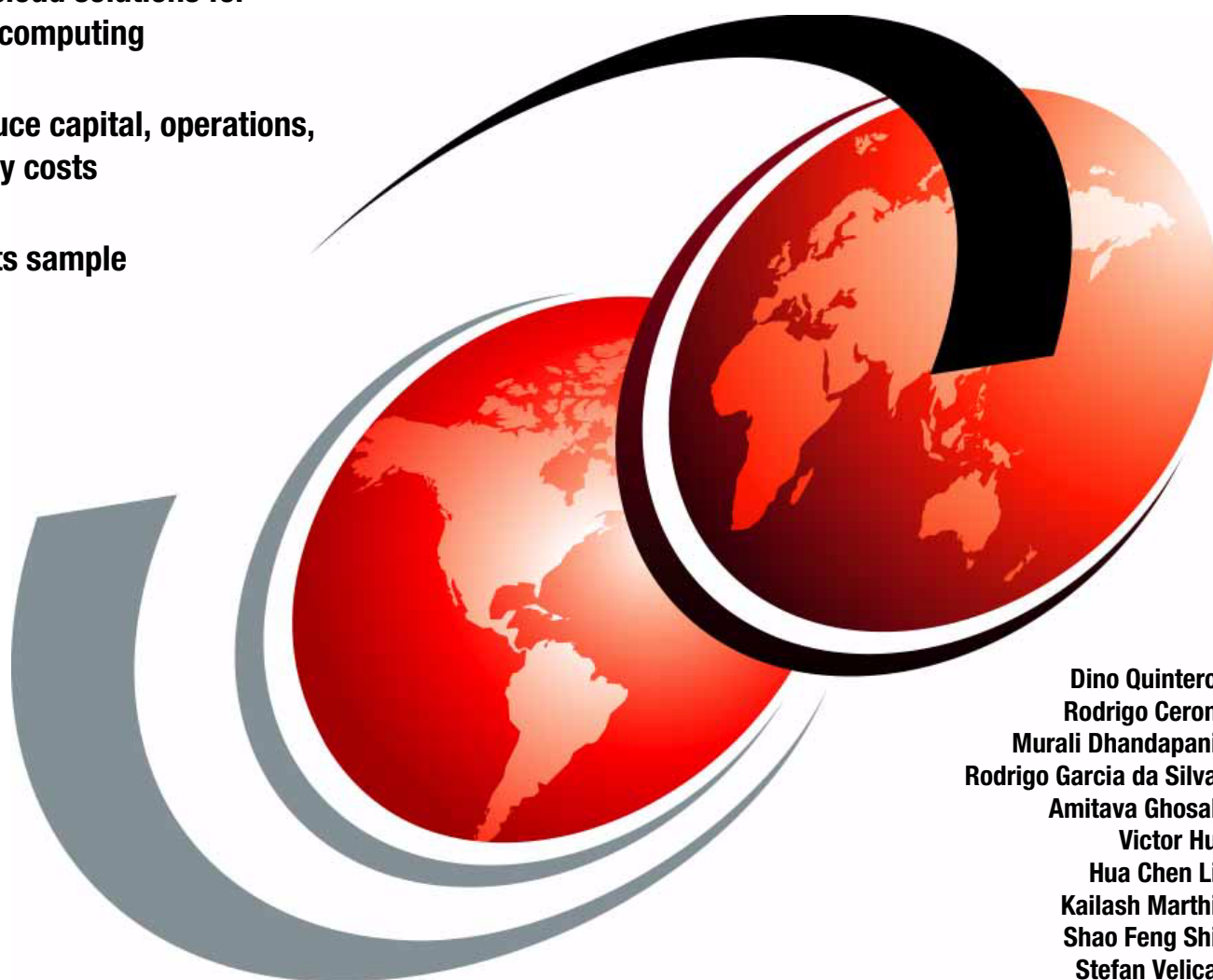


IBM Technical Computing Clouds

Provides cloud solutions for technical computing

Helps reduce capital, operations, and energy costs

Documents sample scenarios



Dino Quintero
Rodrigo Ceron
Murali Dhandapani
Rodrigo Garcia da Silva
Amitava Ghosal
Victor Hu
Hua Chen Li
Kailash Marthi
Shao Feng Shi
Stefan Velica

Redbooks



International Technical Support Organization

IBM Technical Computing Clouds

October 2013

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

First Edition (October 2013)

This edition applies to IBM InfoSphere BigInsights 2.0, IBM Platform Symphony 6.1, IBM Platform Process Manager 9.1 client for windows, IBM Platform Cluster Manager Advanced Edition (PCM-AE) 4.1, General Parallel File System Version 3 Release 5.0.7, Red Hat Enterprise Linux 6.2 x86_64.

