels are respected
- ▶ Service-oriented workloads for various low-latency, scatter-gather scheduling problems

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 usage, workload managers need to be optimized to the specific workload pattern 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 transparent to higher level application frameworks that rely on distributed computing services such as IBM InfoSphere BigInsights.

11.3 About IBM Platform Symphony

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 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 like pricing, risk management, fraud-detection and credit risk modeling, not only is raw performance 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 purpose built to deliver extremely high levels of performance and agility at scale to serve the unique challenges of this market where seconds count.

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

- ▶ Ability to obtain 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

As of Version 5.2, Platform Symphony Advanced Edition has included a scheduling framework optimized for Hadoop-compatible MapReduce workloads. This framework is compatible with IBM InfoSphere BigInsights, and customers can optionally use Platform Symphony as a scheduler rather than the native scheduler included in BigInsights.

11.4 IBM Platform Symphony architecture

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

In Platform Symphony, client applications connect through a client-side application programming interface (API). Among the software environments supported by Platform Symphony are Java, C++, C#/.NET, Microsoft Excel (COM) and native application services. To use Symphony, clients do not necessarily need to adapt their programs to the grid using the Symphony APIs, but most do in practice to benefit from the performance provided by native integrations. An alternative, namely a scriptable interface called “symexec” allows clients to wrap existing applications as callable services without the need for recompiling or linking applications.

Platform Symphony supports heterogeneous computing environments consisting of Linux, UNIX, and Microsoft operating environments. It provides developers with the freedom to choose their preferred development environment. It also supports integrations with a variety of distributed data management architectures including distributed databases, distributed file systems, and distributed data caches.

Platform Symphony Service Instance Managers (SIMs) are started on compute hosts in response to demand for scaled-out application services. Tasks received from clients are distributed to compute hosts optimally by one or more Platform Symphony Session Managers (SSM). The Symphony Session Manager fulfills the role of a broker, and after tasks are handed from a client to a session manager, task execution is guaranteed by the middleware. This saves developers significant headaches in writing reliable, distributed applications because the middleware handles runtime exceptions automatically.

The Platform Symphony Developer Edition is a freely downloadable toolkit and grid simulator for developers that helps them to grid-enable services without actually needing access to a production-scale grid. This makes it easier and more cost effective to validate the correct functioning of frameworks requiring grid services like BigInsights.

Many of the tedious aspects of building, deploying and managing distributed applications including error handling, load balancing, and maximizing the performance of multi-CPU and multi-core systems are handled automatically by Platform Symphony. Platform Symphony ensures that services are transparently recoverable in the event of software or host failures. With Platform Symphony handling all these complexities, developers become more productive, and application services are made inherently more reliable.

11.5 Platform Symphony MapReduce framework

Platform Symphony Advanced Edition includes a Hadoop compatible Java MapReduce API optimized for low-latency MapReduce workloads. Higher level Hadoop applications such as Pig, Hive, Jaql and other BigInsights components run directly on the Symphony MapReduce framework.

Hadoop components such as the Hadoop job tracker and task tracker in Symphony have been reimplemented as Platform Symphony applications, taking advantage of Symphony's fast middleware, resource sharing and fine-grained scheduling capabilities.

Figure 11-1 on page 163 depicts the implementation of MapReduce in the Platform Symphony environment. Using a MapReduce application adapter, clients can submit native MapReduce workloads directly to IBM Platform Symphony. Applications continue to use standard Hadoop MapReduce and common library APIs. The Java MapReduce API is built on top of Symphony's native APIs so that applications can benefit from Symphony features in a way that is transparent to the application. The actual Symphony Session Manager (SSM) and Service Instance Managers (SIMs) are written in C++ to maximize performance.

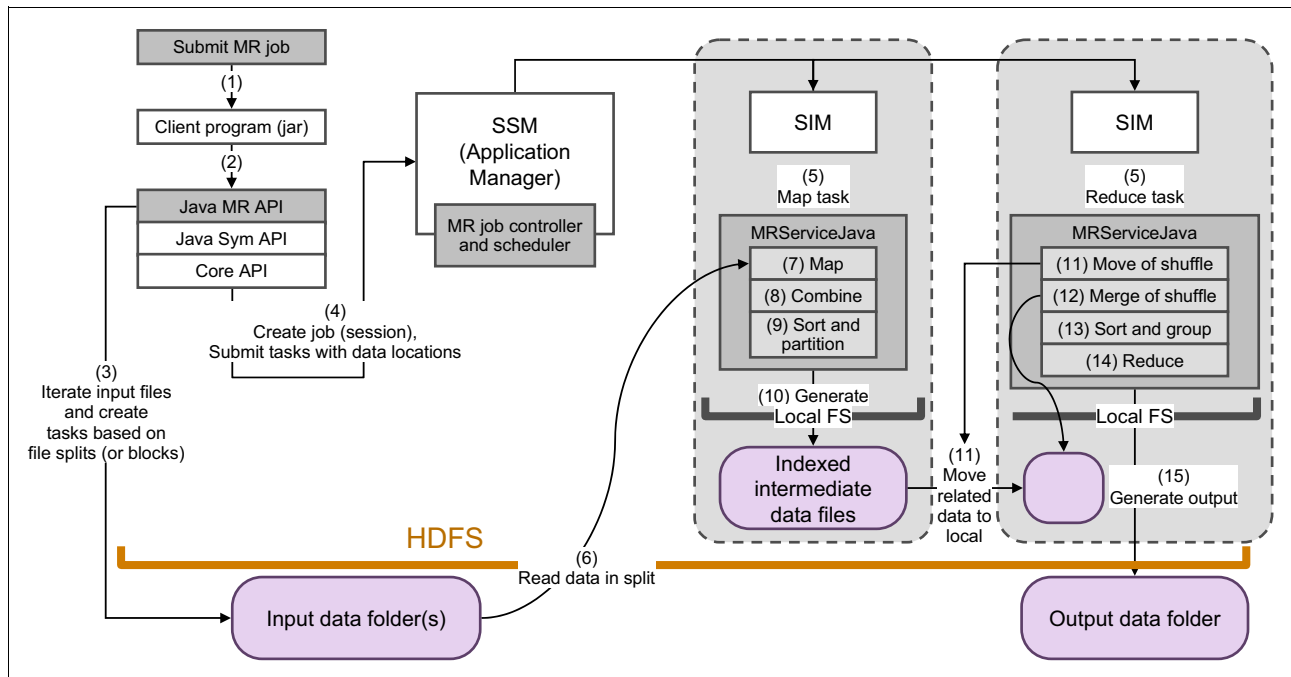


Figure 11-1 MapReduce implementation in Platform Symphony

In the Symphony MapReduce framework, Hadoop jobs map to Platform Sessions, shown at (4) in Figure 11-1. Whereas Hadoop normally allows only a single job tracker to run on a Hadoop cluster, in the Symphony architecture, the Symphony Session Manager implements the functions of the Job Tracker in Hadoop.

The following list includes benefits of this approach:

- ▶ Up to 300 applications (JobTrackers) can run concurrently on the same grid (4).
- ▶ Each SSM schedules and manages task execution independently for greater scale.
- ▶ The scheduling of Map, Reduce, and Shuffle tasks benefit from the Symphony low-latency scheduling framework for better performance (5).
- ▶ The Symphony native data transfer mechanisms and other shuffle stage optimizations improve performance during the critical shuffle phase (11).
- ▶ MapReduce workloads inherit the ability to dynamically borrow and loan resources at run time for fast, fine-grained dynamic resource sharing.
- ▶ The Symphony generalized file input and output architecture shown at (6) and (15) in the figure provide better control over input data sources and output sources where choices include local file systems, shared file systems (NFS), distributed file systems (HDFS or IBM GPFS™) or databases.
- ▶ Whereas Hadoop allocates slots specifically to either map or reduce tasks, Symphony has the notion of generic slots that can handle either map or reduce workloads. This helps efficiency near the beginning and end of MapReduce workloads by ensuring that slot usage is maximized.
- ▶ Multiple versions of the Hadoop MapReduce run time can coexist on the same cluster at the same time. This means that applications using different Hadoop versions of the MapReduce API can be deployed in the same multi-tenant environment helping reducing cost, providing greater flexibility, and simplifying application lifecycle management.

With IBM Platform Symphony, thousands of Intel- or Power-based compute servers can be shared by one or more MapReduce clients and other applications in the same cluster. The results are:

- ▶ Higher workload throughput
- ▶ Better application agility
- ▶ Improved manageability
- ▶ Improved reliability and capacity to ensure service level agreements (SLAs) are met
- ▶ Improved resource sharing
- ▶ Higher resource utilization and reduced cost

11.6 Multi-tenancy built in

The notion of multi-tenancy and resource sharing is a core capability of Platform Symphony. To support multi-tenancy, the Symphony resource sharing model implements the notion of “consumers”. Consumers are simply entities that require resources on a grid. These might be lines of business, departments, applications, or users. Consumers can be expressed in hierarchies or trees, reflecting multilevel organizational structures in flexible ways.

Individual consumers are configurable to reflect the application services to be run, the types of resources to run on, and ownership and sharing policies around grid resources. For example, one consumer on the grid might be running MapReduce services for a particular BigInsights instance and “own” 1,000 cores on a larger shared grid. This consumer prefers to schedule tasks to nodes on which HDFS or GPFS is installed, but can flex at run time to schedule tasks to other nodes as well. Another consumer on the grid might be running a non-MapReduce workload and own another 1,000 cores.

Individual consumers can be configured to loan or borrow resources with other consumers at run time based on sharing policies and priorities that change at run time. Configurable preemption policies ensure that ownership is preserved, but also ensure that idle capacity on the grid does not go to waste and that any consumer can flex dynamically to tap idle compute cycles on a shared grid.

Figure 11-2 on page 165 provides an illustration of how resource sharing plans are defined. For a particular resource group (in this case a group called “Compute Hosts”), an administrator can specify what consumers (applications) own what share of the available cores on a shared grid. Consumers that own some share of the resource group are guaranteed access when their workloads need to run. Owners of resources can determine whether they want to share resources with other groups when they are not in use, and whether they want to tap unused cores elsewhere on the grid that are eligible for sharing.

Consumers that do not own a specific share of resources will run tasks opportunistically as resources become available according to configurable sharing ratios. These resource-sharing plans can also be configured to change sharing policies automatically based on the time of day for even greater flexibility around resource sharing.

For example, during business hours, priority may be given to time-critical interactive workloads that require a timely response from Hadoop or other grid applications, so these applications will be provided with a minimum service level guarantee. During the overnight hours, service level guarantees may be extended to longer-running batch applications to guarantee that service level objectives are met.

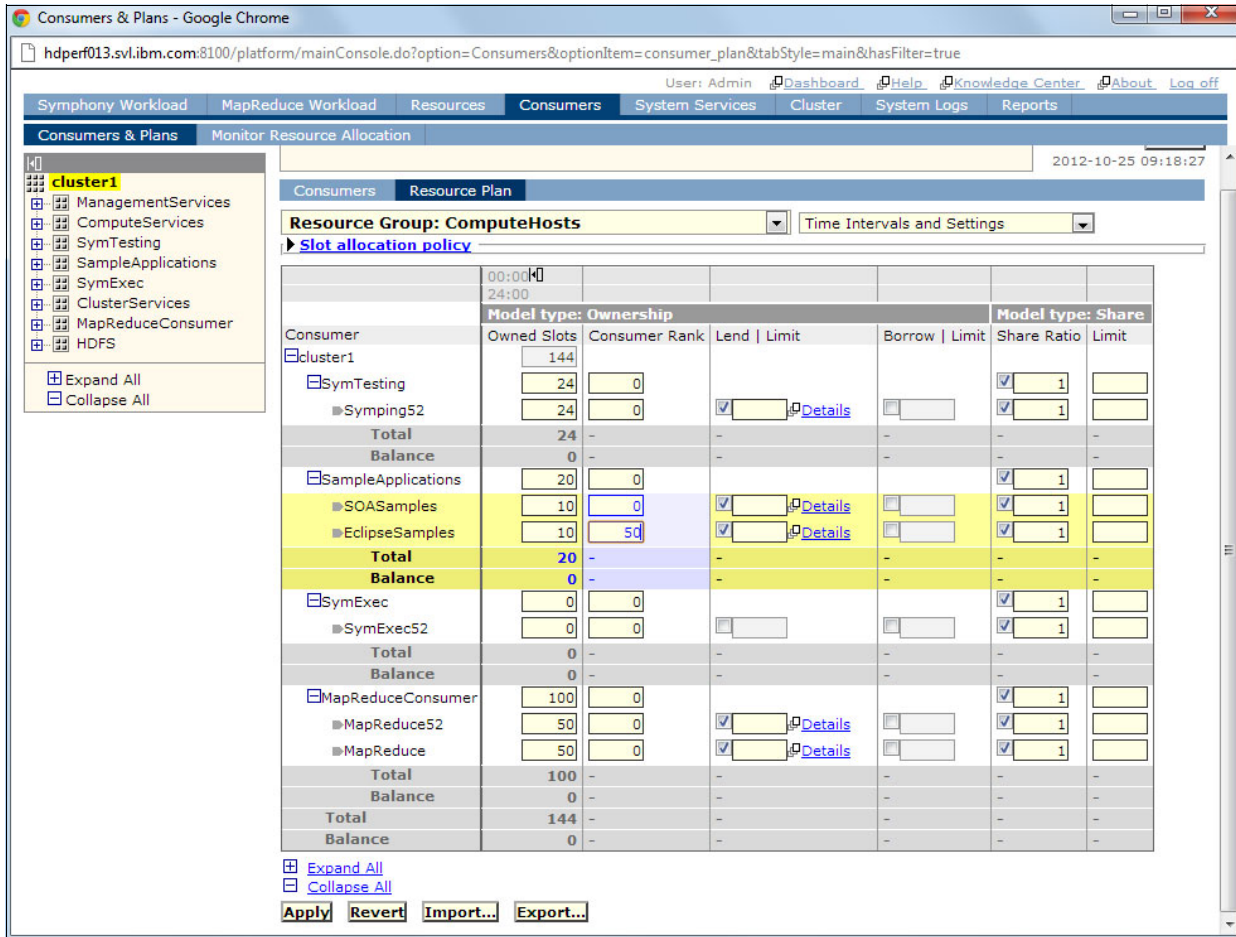


Figure 11-2 Configuring resource sharing plans and consumer trees in Platform Symphony

A powerful feature of Symphony is the dynamic nature of how resources are shared. Multiple Hadoop jobs may be launched and associated with the same consumer. Within that single consumer, jobs can share resources based on individual job priorities.

As an example, consider a MapReduce consumer allocated 1,000 cores on a shared grid. The consumer might start a long-running MapReduce job (Job A) with priority 200 that will initially consume all 1,000 cores allocated to that consumer. If other consumers had idle capacity, the job might flex to dispatch tasks to an even larger number of cores.

Assume that the consumer suddenly had a high priority workload (Job B) that needed to get completed quickly. In this case they can run another workload associated within the same consumer at a higher priority of 800. With a proportional allocation policy, Symphony will dynamically change the number of slots available to each application. Job A will be throttled back to schedule tasks to only 200 cores; Job B (given its high proportional allocation) will have access to 800 cores, thus enabling the higher priority job to finish more quickly while Job A progresses at a slower rate. Depending on how preemption policies are configured in Symphony, this rebalancing can take place almost instantly, making Symphony an exceptionally agile grid platform for time-critical applications.

To help instrument these sharing policies, and understand at run time how resources are being used, Platform Symphony provides resource allocation views shown in Figure 11-3 on page 166 that show real-time views of how cores are being allocated between consumers.

In this example we see that a MapReduce 5.2 consumer has 119 cores allocated to it. The shaded line beyond the current allocation illustrates that there are actually a great many map or reduce tasks pending (approximately 10,000 in this example) showing the grid administrator that performance can be gained by providing this consumer with additional resources, possibly by adjusting resource sharing policies.

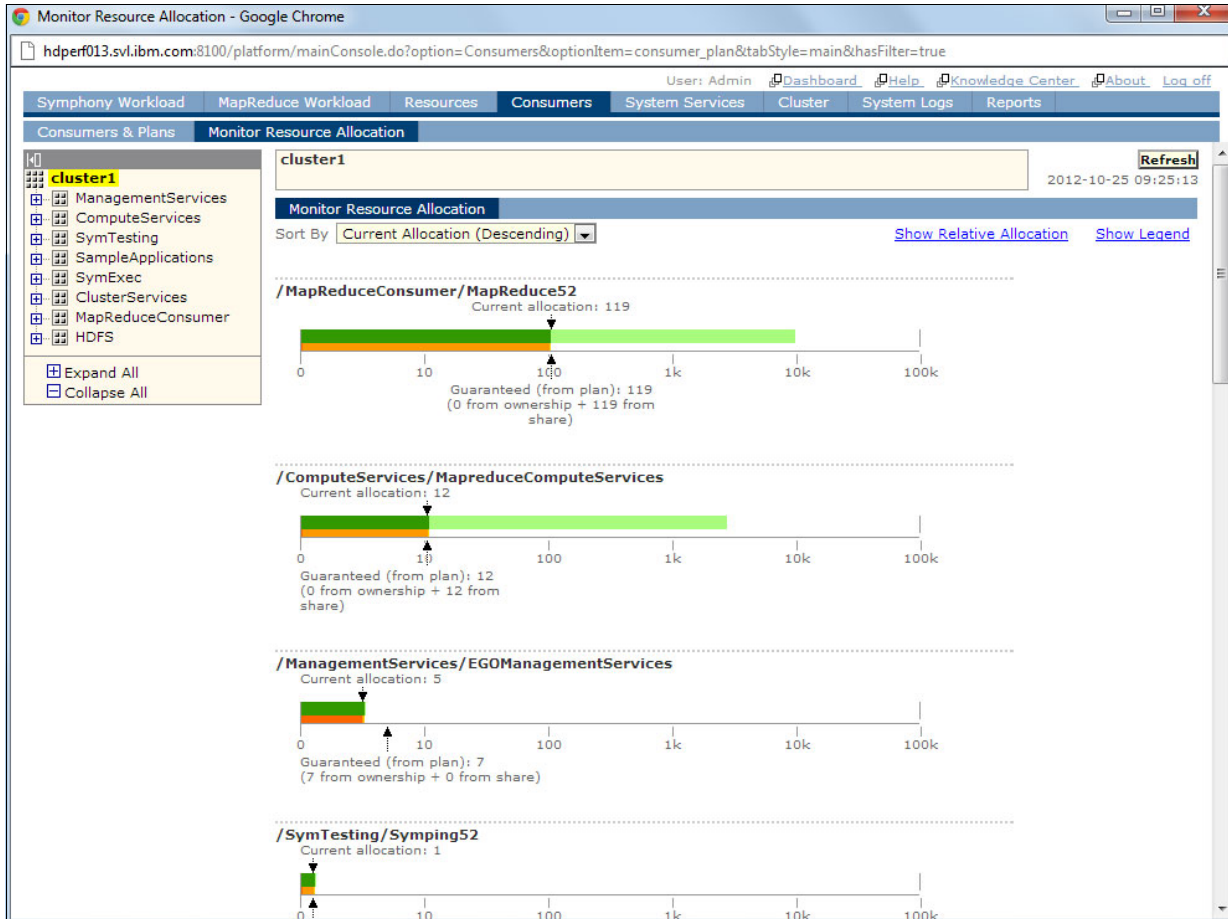


Figure 11-3 Monitor resource allocations in Platform Symphony to maximize efficiency

11.7 How InfoSphere BigInsights works with Platform Symphony

Figure 11-4 on page 167 shows the various components that make up the BigInsights Enterprise Edition. Many technology components in BigInsights have been developed by IBM, but others, including the MapReduce framework, are open source components supported by IBM.