© 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	vii
Trademarks	viii
 Preface	 ix
Authors	ix
Now you can become a published author, too!	xi
Comments welcome	xi
Stay connected to IBM Redbooks	xii
 Chapter 1. Introduction to technical cloud computing	 1
1.1 What is Technical Computing	2
1.1.1 History	2
1.1.2 Infrastructure	5
1.1.3 Workloads	6
1.2 Why use clouds?	7
1.2.1 Flexible infrastructure	8
1.2.2 Automation	8
1.2.3 Monitoring	9
1.3 Types of clouds	9
 Chapter 2. IBM Platform Load Sharing Facilities for technical cloud computing	 13
2.1 Overview	14
2.2 IBM Platform LSF family features and benefits	14
2.2.1 IBM Platform Application Center (PAC)	15
2.2.2 IBM Platform Process Manager (PPM)	16
2.2.3 IBM Platform License Scheduler	17
2.2.4 IBM Platform Session Scheduler	17
2.2.5 IBM Platform Dynamic Cluster	18
2.2.6 IBM Platform RTM	18
2.2.7 IBM Platform Analytics	19
2.3 IBM Platform LSF job management	19
2.3.1 Job submission	20
2.3.2 Job status	21
2.3.3 Job control	21
2.3.4 Job display	22
2.3.5 Job lifecycle	24
2.4 Resource management	24
2.5 MultiCluster	25
2.5.1 Architecture and flow	25
2.5.2 MultiCluster models	26
 Chapter 3. IBM Platform Symphony for technical cloud computing	 29
3.1 Overview	30
3.2 Supported workload patterns	31
3.2.1 Compute intensive applications	31
3.2.2 Data intensive applications	36
3.3 Workload submission	37
3.3.1 Commercial applications that are written to the Platform Symphony APIs	37
3.3.2 The symexec facility	38

3.3.3 Platform Symphony MapReduce client	38
3.3.4 Guaranteed task delivery	38
3.3.5 Job scheduling algorithms	39
3.3.6 Services (workload execution)	40
3.4 Advanced resource sharing	42
3.4.1 Lending	42
3.4.2 Borrowing	43
3.4.3 Resource sharing models	43
3.4.4 Heterogeneous environment support	45
3.4.5 Multi-tenancy	46
3.4.6 Resources explained	47
3.5 Dynamic growth and shrinking	48
3.5.1 Desktop and server scavenging	48
3.5.2 Virtual server harvesting	49
3.5.3 On-demand HPC capacity	50
3.6 Data management	52
3.6.1 Data-aware scheduling	53
3.7 Advantages of Platform Symphony	55
3.7.1 Advantages of Platform Symphony in Technical Computing Cloud	56
3.7.2 Multi-core optimizer	56
Chapter 4. IBM Platform Symphony MapReduce	59
4.1 Overview	60
4.1.1 MapReduce technology	60
4.1.2 Hadoop architecture	62
4.1.3 IBM Platform Symphony MapReduce framework	64
4.2 Key advantages for Platform Symphony MapReduce	71
4.2.1 Higher performance	71
4.2.2 Improved multi-tenant shared resource utilization	75
4.2.3 Improved scalability	82
4.2.4 Heterogeneous application support	83
4.2.5 High availability and resiliency	84
4.3 Key benefits	87
Chapter 5. IBM Platform Cluster Manager - Advanced Edition (PCM-AE) for technical cloud computing	89
5.1 Overview	90
5.2 Platform Cluster Manager - Advanced Edition capabilities and benefits	90
5.3 Architecture and components	94
5.3.1 Hardware	94
5.3.2 External software components	94
5.3.3 Internal software components	95
5.4 PCM-AE managed clouds support	95
5.5 PCM-AE: a cloud-oriented perspective	96
5.5.1 Cluster definition	96
5.5.2 Cluster deployment	98
5.5.3 Cluster flexing	100
5.5.4 Users and accounts	103
5.5.5 Cluster metrics	105
Chapter 6. The IBM General Parallel File System for technical cloud computing	111
6.1 Overview	112
6.1.1 High capacity	112
6.1.2 High performance	112