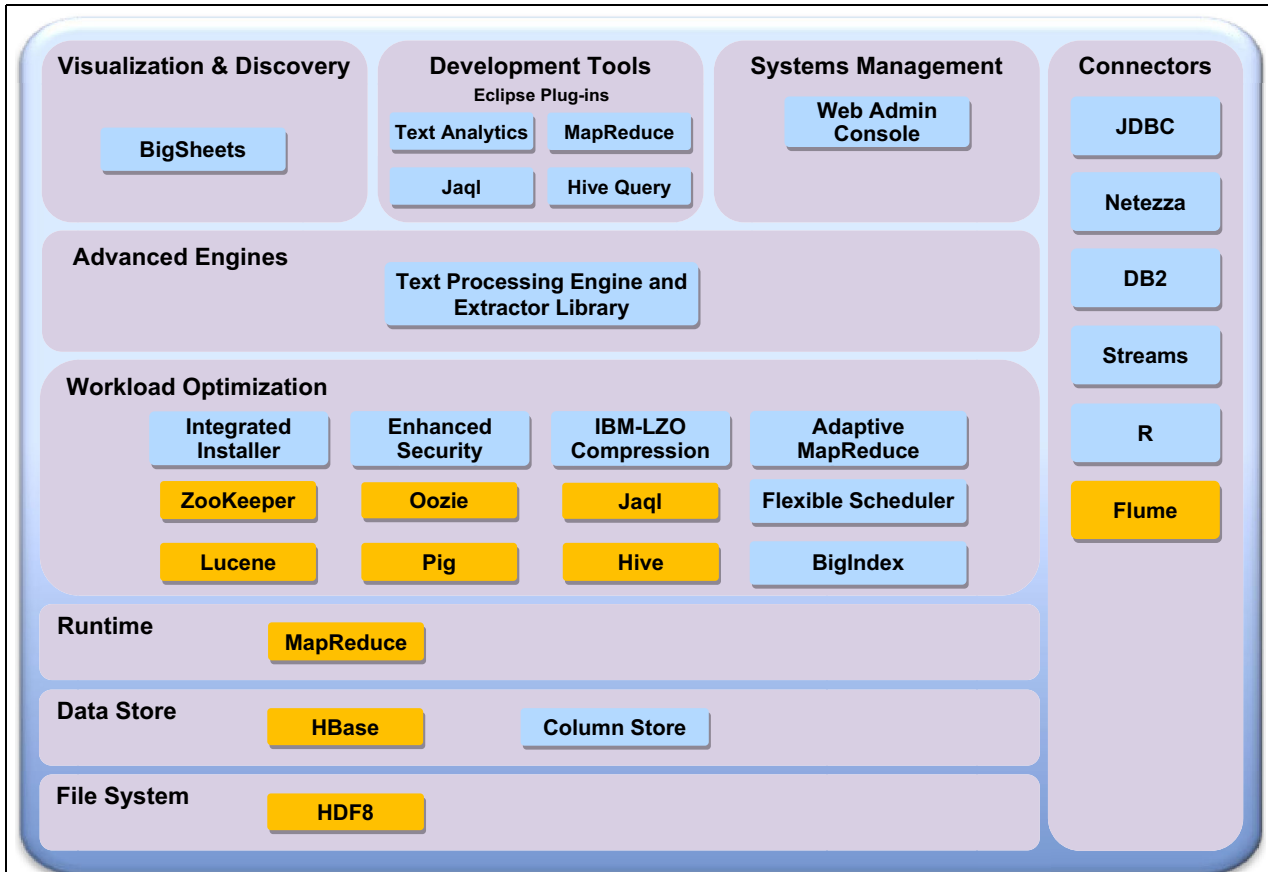


Figure 11-4 IBM InfoSphere BigInsights Components

This view makes it clear how BigInsights augments open-source Apache Hadoop while providing additional capabilities including visualization and query tools, development tools, management tools, and data connectors to external data stores.

When Platform Symphony is deployed with IBM InfoSphere BigInsights, Symphony essentially replaces the open source MapReduce layer in the Hadoop framework. It is important to note that Platform Symphony itself is not a Hadoop distribution. Platform Symphony relies on a Hadoop MapReduce implementation being present, along with various open source components such as Pig, Hive, HBase, and HDFS file systems.

As shown in Figure 11-5 on page 168, Platform Symphony replaces the MapReduce scheduling layer in the BigInsights software environment to provide better performance and multi-tenancy in a way that is transparent to BigInsights and BigInsights users.

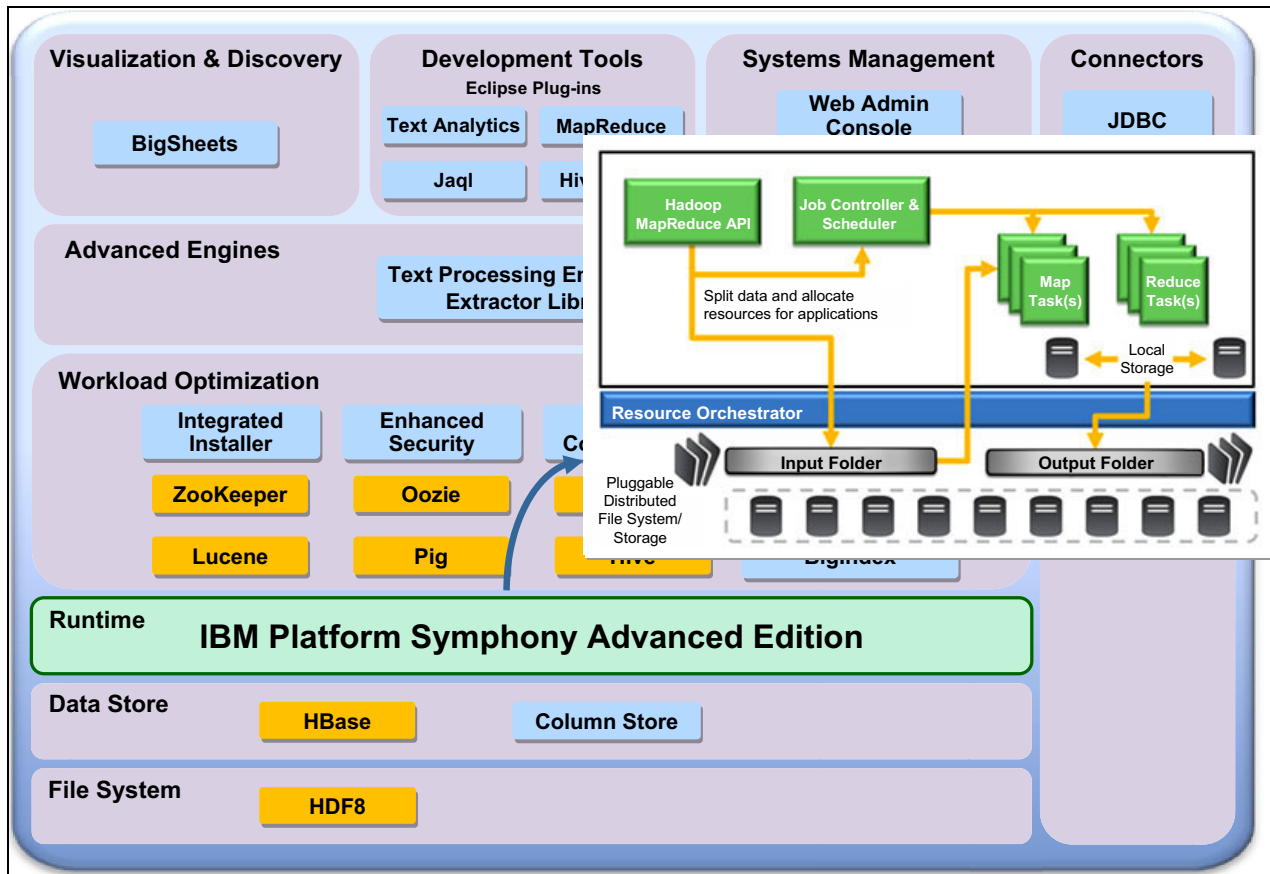


Figure 11-5 IBM Platform Symphony replaces Hadoop MapReduce run time

Big data workloads may be submitted from the BigInsights graphical interface, from the command line, or from client applications that interact with the Hadoop MapReduce APIs. After a few documented integration steps are performed to configure BigInsights to use the Symphony scheduler in place of the Hadoop scheduler, BigInsights workloads will run seamlessly and will be manageable from within the BigInsights environment.

Administrators need to be aware that when running BigInsights on a shared Platform Symphony grid, some cluster and service management capabilities accessible from within BigInsights become redundant. For example, it is no longer assumed that BigInsights has exclusive use of the grid, so capabilities such as cluster node management, service management, and high availability features for components like the NameNode, JobTrackers and TaskTrackers are all provided natively by Platform Symphony.

An example of the seamless nature of the integration is shown in Figure 11-6 on page 169. In this example, a user is interacting with BigSheets, the spreadsheet-oriented interface included in BigInsights to search through terabytes of data to determine the top ten products sold in a given month. This illustrates the power of BigInsights. Rather than needing to write custom Pig scripts or MapReduce applications, the BigSheets interface automatically translates the request into a MapReduce job that runs on the underlying Platform Symphony cluster. Because the job is running on Symphony, users benefit from faster run-times. Also, Symphony's improved performance means that less hardware infrastructure is often needed to meet performance objectives, thereby helping to reduce infrastructure costs.



Figure 11-6 BigInsights users need not be aware they are running on a Platform Symphony grid

Workloads can be monitored and managed from with the Platform Symphony management console as shown in Figure 11-7 on page 170. From the Symphony perspective we see that there are multiple MapReduce workloads running concurrently on the shared grid, sharing resources based on sharing policies and workload priorities.

Notice in Figure 11-7 that there are multiple long-running analytic jobs on the grid running at priority 5,000, but jobs submitted by BigSheets users are deployed to the grid at a high priority (10,000) to ensure that interactive jobs like BigSheets complete more quickly in the shared environment. Unlike standard Hadoop that has limited prioritization features, Platform Symphony has 10,000 priority levels and multiple configurable options related to resource sharing. This kind of sophisticated resource sharing, giving priorities to interactive workloads, is not possible in a Hadoop MapReduce environment without Platform Symphony.

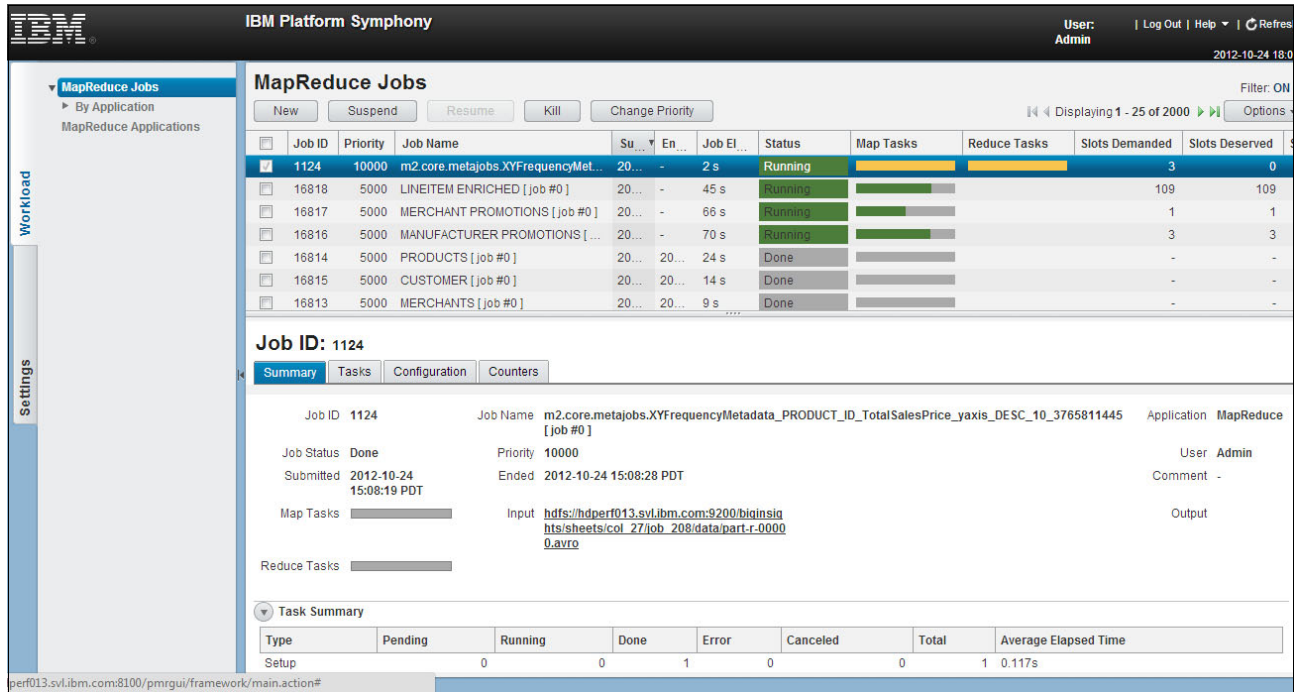


Figure 11-7 Multiple concurrent MapReduce workloads where interactive workloads have priority

11.8 Understanding the Platform Symphony performance benefit

IBM continues to achieve record MapReduce performance results with IBM Platform Symphony, as documented in a report of testing recently conducted and audited by an independent third-party testing lab¹.

Performance improvements for MapReduce workloads running on Symphony are a result of the following factors:

- ▶ Low-latency scheduling means that jobs start (and complete) faster.
- ▶ By design, Symphony monitors hosts dynamically and always schedules tasks preferentially to the host best able to respond quickly to workload.
- ▶ By avoiding Hadoop's heartbeat (polling) model, the task scheduling rate for Hadoop workloads is improved dramatically with Symphony, and scheduling latency is reduced.
- ▶ If resources are idle on the cluster, Symphony MapReduce workloads can expand resource allocations dynamically to borrow unused nodes to maximize utilization.
- ▶ Symphony uses generic slots rather than slots statically allocated to map and reduce functions, thus enabling slots to be shared between map and reduce tasks.
- ▶ The MapReduce shuffle-phase is improved and attempts to keep data in memory while using Symphony's more efficient data transfer mechanisms for moving data between hosts.

¹ Access to the report is available at the following URL:
<http://www-03.ibm.com/systems/technicalcomputing/platformcomputing/products/symphony/highperfhadoop.html>

- ▶ Developers can optionally use and benefit from API feature and data handling enhancements unique to Symphony to achieve additional advantages beyond what can be achieved within MapReduce itself.

The performance benefit is due in part due to Symphony’s faster scheduling framework. A standard way of measuring scheduling efficiency in Hadoop is a test called the “sleep test,” where a large number of short-running tasks are scheduled to the cluster to stress the scheduler². This increase in the rate at which tasks can be scheduled translates into performance gains in real MapReduce applications. Platform Symphony will be especially beneficial to applications with large numbers of relatively short-running tasks.

11.9 Supported applications

Platform Symphony supports a variety of application integrations for both non-MapReduce and MapReduce applications. The following capabilities within InfoSphere BigInsights have been tested and are known to work correctly with IBM Platform Symphony:

- ▶ Oozie workflows
- ▶ HIVE
- ▶ Pig
- ▶ BigSheets
- ▶ JAQL
- ▶ HBASE

11.10 BigInsights versions supported

IBM Platform Symphony supports BigInsights on Version 5.2 and later of IBM Platform Symphony. Table 11-1 shows version requirements to run BigInsights on Symphony.

Table 11-1 InfoSphere BigInsights and Platform Symphony compatibility

IBM Platform Symphony version	IBM InfoSphere BigInsights version	Patch required
IBM Platform Symphony 5.2 Advanced Edition	IBM InfoSphere BigInsights Enterprise Edition 1.3.0.1 IBM InfoSphere BigInsights Enterprise Edition 1.4 IBM InfoSphere BigInsights Enterprise Edition 2.0	Yes – BI Integration Patch is required
IBM Platform Symphony 6.1 Advanced Edition	IBM InfoSphere BigInsights Enterprise Edition 2.0	Yes – BI Integration Patch is required

IBM Platform Symphony has been tested with the following Hadoop-compatible distributions:

- ▶ Apache Hadoop 1.0.0, 1.0.1, 0.21.0, 0.20.2, 0.20.203, and 0.20.204
- ▶ Cloudera CDH3 update 1 and update 2
- ▶ IBM InfoSphere BigInsights 1.3.0.1, 1.4, and 2.0

² The use of the sleep test for this purpose is described in a Hadoop World presentation shared by Todd Lipcon and Yanpei Chen of Cloudera in 2010.

IBM Platform Symphony has also been tested with open source and third party commercial applications and compatible file systems including the following:

- ▶ IBM GPFS 3.4 or later
- ▶ Appistry CloudIQ Storage
- ▶ Apache Pig, Mahout, Nutch, Hbase, Oozie, Zookeeper, Hive
- ▶ Hadoop Pipes (an adapter that allows MapReduce application logic to be written in other languages)
- ▶ Datameer Analytics Solutions 1.3.7 (running on Cloudera CDH 3 update 2)

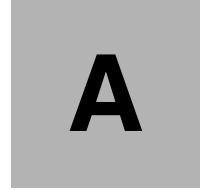
11.11 Summary

Clients deploying IBM InfoSphere BigInsights or other Big Data application environments can realize significant benefits by using IBM Platform Symphony as a grid manager. These benefits include:

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

Symphony will be especially beneficial to BigInsights clients running heterogeneous workloads that benefit from low latency scheduling. But the resource sharing and cost savings opportunities provided by Platform Symphony extend to all types of workloads.

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



M4 reference architecture

This appendix contains a bill of materials for the M4 reference architecture.

The M4 series of servers: Bill of materials

The bill of materials for the M4 server series is listed in the following sections.

IBM x3630 M4: The data node

The M4 revision is still a 2U high-density storage server but it has some serious upgrades over the M3. The first change of note is the upgrade from 2 x 3.06 GHz hexa-core processor to 2 x Sandy Bridge-EN 2.1 GHz octa-core processor. The performance is still fit for purpose at 2.1 GHz but with the extra four cores per server, the system is able to work much faster. The DIMM slots have been changed to 12 slots (UDIMM/RDIMM) that support 2DPC@1600 Mhz allowing up to 384 GB RAM and also provide better performance than the M3 model. The server starts with software (SW) Redundant Array of Independent Disks (RAID) or slot-less hardware (HW) RAID which allows an upgrade to advanced RAID without using a valuable Peripheral Component Interconnect (PCI) slot. The storage rich model also has 14 x 3.5-inch HS HDDs providing ultra dense storage. Even with the storage rich model, the server still has two remaining PCI slots. The Ethernet controller had an upgrade from the Intel 82575 to the 13504. To provide the power to drive this performance, the supply has increased to redundant 750 W cables, providing an efficient power supply unit (PSU).

Specification list with options

Processor

- ▶ Performance option 1: 2 E5-2450 2.1 GHz 8-Core, 95 W (transition from x3630M3 X5675 6-Core)
- ▶ Performance option 2: 2 E5-2430 2.2 GHz 6-Core, 95 W (transition from x3630M3 E5645 6-Core)
- ▶ Value data node: 2 E5-2420 1.9 GHz 6-Core, 95 W (transition from x3630M3 E5620 4-Core)

Memory

- ▶ Performance data node: 96 GB memory, 12 x 8 GB 1600 MHz RDIMMs
- ▶ Value data node: 48 GB memory, 6 x 8 GB 1333 MHz RDIMMs

Storage controller

- ▶ Slotless 6 Gbps Performance Optimized host bus adapter (HBA)

Hard disk drives (HDDs)

- ▶ 14 x 3.5" 2 TB or 3 TB NL serial-attached SCSI (SAS) drives
- ▶ 14 x 3.5" 2 TB or 3 TB NL Serial Advanced Technology Attachment (SATA) drives

Network

- ▶ 1 Gb network:
 - Option 1: Use 2 x 1 Gb server integrated ports from base offering, LACP 2 x 1 Gb ports to TOR G8052 for higher network throughput and HA, NIC1 is shared with IMM. If adapter HA is required, add one more 1 Gb dual ports Ethernet adapter
 - Option 2: use 4 x 1 Gb ports with FoD, LACP 4x 1 Gb to 2 G8052 for higher network throughput and HA, NIC1 is shared with IMM. This configuration requires two G8052 per rack

- ▶ 10 Gb network:
 - Use 1 Mellanox ConnectX-3 EN dual port SFP+ 10 GbE adapter for Performance data node, or 2 Mellanox adapters if redundancy is required. LACP or vLAG all 10 Gb ports to 1 or 2 G8264s. Use standard 1 Gb base offering for IMM network
 - Assess 10 GBase-T after Refresh 2

750 W Power supply

- ▶ 1 for Value or Performance data node
- ▶ 2 for Enterprise data node

IBM System x3550 M4: The management node

The IBM System x3550 is still a tiny 1U dual socket rack server but the specs have grown. The *M4* has a Sandy Bridge-EP octo-core at 2.9 GHz. The server has up to 24 slots for UDIMM or RDIMM which means it can hold a massive 768 GB of RAM (32 GB LR DIMM). The server still makes room for 8 x 2.5-inch or 3 x 3.5-inch HDDs and slotless RAID. The NIC is set up with 4 x 1 GB as standard plus 2 x 10 Gb (slotless opt).

The following are the full specifications of the IBM System x3550 M4:

CPU/Chipset:

Dual CPU/Socket R

Processors: Intel E5-2600 (2, 4, 6, 8 core)

Chipset: Intel Patsburg-B

2 QPI (Quick Path Interconnect), up to 8.0 GT/s

TPM1.2 Rev 1.03 Chip down

HDD and ODD:

8 SAS/SATA 2.5-inch HS HDD/SSDs, optional ODD. Or

3 SATA 3.5-inch HS/SS HDDs

HDD Controllers:

Slotless RAID

NIC:

4 x 1 Gb: Intel I350AM4

Optional 2x 10 GbE Mezz card

I/O:

USB: 2 front (3 for 3.5-inch)/4 back /2 internal

VGA: 1 front/1 back

2 PCIe 3.0 slots, also support PCI-X

Memory:

Mixing different types of memory (UDIMMs, RDIMMs, HyperCloud DIMMs, and LRDIMMs) is not supported.

768 GB* max (with 32 GB LRDIMM)

Memory mirroring and sparing is supported.