6.1.3 High availability	112
6.1.4 Single system image	113
6.1.5 Multiple operating system and server architecture support	113
6.1.6 Parallel data access	114
6.1.7 Clustering of nodes	114
6.1.8 Shared disks architecture	114
6.2 GPFS layouts for technical computing	115
6.2.1 Shared disk	115
6.2.2 Network block I/O	116
6.2.3 Mixed clusters	117
6.2.4 Sharing data between clusters	117
6.3 Integration with IBM Platform Computing products	119
6.3.1 IBM Platform Cluster Manager - Advanced Edition (PCM-AE)	119
6.3.2 IBM Platform Symphony	122
6.4 GPFS features for Technical Computing	124
6.4.1 Active File Management (AFM)	124
6.4.2 File Placement Optimizer (FPO)	129
Chapter 7. Solution for engineering workloads	139
7.1 Solution overview	140
7.1.1 Traditional engineering deployments	140
7.1.2 Engineering cloud solution	141
7.1.3 Key benefits	144
7.2 Architecture	146
7.2.1 Engineering cloud solution architecture	146
7.3 Components	149
7.3.1 Cloud service consumer	149
7.3.2 Security layer	150
7.3.3 Cloud services provider	150
7.3.4 Systems management	152
7.3.5 Third-party products	153
7.3.6 Hardware configuration	154
7.4 Use cases	158
7.4.1 Local workstation and remote cluster	160
7.4.2 Thin client and remote cluster	162
Chapter 8. Solution for life sciences workloads	173
8.1 Overview	174
8.1.1 Bioinformatics	174
8.1.2 Workloads	175
8.1.3 Trends and challenges	176
8.1.4 New possibilities	177
8.2 Architecture	178
8.2.1 Shared service models	179
8.2.2 Components	181
8.3 Use cases	183
8.3.1 Mixed workloads on hybrid clouds	184
8.3.2 Integration for life sciences private clouds	185
8.3.3 Genome sequencing workflow with Galaxy	193
Chapter 9. Solution for financial services workloads	199
9.1 Overview	200
9.1.1 Challenges	200
9.1.2 Types of workloads	201

9.2 Architecture	203
9.2.1 IBM Platform Symphony	203
9.2.2 General Parallel File System (GPFS)	206
9.2.3 IBM Platform Process Manager (PPM)	207
9.3 Use cases	210
9.3.1 Counterparty CCR and CVA	210
9.3.2 Shared grid for high-performance computing (HPC) risk analytics	213
9.3.3 Real-time pricing and risk	214
9.3.4 Analytics for faster fraud detection and prevention	215
9.4 Third-party integrated solutions	218
9.4.1 Algorithmics Algo One	218
9.4.2 SAS	220
Chapter 10. Solution for oil and gas workloads	223
10.1 Overview	224
10.1.1 Enhance exploration and production	225
10.1.2 Workloads	225
10.1.3 Application software	227
10.2 Architecture	227
10.2.1 Components	228
Chapter 11. Solution for business analytics workloads	233
11.1 MapReduce	234
11.1.1 IBM InfoSphere BigInsights	234
11.1.2 Deploying a BigInsights workload inside a cloud	235
11.2 NoSQL	243
11.2.1 HBase	245
Related publications	247
IBM Redbooks	247
Other publications	247
Online resources	248
Help from IBM	248

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.


COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

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

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

Active Cloud Engine™	IBM PureData™	PureFlex™
AIX®	IBM SmartCloud®	pureScale®
Algo®	iDataPlex®	RackSwitch™
Algo One®	InfoSphere®	Rational®
Algorithmics®	Intelligent Cluster™	Redbooks®
BigInsights™	Jazz™	Redbooks (logo)  ®
BNT®	LoadLeveler®	RiskWatch®
DataStage®	LSF®	Storwize®
DB2®	Mark-to-Future®	Symphony®
developerWorks®	POWER®	System p®
eServer™	Power Systems™	System Storage®
GPFS™	PowerHA®	System x®
HACMP™	PowerLinux™	SystemMirror®
IBM®	PowerVM®	Tivoli®
IBM Flex System™	PureData™	WebSphere®

The following terms are trademarks of other companies:

PostScript, the Adobe logo, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

Intel, Intel Xeon, Itanium, Pentium, 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

This IBM® Redbooks® publication highlights IBM Technical Computing as a flexible infrastructure for clients looking to reduce capital and operational expenditures, optimize energy usage, or re-use the infrastructure.