Power:

550 W/750 W AC with 80PLUS Platinum certification
750 W DC PSU*

BMC:

IMM2
Renesas 7757

Power Management:

AEM 4.x application
xSmart Energy Control (HW and FW)

Form Factor:

Power Capping
System Power Maximizer
Slotless RAID
H1110 RAID 0/1/10 at the lowest cost
M1115 RAID 0/1/10, FoD upgrades to RAID 5/50 (no battery / no cache)
M5110 RAID 0/1/10/5/50 with
1 GB Flash back, 512 MB Flash Back, or cache and battery
FoD upgrades to 6/60

New Options:

New CPU options
New Memory
New Riser
New 10 GbE LAN Mezz card (QLogic, Emulex, and IB*)
New HDD
2.5-inch HDD upgrade kit

OEM:

Enabled for OEM at GA

System Management/Software:

IMM2
UpdateXpress Service Package
Tivoli Provisioning Manager for Images System x Edition 8.0 (Replaces Tivoli Provisioning Manager for OS Deployment)
Director 6.3 (or latest versions)
AEM4.x application
Toolscenter support (ASU, BoMC, UXSPI, DSA, Server Guide, and Scripting Toolkit)

Firmware:

BIOS: IBM UEFI
BMC: IBM IMM2 with FoD
DIAGs: (DSA/Preboot-DSA in UEFI Environment)
LSI
RTMM

Green Server:

RoHS (Pb-Free)
Energy Efficient Design
Energy Star Compliance*
80Plus Platinum certification

Warranty:

3-year (4-year for Intel model)

Recommended features for BigInsights x3550 M4 management node

Standardize processor, RAID adapter, HDDs, power supplies

Processor

2 E5-2650 2.0 GHz 8-Core, 95 W (+26% perf, -23% cost relative to x5675)

Memory

128 GB memory, 16 x 8 GB 1600 MHz CL11 RDIMMs (for predefined configuration)

128 GB memory, 16 x 8 GB 1333 MHz CL9 RDIMMs (for cheaper memory price)

RAID adapter

Slotless ServeRAID M1115, no cache

HDDs

4 x 2.5" 600 GB SAS HS drives

Optional DVD

Two 550 W Power supplies

Network

1 Gb network with redundant network adapters

Use integrated 1 Gb ports and one more PCIe 1 Gb adapter. Improve throughput and HA by link aggregating all 1 Gb ports to G8052 or vLAG 1 Gb ports to different G8052s

IMM can use either dedicated management port or shared over integrated 1 Gb port

10 Gb network with redundant network adapters

Use 2 Mellanox ConnectX-3 EN Dual port SFP+ 10 GbE adapters to improve HA and throughput by link aggregating all four 10 Gb ports to a single G8264 or vLAG 10 Gb ports to different G8264s

IMM uses dedicated management port

Assess 10 GBase-T after refresh 2



B

Installation values

Certain components within the cluster provide installation values and items that you can potentially configure. A portion of this appendix provides a quick reference listing of the IBM BigInsights installation, *default values*. We also provide the version numbers for each of the open source, software components that are included as part of the BigInsights product installation.

As a way to make monitoring easier, we also list all of the options you can select to monitor with Ganglia.

BigInsights default installation values

This section provides a summary of default installation values used by BigInsights. Values that are listed as *configurable* signify that you are allowed by BigInsights to specify a *user-defined* value.

Table 11-2 BigInsights installation items and values

Installation item/option	Value
InfoSphere BigInsights administrator group ID	biadmin (configurable)
InfoSphere BigInsights administrator user ID	biadmin (configurable)
InfoSphere BigInsights installation directory	/opt/ibm/biginsights (configurable)
InfoSphere BigInsights data/log directory	/var/ibm/biginsights (configurable)
BigInsights web console security	<ul style="list-style-type: none"> ▶ Use https for BigInsights web console false (default)/ true ▶ The https setting is only for InfoSphere BigInsights Enterprise Edition
BigInsights web console port (HTTP)	8080
BigInsights web console port (HTTPS)	8443
Configure Jaql UDF server	Yes/No
Jaql UDF server port	8200 (configurable)
Derby port	(configurable)
Cache directories	/hadoop/mapred/local (configurable)
MapReduce system directory	/hadoop/mapred/system (configurable)
Shared POSIX file system root directory	Not applicable (configurable)
NameNode port	9000 (configurable)
NameNode HTTP port	50070 (configurable)
JobTracker port	9001 (configurable)
JobTracker HTTP port	50030 (configurable)
Secondary NameNode HTTP port	50090 (configurable)
Secondary NameNode data directories	/hadoop/hdfs/namesecondary (configurable)
Data node IPC port	50020 (configurable)
Data node HTTP port	50075 (configurable)
TaskTracker HTTP port	50060 (configurable)
Data directory	/hadoop/hdfs/data (configurable)
Configure Hive	Yes/No
Hive Web Interface port	9999 (configurable)
Hive node port	10000 (configurable)
Configure Pig	Yes/No
ZooKeeper port	2181 (configurable)

Installation item/option	Value
Tick time (in milliseconds)	2,000 (configurable)
Initial time (in ticks)	5 (configurable)
Sync interval (in ticks)	2 (configurable)
Flume ZooKeeper mode	Use a shared/separate ZooKeeper installation
Oozie port	8280 (configurable)
Configure HBase	Yes/No
HBase ZooKeeper mode	Use a shared/separate ZooKeeper installation
HBase root directory	/hbase (configurable)
HBase master port	60000 (configurable)
HBase master UI port	60010 (configurable)
HBase region server port	60020 (configurable)
HBase region server UI port	60030 (configurable)

Open source technologies and version numbers

In addition to Apache Hadoop, IBM BigInsights also provides several more integrated, open source components. Some of these items are not required to be installed. However, the installation of certain components is mandatory for the proper operation of your BigInsights cluster. In the following list, you can see the version and a short description of each component that is included as part of BigInsights V1.4:

- ▶ IBM distribution of Apache Hadoop (Version 1.0.0). An open source distributed platform, which includes the IBM SDK for Java 6.
- ▶ Avro (Version 1.5.1). A data serialization system to store persistent data and integrate it with dynamic languages.
- ▶ Derby (Version 10.5.3.1). A relational database that is implemented in Java.
- ▶ Flume (Version 0.9.4). A distributed service for moving large amounts of log data around a cluster.
- ▶ HBase (Version 0.90.5). A non-relational database to provide real-time read/write, random access to your data.
- ▶ Hive (Version 0.8.0). A data warehouse system.
- ▶ IBM Jaql. A language whose objectives are to research semi-structured query processing, extensibility, and parallelization. Jaql uses *JSON (JavaScript Object Notation)* as a simple, yet flexible way to represent data that ranges from flat, relational data to semi-structured, XML data.
- ▶ Lucene (Version 3.3.0). A high-performance, text-based search engine.
- ▶ Oozie (Version 2.3.1). A workflow management system to coordinate Apache Hadoop jobs.
- ▶ Pig (Version 0.9.1). A component that interacts with Hadoop for analyzing large data sets. Pig provides a high-level language for data analysis.
- ▶ ZooKeeper (version 3.3.4). A centralized framework to enable distributed coordination.

Ganglia monitoring options

Example B-1 displays a list of the DFS, JVM, MapReduce, Hadoop cluster, RPC, and User group metrics that you can configure for monitoring within Ganglia at the time of writing of this book.

Example B-1 A list of Ganglia monitoring metrics

DFS Metrics

```
dfs.datanode.blockChecksumOp_avg_time.rrd
dfs.datanode.blockChecksumOp_num_ops.rrd
dfs.datanode.blockReports_avg_time.rrd
dfs.datanode.blockReports_num_ops.rrd
dfs.datanode.blocks_get_local_pathinfo.rrd
dfs.datanode.blocks_read.rrd
dfs.datanode.blocks_removed.rrd
dfs.datanode.blocks_replicated.rrd
dfs.datanode.blocks_verified.rrd
dfs.datanode.blocks_written.rrd
dfs.datanode.block_verification_failures.rrd
dfs.datanode.bytes_read.rrd
dfs.datanode.bytes_written.rrd
dfs.datanode.copyBlockOp_avg_time.rrd
dfs.datanode.copyBlockOp_num_ops.rrd
dfs.datanode.heartBeats_avg_time.rrd
dfs.datanode.heartBeats_num_ops.rrd
dfs.datanode.readBlockOp_avg_time.rrd
dfs.datanode.readBlockOp_num_ops.rrd
dfs.datanode.reads_from_local_client.rrd
dfs.datanode.reads_from_remote_client.rrd
dfs.datanode.replaceBlockOp_avg_time.rrd
dfs.datanode.replaceBlockOp_num_ops.rrd
dfs.datanode.writeBlockOp_avg_time.rrd
dfs.datanode.writeBlockOp_num_ops.rrd
dfs.datanode.writes_from_local_client.rrd
dfs.datanode.writes_from_remote_client.rrd
dfs.FSNamesystem.BlockCapacity.rrd
dfs.FSNamesystem.BlocksTotal.rrd
dfs.FSNamesystem.CapacityRemainingGB.rrd
dfs.FSNamesystem.CapacityTotalGB.rrd
dfs.FSNamesystem.CapacityUsedGB.rrd
dfs.FSNamesystem.CorruptBlocks.rrd
dfs.FSNamesystem.ExcessBlocks.rrd
dfs.FSNamesystem.FilesTotal.rrd
dfs.FSNamesystem.MissingBlocks.rrd
dfs.FSNamesystem.PendingDeletionBlocks.rrd
dfs.FSNamesystem.PendingReplicationBlocks.rrd
dfs.FSNamesystem.ScheduledReplicationBlocks.rrd
dfs.FSNamesystem.TotalLoad.rrd
dfs.FSNamesystem.UnderReplicatedBlocks.rrd
dfs.namenode.AddBlockOps.rrd
dfs.namenode.blockReport_avg_time.rrd
dfs.namenode.blockReport_num_ops.rrd
dfs.namenode.CreateFileOps.rrd
dfs.namenode.DeleteFileOps.rrd
```


dfs.namenode.FileInfoOps.rrd
dfs.namenode.FilesAppended.rrd
dfs.namenode.FilesCreated.rrd
dfs.namenode.FilesDeleted.rrd
dfs.namenode.FilesInGetListingOps.rrd
dfs.namenode.FilesRenamed.rrd
dfs.namenode.fsImageLoadTime.rrd
dfs.namenode.GetBlockLocations.rrd
dfs.namenode.GetListingOps.rrd
dfs.namenode.JournalTransactionsBatchedInSync.rrd
dfs.namenode.SafemodeTime.rrd
dfs.namenode.Syncs_avg_time.rrd
dfs.namenode.Syncs_num_ops.rrd
dfs.namenode.Transactions_avg_time.rrd
dfs.namenode.Transactions_num_ops.rrd

JVM Metrics

jvm.metrics.gcCount.rrd
jvm.metrics.gcTimeMillis.rrd
jvm.metrics.logError.rrd
jvm.metrics.logFatal.rrd
jvm.metrics.logInfo.rrd
jvm.metrics.logWarn.rrd
jvm.metrics.memHeapCommittedM.rrd
jvm.metrics.memHeapUsedM.rrd
jvm.metrics.memNonHeapCommittedM.rrd
jvm.metrics.memNonHeapUsedM.rrd
jvm.metrics.threadsBlocked.rrd
jvm.metrics.threadsNew.rrd
jvm.metrics.threadsRunnable.rrd
jvm.metrics.threadsTerminated.rrd
jvm.metrics.threadsTimedWaiting.rrd
jvm.metrics.threadsWaiting.rrd

MAPREDUCE Metrics

mapred.jobtracker.blacklisted_maps.rrd
mapred.jobtracker.blacklisted_reduces.rrd
mapred.jobtracker.heartbeats.rrd
mapred.jobtracker.jobs_completed.rrd
mapred.jobtracker.jobs_failed.rrd
mapred.jobtracker.jobs_killed.rrd
mapred.jobtracker.jobs_preparing.rrd
mapred.jobtracker.jobs_running.rrd
mapred.jobtracker.jobs_submitted.rrd
mapred.jobtracker.maps_completed.rrd
mapred.jobtracker.maps_failed.rrd
mapred.jobtracker.maps_killed.rrd
mapred.jobtracker.maps_launched.rrd
mapred.jobtracker.map_slots.rrd
mapred.jobtracker.occupied_map_slots.rrd
mapred.jobtracker.occupied_reduce_slots.rrd
mapred.jobtracker.reduces_completed.rrd
mapred.jobtracker.reduces_failed.rrd
mapred.jobtracker.reduces_killed.rrd
mapred.jobtracker.reduces_launched.rrd

mapred.jobtracker.reduce_slots.rrd
mapred.jobtracker.reserved_map_slots.rrd
mapred.jobtracker.reserved_reduce_slots.rrd
mapred.jobtracker.running_maps.rrd
mapred.jobtracker.running_reduces.rrd
mapred.jobtracker.trackers_blacklisted.rrd
mapred.jobtracker.trackers_decommissioned.rrd
mapred.jobtracker.trackers_graylisted.rrd
mapred.jobtracker.trackers.rrd
mapred.jobtracker.waiting_maps.rrd
mapred.jobtracker.waiting_reduces.rrd
mapred.Queue.jobs_completed.rrd
mapred.Queue.jobs_failed.rrd
mapred.Queue.jobs_killed.rrd
mapred.Queue.jobs_preparing.rrd
mapred.Queue.jobs_running.rrd
mapred.Queue.jobs_submitted.rrd
mapred.Queue.maps_completed.rrd
mapred.Queue.maps_failed.rrd
mapred.Queue.maps_killed.rrd
mapred.Queue.maps_launched.rrd
mapred.Queue.reduces_completed.rrd
mapred.Queue.reduces_failed.rrd
mapred.Queue.reduces_killed.rrd
mapred.Queue.reduces_launched.rrd
mapred.Queue.reserved_map_slots.rrd
mapred.Queue.reserved_reduce_slots.rrd
mapred.Queue.waiting_maps.rrd
mapred.Queue.waiting_reduces.rrd
mapred.shuffleOutput.shuffle_exceptions_caught.rrd
mapred.shuffleOutput.shuffle_failed_outputs.rrd
mapred.shuffleOutput.shuffle_handler_busy_percent.rrd
mapred.shuffleOutput.shuffle_output_bytes.rrd
mapred.shuffleOutput.shuffle_success_outputs.rrd
mapred.tasktracker.maps_running.rrd
mapred.tasktracker.mapTaskSlots.rrd
mapred.tasktracker.reduces_running.rrd
mapred.tasktracker.reduceTaskSlots.rrd
mapred.tasktracker.tasks_completed.rrd
mapred.tasktracker.tasks_failed_ping.rrd
mapred.tasktracker.tasks_failed_timeout.rrd

Hadoop cluster metrics

metricssystem.MetricsSystem.dropped_pub_all.rrd
metricssystem.MetricsSystem.num_sinks.rrd
metricssystem.MetricsSystem.num_sources.rrd
metricssystem.MetricsSystem.publish_avg_time.rrd
metricssystem.MetricsSystem.publish_imax_time.rrd
metricssystem.MetricsSystem.publish_imin_time.rrd
metricssystem.MetricsSystem.publish_max_time.rrd
metricssystem.MetricsSystem.publish_min_time.rrd
metricssystem.MetricsSystem.publish_num_ops.rrd
metricssystem.MetricsSystem.publish_stdev_time.rrd
metricssystem.MetricsSystem.sink.ganglia.dropped.rrd
metricssystem.MetricsSystem.sink.ganglia.latency_avg_time.rrd

metricssystem.MetricsSystem.sink.ganglia.latency_num_ops.rrd
metricssystem.MetricsSystem.sink.ganglia.qsize.rrd
metricssystem.MetricsSystem.snapshot_avg_time.rrd
metricssystem.MetricsSystem.snapshot_imax_time.rrd
metricssystem.MetricsSystem.snapshot_imin_time.rrd
metricssystem.MetricsSystem.snapshot_max_time.rrd
metricssystem.MetricsSystem.snapshot_min_time.rrd
metricssystem.MetricsSystem.snapshot_num_ops.rrd
metricssystem.MetricsSystem.snapshot_stdev_time.rrd

RPC Metrics

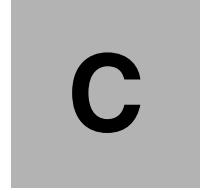
rpcdetailed.rpcdetailed.addBlock_avg_time.rrd
rpcdetailed.rpcdetailed.addBlock_num_ops.rrd
rpcdetailed.rpcdetailed.blockReceived_avg_time.rrd
rpcdetailed.rpcdetailed.blockReceived_num_ops.rrd
rpcdetailed.rpcdetailed.blockReport_avg_time.rrd
rpcdetailed.rpcdetailed.blockReport_num_ops.rrd
rpcdetailed.rpcdetailed.blocksBeingWrittenReport_avg_time.rrd
rpcdetailed.rpcdetailed.blocksBeingWrittenReport_num_ops.rrd
rpcdetailed.rpcdetailed.canCommit_avg_time.rrd
rpcdetailed.rpcdetailed.canCommit_num_ops.rrd
rpcdetailed.rpcdetailed.commitBlockSynchronization_avg_time.rrd
rpcdetailed.rpcdetailed.commitBlockSynchronization_num_ops.rrd
rpcdetailed.rpcdetailed.commitPending_avg_time.rrd
rpcdetailed.rpcdetailed.commitPending_num_ops.rrd
rpcdetailed.rpcdetailed.complete_avg_time.rrd
rpcdetailed.rpcdetailed.complete_num_ops.rrd
rpcdetailed.rpcdetailed.create_avg_time.rrd
rpcdetailed.rpcdetailed.create_num_ops.rrd
rpcdetailed.rpcdetailed.delete_avg_time.rrd
rpcdetailed.rpcdetailed.delete_num_ops.rrd
rpcdetailed.rpcdetailed.done_avg_time.rrd
rpcdetailed.rpcdetailed.done_num_ops.rrd
rpcdetailed.rpcdetailed.getBlockLocations_avg_time.rrd
rpcdetailed.rpcdetailed.getBlockLocations_num_ops.rrd
rpcdetailed.rpcdetailed.getBuildVersion_avg_time.rrd
rpcdetailed.rpcdetailed.getBuildVersion_num_ops.rrd
rpcdetailed.rpcdetailed.getClusterStatus_avg_time.rrd
rpcdetailed.rpcdetailed.getClusterStatus_num_ops.rrd
rpcdetailed.rpcdetailed.getDatanodeReport_avg_time.rrd
rpcdetailed.rpcdetailed.getDatanodeReport_num_ops.rrd
rpcdetailed.rpcdetailed.getEditLogSize_avg_time.rrd
rpcdetailed.rpcdetailed.getEditLogSize_num_ops.rrd
rpcdetailed.rpcdetailed.getFileInfo_avg_time.rrd
rpcdetailed.rpcdetailed.getFileInfo_num_ops.rrd
rpcdetailed.rpcdetailed.getJobCounters_avg_time.rrd
rpcdetailed.rpcdetailed.getJobCounters_num_ops.rrd
rpcdetailed.rpcdetailed.getJobProfile_avg_time.rrd
rpcdetailed.rpcdetailed.getJobProfile_num_ops.rrd
rpcdetailed.rpcdetailed.getJobStatus_avg_time.rrd
rpcdetailed.rpcdetailed.getJobStatus_num_ops.rrd
rpcdetailed.rpcdetailed.getListing_avg_time.rrd
rpcdetailed.rpcdetailed.getListing_num_ops.rrd
rpcdetailed.rpcdetailed.getNewJobId_avg_time.rrd
rpcdetailed.rpcdetailed.getNewJobId_num_ops.rrd

rpcdetailed.rpcdetailed.getProtocolVersion_avg_time.rrd
rpcdetailed.rpcdetailed.getProtocolVersion_num_ops.rrd
rpcdetailed.rpcdetailed.getQueueAdmins_avg_time.rrd
rpcdetailed.rpcdetailed.getQueueAdmins_num_ops.rrd
rpcdetailed.rpcdetailed.getStagingAreaDir_avg_time.rrd
rpcdetailed.rpcdetailed.getStagingAreaDir_num_ops.rrd
rpcdetailed.rpcdetailed.getStats_avg_time.rrd
rpcdetailed.rpcdetailed.getStats_num_ops.rrd
rpcdetailed.rpcdetailed.getSystemDir_avg_time.rrd
rpcdetailed.rpcdetailed.getSystemDir_num_ops.rrd
rpcdetailed.rpcdetailed.getTask_avg_time.rrd
rpcdetailed.rpcdetailed.getTaskCompletionEvents_avg_time.rrd
rpcdetailed.rpcdetailed.getTaskCompletionEvents_num_ops.rrd
rpcdetailed.rpcdetailed.getTask_num_ops.rrd
rpcdetailed.rpcdetailed.heartbeat_avg_time.rrd
rpcdetailed.rpcdetailed.heartbeat_num_ops.rrd
rpcdetailed.rpcdetailed.mkdir_avg_time.rrd
rpcdetailed.rpcdetailed.mkdir_num_ops.rrd
rpcdetailed.rpcdetailed.nextGenerationStamp_avg_time.rrd
rpcdetailed.rpcdetailed.nextGenerationStamp_num_ops.rrd
rpcdetailed.rpcdetailed.ping_avg_time.rrd
rpcdetailed.rpcdetailed.ping_num_ops.rrd
rpcdetailed.rpcdetailed.recoverLease_avg_time.rrd
rpcdetailed.rpcdetailed.recoverLease_num_ops.rrd
rpcdetailed.rpcdetailed.register_avg_time.rrd
rpcdetailed.rpcdetailed.register_num_ops.rrd
rpcdetailed.rpcdetailed.rename_avg_time.rrd
rpcdetailed.rpcdetailed.rename_num_ops.rrd
rpcdetailed.rpcdetailed.renewLease_avg_time.rrd
rpcdetailed.rpcdetailed.renewLease_num_ops.rrd
rpcdetailed.rpcdetailed.rollEditLog_avg_time.rrd
rpcdetailed.rpcdetailed.rollEditLog_num_ops.rrd
rpcdetailed.rpcdetailed.rollFsImage_avg_time.rrd
rpcdetailed.rpcdetailed.rollFsImage_num_ops.rrd
rpcdetailed.rpcdetailed.sendHeartbeat_avg_time.rrd
rpcdetailed.rpcdetailed.sendHeartbeat_num_ops.rrd
rpcdetailed.rpcdetailed.setPermission_avg_time.rrd
rpcdetailed.rpcdetailed.setPermission_num_ops.rrd
rpcdetailed.rpcdetailed.setReplication_avg_time.rrd
rpcdetailed.rpcdetailed.setReplication_num_ops.rrd
rpcdetailed.rpcdetailed.setSafeMode_avg_time.rrd
rpcdetailed.rpcdetailed.setSafeMode_num_ops.rrd
rpcdetailed.rpcdetailed.statusUpdate_avg_time.rrd
rpcdetailed.rpcdetailed.statusUpdate_num_ops.rrd
rpcdetailed.rpcdetailed.submitJob_avg_time.rrd
rpcdetailed.rpcdetailed.submitJob_num_ops.rrd
rpcdetailed.rpcdetailed.versionRequest_avg_time.rrd
rpcdetailed.rpcdetailed.versionRequest_num_ops.rrd
rpc.rpc.callQueueLen.rrd
rpc.rpc.NumOpenConnections.rrd
rpc.rpc.ReceivedBytes.rrd
rpc.rpc.rpcAuthenticationFailures.rrd
rpc.rpc.rpcAuthenticationSuccesses.rrd
rpc.rpc.rpcAuthorizationFailures.rrd
rpc.rpc.rpcAuthorizationSuccesses.rrd