This book strengthens IBM SmartCloud® solutions, in particular IBM Technical Computing clouds, with a well-defined and documented deployment model within an IBM System x® or an IBM Flex System™. This provides clients with a cost-effective, highly scalable, robust solution with a planned foundation for scaling, capacity, resilience, optimization, automation, and monitoring.

This book is targeted toward technical professionals (consultants, technical support staff, IT Architects, and IT Specialists) responsible for providing cloud-computing solutions and support.

Authors

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

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

Rodrigo Ceron is a Master Inventor and Consultant in IBM Lab Services and Training Latin America, in Brazil. He has 13 years of experience in the UNIX/Linux area and nine years working at IBM, where he received eight patents in multiple areas. He graduated with honors in Computer Engineering from the University of Campinas (UNICAMP) and holds an IEEE CSDA credential. His areas of expertise include IBM Power Systems™ high availability, performance, cloud, and analytics. He has also published papers about Operations Research in IBM developerWorks®.

Murali Dhandapani is a Certified IT Specialist in Systems Management in IBM India. He works for the IBM India Software Lab Operations team, where he is a technical lead for IBM Rational® Jazz™ products infrastructure, high availability, and disaster recovery deployment. He has 10 years of experience. His areas of expertise include Linux, IBM AIX®, IBM POWER® Virtualization, IBM PowerHA® SystemMirror®, System Management, and Rational tools. Murali has a Master of Computer Science degree. He is an IBM developerWorks Contributing Author, IBM Certified Specialist in System p® administration, and an IBM eServer™ Certified Systems Expert - pSeries High Availability Cluster Multi-Processing (IBM HACMP™).

Rodrigo Garcia da Silva is an Accredited IT Architect and Technical Computing Client Technical Architect with the IBM System and Technology Group in Brazil. He joined IBM in 2007, and has a total of 11 years of experience in the IT industry. He holds a B.S. in Electrical Engineering from Universidade Estadual de Campinas. His areas of expertise include High Performance Computing, systems architecture, OS provisioning, Linux, systems

management, and open source software development. He also has a strong background in intellectual capital protection, including publications and patents. Rodrigo is responsible for Technical Computing presales support of customers in the Oil and Gas, Automotive, Aerospace, Life Sciences, and Higher Education industries in Brazil. Rodrigo is a previous Redbooks author of IBM POWER 775 HPC Solutions, SG24-8003 and IBM Platform Computing Solutions, SG24-8073.

Amitava Ghosal has worked on Network Devices, OS, Storage, Databases, and Software Platforms. His core expertise is on AIX platform on which he has worked for over five years and related technologies like IBM GPFS™ and HACMP. He currently works as an SME - Manager in one of the biggest Telecom projects of IBM India.

Victor Hu is an Advisory Software Engineer working at the IBM Poughkeepsie site, which is in upstate New York. He currently works on HPC Cloud Solutions using Platform Cluster Manager Advanced Edition. He has 8 years of experience in the High Performance Computing field working extensively on IBM Tivoli® Workload Scheduler LoadLeveler®. He holds a Bachelors of Science degree in Computer Science from the University of Michigan, Ann Arbor.

Hua Chen Li is a Staff Software Engineer in IBM China. His areas of knowledge include IBM Power Systems, System x, BladeCenter®, General Parallel File System (GPFS), and IBM Platform Computing. He has three years of experience in the Linux field and two years of experience in the High Performance Computing field. He was given a Contribution to IBM System p Linux Project Award by IBM for his work on PowerLinux™ project in 2011, and is now working on GPFS test. He has a master's degree in Software Engineering from University of Science and Technology of China.

Kailash Marthi is a High Performance Computing Cloud Architect. He currently works on Cloud solutions for BigData Analytics as one of the architects of the IBM SmartCloud for Analytics. Kailash has been a developer/architect for various IBM HPC products over 16 years of his career with IBM.