rpc.rpc.RpcProcessingTime_avg_time.rrd
rpc.rpc.RpcProcessingTime_num_ops.rrd
rpc.rpc.RpcQueueTime_avg_time.rrd
rpc.rpc.RpcQueueTime_num_ops.rrd
rpc.rpc.SentBytes.rrd
swap_free.rrd

User group metrics

ugi.ugi.loginFailure_avg_time.rrd
ugi.ugi.loginFailure_num_ops.rrd
ugi.ugi.loginSuccess_avg_time.rrd
ugi.ugi.loginSuccess_num_ops.rrd



Checklist

This appendix provides a set of quick reference items that can be useful when you set up your cluster. We include a set of networking items, a set of operating system settings, and a list of BigInsights configurations that might adversely affect cluster performance. The goal of this appendix is to provide you with a quick checklist to help ensure that nothing was overlooked in your efforts to get the cluster operational. This appendix is formatted for easy printing so that you can physically check things off after you review them.

BIOS settings to check

In 4.6.7, “Basic input/output system tool” on page 62, we covered the settings that you have to make at the BIOS level. Refer to that section if you feel like something might be missing here. A summary of items to check is shown in Table C-1.

Table C-1 BIOS items checklist

Check when done	Setting	Value to check
<input type="checkbox"/>	BIOS Firmware Level	latest release
<input type="checkbox"/>	IMM IP address	value that is allocated for management IP
<input type="checkbox"/>	Review BIOS settings to ensure that they comply with IT policy	multiple setting
<input type="checkbox"/>	BIOS settings across nodes equivalent	differences across nodes

Networking settings to verify operating system

If you need more details about networking configuration considerations, see Chapter 3, “BigInsights network architecture” on page 29. The list in Table C-2 is a simple checklist of items that are often forgotten to check. If you discover other items that you think will be useful, consider contacting one of the authors of this book so that we might penitentially add your suggestion in a future version of the book.

Table C-2 Networking items checklist

Check when done	If you...	...then do this
<input type="checkbox"/>	use a single public IP address per node for with each node being part of the corporate network,	ensure that <code>/etc/sysconfig/network-scripts/if-eth0.cfg</code> is properly configured with IP/netmask and gateway information
<input type="checkbox"/>	use a single public IP address per node with each node being part of the corporate network,	verify that <code>/etc/resolv.conf</code> is updated with the correct dns and domain suffix
<input type="checkbox"/>	have separate IP addresses for administration and performance networks	verify that there are unique Ethernet definitions in <code>/etc/sysconfig/network-scripts</code>
<input type="checkbox"/>	have a 10 Gb Ethernet card added to your configuration and want to bond Ethernet ports	verify that there are unique Ethernet definitions in <code>/etc/sysconfig/network-scripts</code> verify that there are unique bonding definitions in <code>/etc/sysconfig/network-scripts</code> verify that <code>/etc/modprobe.d</code> has bonding aliases defined

Operating system settings to check

You are now at a point with your cluster configuration where you just want to do a few double checks before you turn the systems over to the users. Additional information that is related to selecting, sizing, and configuring your hardware can be found in Chapter 4, “BigInsights hardware architecture” on page 49. Table C-3 contains a checklist of items at the operating system level that you might want to double check.

Table C-3 Operating system item checklist

Check when done	Item	Action required
<input type="checkbox"/>	Kernel page swap setting in /proc/sys/vm/swappiness	verify that the setting is 5 or less
<input type="checkbox"/>	Disk partitioning on data nodes	verify unique file systems on each drive
<input type="checkbox"/>	File systems that are mounted with the <i>noatime</i> option	ensure that the <i>noatime</i> option is set
<input type="checkbox"/>	Ganglia operation	verify that Ganglia is installed gmond is running on every node in cluster gmetad and HTTPd is running on the Ganglia web interface node
<input type="checkbox"/>	verify all processors are enabled	cat /proc/cpuinfo
<input type="checkbox"/>	verify memory is available	cat /proc/meminfo

Non-package-related items

Table C-4 Operating system items checklist

Check when done	If you...	...then do this
<input type="checkbox"/>	have not setup a valid YUM repository,	<< See repos.tar file for the files to embed into /etc/yum.repos.d >>
<input type="checkbox"/>	are running RHEL 6.2,	check your system against the list of package items that are listed in , “Installed Red Hat package items” on page 194
<input type="checkbox"/>	have not setup your file systems on the nodes,	verify unique file systems on each drive consider using ext3 or ext4 files systems with the <i>-noatime</i> option
<input type="checkbox"/>	have not performed the correct steps to ensure the time-of-day settings are synchronized across all nodes in your cluster,	you need to set up ntp by running the following commands: # Install and configure ntp service ntpd stop ntpdate -u 0.rhel.pool.ntp.org chkconfig ntpd on service ntpd start

Check when done	If you...	...then do this
<input type="checkbox"/>	have not disabled selinux,	disable selinux
<input type="checkbox"/>	have not disabled ipv6,	disable ipv6
<input type="checkbox"/>	have not setup password-less Secure Shell (SSH) on all nodes,	set up password-less SSH using root on all nodes
<input type="checkbox"/>	have not set vm.swappiness,	set vm.swappiness - 5 in /etc/sysctl.conf
<input type="checkbox"/>	have not set ulimits to a higher level than the OS default,	set ulimits using the commands in Example C-1

As an expansion of the table in Table C-4 on page 191, the commands that are shown in Example C-1 are the commands that are needed to set the correct ulimits settings on your nodes.

Example C-1 Commands for setting the correct value for ulimits

```
echo "biadmin hard nfile 16384" >> /etc/security/limits.conf
echo "biadmin soft nfile 16384" >> /etc/security/limits.conf
echo "biadmin hard nproc 4096" >> /etc/security/limits.conf
echo "biadmin soft nproc 4096" >> /etc/security/limits.conf
```

BigInsights configuration changes to consider

BigInsights configures ports and connections between components for you during installation. However, if you want BigInsights to perform at its best, there are some settings that we found useful when working with our example cluster. You can review these settings in Table C-5 and test them within your environment. These settings are not required changes but merely suggestions. You might find that your cluster performs better (as we did) if you make the changes that are suggested in Table C-5.

Table C-5 BigInsights optional configuration changes

Check if done	If you...	...you might consider this
<input type="checkbox"/>	see that your BigInsights jobs appear to be CPU bound,	changing your MapReduce Java options by changing the <code>mapred.child.java.opts</code> setting in your BigInsights cluster within the <code>/opt/ibm/biginsights/hdm/hadoop-conf-staging/mapred-site.xml</code> file as shown in Example C-2 on page 193
<input type="checkbox"/>	see that your BigInsights jobs might run faster with a larger file buffer size,	changing your MapReduce input/output (I/O) settings in your BigInsights cluster within the <code>/opt/ibm/biginsights/hdm/hadoop-conf-staging/core-site.xml</code> file as shown in Example C-4 on page 193

Check if done	If you...	...you might consider this
<input type="checkbox"/>	have a fair number of jobs in your workload that require sorting,	changing your MapReduce I/O sort settings in your BigInsights cluster within the <code>/opt/ibm/biginsights/hdm/hadoop-conf-staging/mapred-site.xml</code> file as shown in Example C-3
<input type="checkbox"/>	Require compression	use <code>com.ibm.biginsights.compress.CmxCodec</code> for temporary space during sort/shuffle phase and for storage of files on HDFS

Example C-2 The `mapred.child.java.opts` setting used for testing in `mapred-site.xml`

```

...
<property>
  <!-- Max heap of child JVM spawned by tasktracker. Ideally as large as the
        task machine can afford. The default -Xmx200m is usually too small. -->
  <name>mapred.child.java.opts</name>
  <value>-Xgcpolicy:gencon -Xms1024m -Xmx1024m -Xjit:optLevel=hot
-Xjit:disableProfiling -Xgcthreads4 -XlockReservation</value>
</property>
...

```

Example C-3 The `io.sort.mb` setting used for testing in `mapred-site.xml`

```

...
<property>
  <!-- The percentage of io.sort.mb dedicated to tracking record boundaries.
        Let this value be r, io.sort.mb be x. The maximum number of records
        collected before the collection thread must block is equal to
        (r * x) / 4. Memory comes out of the task's JVM memory allocation.
        Overrides default 100M. -->
  <name>io.sort.mb</name>
  <value>650</value>
</property>
...

```

Example C-4 The `io.file.buffer.size` setting used for testing in `core-site.xml`

```

...
<property>
  <!-- A larger buffer for lesser disk I/O. Overrides default 4096. -->
  <name>io.file.buffer.size</name>
  <value>524288</value>
</property>
...