Shao Feng Shi is an Advisory Software Engineer from IBM China development lab in Shanghai. He has 7 years of experience in Java enterprise application and SOA development, and is a domain expert in the IBM globalization and localization process. His areas of expertise include Rational Software Architect, IBM WebSphere® application server, Java web application development, SOA solution design, and Web 2.0 technologies. Shao Feng Shi holds a master degree in Computer Science from Shanghai Jiao Tong University.

Stefan Velica is an IT Specialist who currently works for IBM Global Technologies Services in Romania. He has seven years of experience with IBM Power Systems. He is a Certified Specialist for IBM System p Administration, HACMP for AIX, High-end and Entry/Midrange DS Series, and Storage Networking Solutions. His areas of expertise include IBM System Storage®, SAN, PowerVM®, AIX, PowerHA, and GPFS. Stefan holds a bachelor degree in Electronics and Telecommunications Engineering from the Polytechnic Institute of Bucharest.

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

Ella Bushlovic

International Technical Support Organization, Poughkeepsie Center

Gordon McPheeters

Linda Cham

Robert Blackmore

Gautam Shah

IBM Poughkeepsie

Hari Reddy
IBM Dallas

Prasenjit Sarkar
IBM Almaden

Radhika A. Parameswaran
IBM India

Gord Sissons
Jeff Karmiol
IBM Canada

Ying Tao
Yi Li Wang
IBM China

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>



Introduction to technical cloud computing

This chapter introduces the concept of technical computing, the value of cloud computing, and the types of cloud for enterprises.

This chapter includes the following sections:

- ▶ What is Technical Computing
- ▶ Why use clouds?
- ▶ Types of clouds

1.1 What is Technical Computing

This section describes Technical Computing.

1.1.1 History

This section introduces the history of high-performance computing (HPC) and how Technical Computing became mainstream.

Traditional high-performance computing (HPC)

The IT Industry has always tried to maintain a balance between demands from business to deliver services against cost considerations of hardware and software assets. On one hand, business growth depends on information technology (IT) being able to provide accurate, timely, and reliable services. On other hand, there is cost associated with running IT services. These concerns have led to the growth and development of HPC.

HPC has traditionally been the domain of powerful computers (called “supercomputers”) owned by governments and large multinationals. Existing hardware was used to process data and provide meaningful information to single systems working with multiple parallel processing units. Limitations were based on hardware and software processing capabilities. Due to the cost associated with such intensive hardware, the usage was limited to a few nations and corporate entities.

The advent of the workflow-based processing model and virtualization as well as high availability concepts of clustering and parallel processing have enabled existing hardware to provide the performance of the traditional supercomputers. New technologies such as graphics processing units (GPUs) have pushed power of the existing hardware to perform more complicated functions faster than previously possible. Virtualization and clustering have made it possible to provide a greater level of complexity and availability of IT services. Sharing of resources to reduce cost has also become possible due to virtualization. There has been a move from a traditionally static IT model based on maximum load sizing to a leaner IT model based on workflow-based resource allocation through smart clusters. With the introduction of cloud technology, the resource requirement is becoming more on-demand as compared to the traditional forecasted demand, thus optimizing cost considerations further.

These technological innovations have made it possible to push the performance limits of existing IT resources to provide high performance output. The technical power to achieve computing results can be achieved with much cheaper hardware using smart clusters and grids of shared hardware. With workflow-based resource allocation, it is possible to achieve high performance from a set of relatively inexpensive hardware working together as a cluster. Performance can be enhanced by breaking across silos of IT resources, lying dormant to provide on-demand computing power wherever required. Data intensive industries such as engineering and life sciences can now use the computing power on demand provided by the workflow-based technology. Using parallel processing by heterogeneous resources that work as one unit under smart clusters, complex unstructured data can be processed to feed usable information into the system.

Mainstream Technical Computing

With the reduction in the cost of hardware resources, the demand for HPC has spread technical computing from scientific labs to mainstream commercial applications (Figure 1-1 on page 3). Technical computing has been demanded from sectors such as aerodynamics, automobile design, engineering, financial services, and oil and gas Industries. Improvement

in cooling technology and power management of these superfast computing grids have allowed users to extract more efficiency and performance from existing hardware.

Increased complexity of applications and demand for faster analysis of data has led Technical Computing to become widely available. Thus, IBM Technical Computing is focused on helping clients to transform their IT infrastructure to accelerate results. The goal of Technical Computing in mainstream industries is to meet the challenges of applications that require high performance computing, faster access to data, and intelligent workload management.

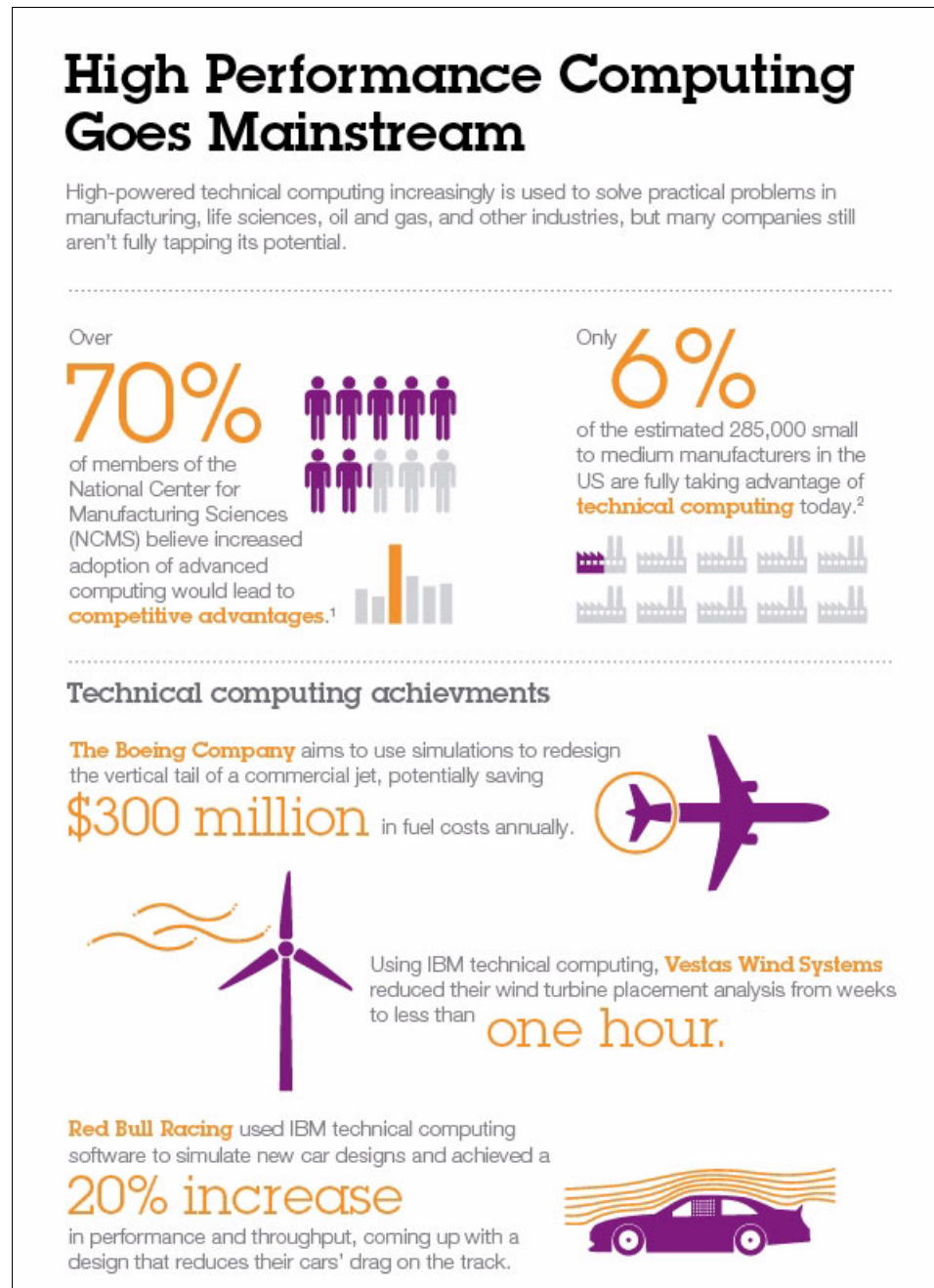


Figure 1-1 Technical Computing goes mainstream

Defining cluster, grids, and clouds

The following provides a description of the terminology used in this book.