```

To update the configuration file and broadcast the change through the cluster, the BigInsights **cluster settings synchronization** command must be used. The synchronization script takes edits that are done in a settings staging directory on the management node, and updates the rest of cluster. This function is useful because it saves the user the task of managing Hadoop configuration files across the cluster. To start, modify the Hadoop configuration settings in `/opt/ibm/biginsights/hdm/hadoop-conf-staging`. When the configuration files are modified, the Hadoop cluster must be resynchronized and restarted to

enable the metrics to be sent to the gmond daemon. This process is done with the following set of commands:

```
# Note: BIGINSIGHTS_HOME=/opt/ibm/biginsights
$BIGINSIGHTS_HOME/bin/synconf.sh hadoop force
$BIGINSIGHTS_HOME/bin/stop.sh hadoop
$BIGINSIGHTS_HOME/bin/start.sh hadoop
```

Optional settings: As with any performance tuning exercise, always check to see how any optional setting might change the overall performance of the cluster.

Setting descriptions

Because different jobs created different demands on the cluster, it is suggested to apply these on a per job basis before you apply them to the cluster-wide settings. Here are some considerations to take into account when you think about the changes and how they might apply to your specific workload:

- ▶ **-Xgcpolicy:gencon** tells the Java virtual machine (JVM) to use the concurrent generational garbage collector (GC). If all the cores for the machine are already in use for computation, this GC setting might waste CPU time on locking and more GC usage.
- ▶ **-Xms1024m** sets the initial Java heap size to 1 GB. Programs that do not need a 1 GB heap grab 1 GB of virtual and physical memory anyway. At best, this overly large heap evicts pages from the operating system's buffer cache, potentially resulting in more I/O usage. At worst, the extra memory usage might cause the system to run out of physical memory and thrash. If a system in a Hadoop cluster starts thrashing, the data node running on that machine times out. When this timeout happens on three nodes in the cluster, the entire distributed file system becomes unusable. Therefore, depending on the available memory on your data nodes, you might have to adjust this setting appropriately and monitor system resource usage after a change is applied.
- ▶ **-Xmx1024m** sets the maximum Java heap size to 1 GB. See comments in the previous bullet point.
- ▶ **-Xjit:optLevel=hot** tells the JIT compiler to spend more time optimizing compiled bytecode than it does by default. Programs that do not benefit from more optimizations might become slower with this option.
- ▶ **-Xjit:disableProfiling** tells the JIT compiler not to use profiling to identify “hot spots” for more optimization. Programs that do benefit from more optimizations might become slower with this option.
- ▶ **-Xgcthreads4** tells the GC to use exactly four threads. Single-threaded, memory-intensive programs required five cores or more, using four of them for potentially wasted GC work to keep the fifth core busy. Multi-threaded, memory-intensive programs might run slower if they need more than four GC threads to fully use all cores.
- ▶ **-XlockReservation** makes it faster for a thread to reacquire a lock it just released, at the expense of making it slower for other threads to acquire the lock. Multi-threaded map tasks that read inputs off a queue incur more locking usage with this option. If **-XlockReservation** has a significant effect on performance, the application probably has a problem with excessive locking that must be addressed directly.

Installed Red Hat package items

For the list in Example C-5 on page 195, we used the following operating system packages on our *Red Hat Enterprise Linux (RHEL) 6* update 2 version of the OS. If you are running a different OS version or another type of OS, refer to the latest package requirements for

BigInsights v1.4 that are publicly accessible at this website:

<http://pic.dhe.ibm.com/infocenter/bigins/v1r4/index.jsp?topic=%2Fcom.ibm.swg.im.infosphere.biginsights.install.doc%2Fdoc%2Fc0057608.html>

The packages that we used for our RHEL 6.2 instance can be reviewed in Example C-5.

Example C-5 RHEL 6.2 package list

```
@base
@cifs-file-server
@compat-libraries
@core
@development
@directory-client
# @eclipse
@emacs
@fonts
@ftp-server
@general-desktop
@graphical-admin-tools
@hardware-monitoring
@internet-browser
@java-platform
@large-systems
@legacy-unix
@network-file-system-client
@network-tools
@nfs-file-server
@performance
@perl-runtime
@scientific
@storage-client-fcoe
@storage-server
@system-admin-tools
@system-management
@x11

acpid
alacarte
amtu
at-spi
audit
authconfig-gtk
autofs
#dvs added apr for ganlia
apr
boost-devel
byacc
ccid
compat-db
compat-gcc-34
compat-gcc-34-c++
compat-gcc-34-g77
compat-glibc
compat-libstdc++-296
compat-libstdc++-33
```

compat-openldap
conman
coolkey
cpuspeed
crash
cryptsetup-luks
cscope
ctags
cvs
cyrus-sasl-devel
db4-devel
dbus-devel
dejagnu
device-mapper-multipath
dhclient
diffstat
dmraid
dos2unix
dosfstools
doxygen
dump
eject
ElectricFence
elfutils
emacs
emacs-nox
enscript
eog
esc
evince
expat-devel
expect
file-roller
finger
firefox
firstboot
freetype-devel
ftp
gcc-gfortran
gcc-gnat
gcc-objc
gdm
gedit
glx-utils
gmp-devel
gnome-backgrounds
gnome-power-manager
gnome-screensaver
gnome-system-monitor
gnome-themes
gnome-user-docs
gnome-utils
gnome-vfs2-smb
gnuplot
gok

gpm
grub
hplip
imake
indent
iptraf
iptstate
irqbalance
jwhois
kexec-tools
krb5-auth-dialog
krb5-devel
krb5-workstation
ksh
lapack
lftp
libacl-devel
libaio
libattr-devel
libcap-devel
libdrm-devel
libjpeg-devel
libpng-devel
libselinux-devel
libSM-devel
libtiff-devel
libusb-devel
libXau-devel
libXext-devel
libXft-devel
libxml2-devel
libXmu-devel
libXrandr-devel
libXrender-devel
libXtst-devel
logwatch
lsscsi
ltrace
man-pages
mcelog
mdadm
mentest86+
mesa-libGL-devel
mesa-libGLU-devel
mgetty
microcode_ctl
mkbootdisk
mlocate
mtools
mtr
nano
nasm
nautilus-open-terminal
nc
nfs-utils

nmap
notification-daemon
nsppluginwrapper
nss_db
ntp
numactl
OpenIPMI
openldap-clients
openldap-devel
openmotif22
openmotif-devel
openssh-askpass
openssh-clients
openssh-server
openssl-devel
oprofile
oprofile-gui
orca
pam_krb5
pam_passwdqc
pam_pkcs11
patchutils
pax
pcmciautils
perl-Crypt-SSLeay
perl-LDAP
perl-Mozilla-LDAP
perl-XML-Dumper
perl-XML-Grove
perl-XML-Twig
pexpect
pinfo
pm-utils
policycoreutils-gui
python-devel
python-docs
rcs
rdate
rdesktop
rdist
readahead
redhat-lsb
rhn-setup-gnome
rp-pppoe
rsh
rsh-server
rsync
rusers
rwho
sabayon
sabayon-apply
samba-client
screen
SDL-devel
sendmail

setroubleshoot
setuptools
smartmontools
sos
stunnel
subversion
sudo
swig
symlinks
sysstat
system-config-kdump
system-config-keyboard
system-config-lvm
system-config-network-tui
system-config-printer
system-config-services
system-config-users
systemtap
talk
tcl-devel
tcpdump
tcp_wrappers
tclsh
telnet
telnet-server
texinfo
tftp-server
time
tog-pegasus
tree
udftools
units
unix2dos
unzip
usbutils
valgrind
vconfig
vim-enhanced
vim-X11
vino
wget
which
wireless-tools
words
x86info
xdelta
xinetd
xorg-x11-twm
xorg-x11-utils
xrestop
xterm
ypbind
yum
yum-rhn-plugin
zip

```
zsh
-NetworkManager
-iscsi-initiator-utils
-fcoe-utils
-iptables-ipv6
```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy. There are three InfoSphere Streams Redbooks, one for each code version:

- ▶ *IBM InfoSphere Streams Harnessing Data in Motion*, SG24-7865
- ▶ *IBM InfoSphere Streams: Assembling Continuous Insight in the Information Revolution*, SG24-7970
- ▶ *Addressing Data Volume, Velocity, and Variety with IBM InfoSphere Streams V3.0*, SG24-8108

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

This publication is also relevant as a further information source:

- ▶ *IBM InfoSphere BigInsights Enterprise Edition Starter Kit*, LCD7-4860

Online resources

These websites are also relevant as further information sources:

- ▶ Bringing big data to the enterprise
<http://www-01.ibm.com/software/data/bigdata/>
- ▶ developerWorks big data site
<http://www.ibm.com/developerworks/data/products/bigdata/index.html/>
- ▶ InfoSphere BigInsights
<http://www-01.ibm.com/software/data/infosphere/biginsights/>
- ▶ developerWorks BigInsights site
<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/BigInsights/page/Welcme?lang=en/>
- ▶ InfoSphere Streams
<http://www-01.ibm.com/software/data/infosphere/streams/>

- ▶ MapReduce performance results with IBM Platform Symphony test report
<http://www-03.ibm.com/systems/technicalcomputing/platformcomputing/products/symphony/highperfhadoop.html>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



Implementing IBM InfoSphere BigInsights on IBM System X

(0.2"spine)
0.17"->0.473"
90->249 pages



Implementing IBM InfoSphere BigInsights on IBM System x



Introducing big data and IBM InfoSphere BigInsights

Installing an InfoSphere BigInsights environment

Monitoring and securing InfoSphere BigInsights

As world activities become more integrated, the rate of data growth has been increasing exponentially. And as a result of this data explosion, current data management methods can become inadequate. People are using the term big data (sometimes referred to as Big Data) to describe this latest industry trend. IBM is preparing the next generation of technology to meet these data management challenges.

To provide the capability of incorporating big data sources and analytics of these sources, IBM developed a stream-computing product that is based on the open source computing framework Apache Hadoop. Each product in the framework provides unique capabilities to the data management environment, and further enhances the value of your data warehouse investment.

In this IBM Redbooks publication, we describe the need for big data in an organization. We then introduce IBM InfoSphere BigInsights and explain how it differs from standard Hadoop. BigInsights provides a packaged Hadoop distribution, a greatly simplified installation of Hadoop and corresponding open source tools for application development, data movement, and cluster management. BigInsights also brings more options for data security, and as a component of the IBM big data platform, provides potential integration points with the other components of the platform.

A new chapter has been added to this edition. Chapter 11 describes IBM Platform Symphony, which is a new scheduling product that works with IBM Insights, bringing low-latency scheduling and multi-tenancy to IBM InfoSphere BigInsights. The book is designed for clients, consultants, and other technical professionals.

**INTERNATIONAL
TECHNICAL
SUPPORT
ORGANIZATION**

**BUILDING TECHNICAL
INFORMATION BASED ON
PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:
ibm.com/redbooks**

SG24-8077-01

ISBN 0738438286