- Cluster** Typically an application or set of applications whose primary aim is to provide improved performance and availability at a lower cost as compared to a single computing system.
- Grid** Typically a distributed system of homogeneous or heterogeneous computer resources for general parallel processing of related workflow that is usually scheduled using advanced management policies.
- Cloud** A system (private or public) that allows on-demand self service such as resource creation on demand, dynamic sharing of resources, and elasticity of resource sizing based on advanced workflow models.

IBM Platform Computing solutions have gone through the evolution from cluster to grid to cloud due to its abilities to manage heterogeneous complexities of distributed computing resources. Figure 1-2 shows the evolution of clusters, grids, and HPC clouds.

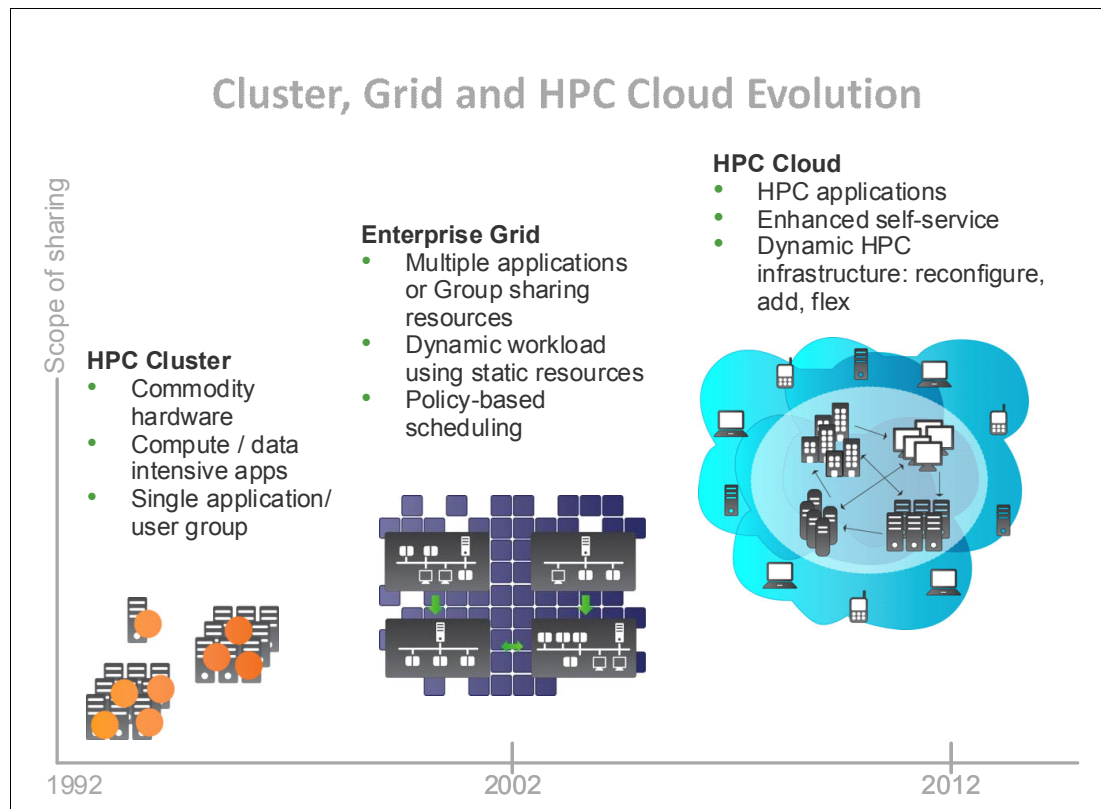


Figure 1-2 Cluster, grid and High Performance Computing (HPC) cloud evolution

IBM Platform Computing provides solutions for mission-critical applications that require complex workload management across heterogeneous environment for diverse industries from life sciences to engineering and financial sectors that involve complex risk analysis. IBM Platform Computing has a 20-year history of working on highly complex solutions for some of the largest multinational companies. It has proven examples of robust management of highly complex workflow across large distributed environments that deliver results.

1.1.2 Infrastructure

This section provides a brief overview of the components (hardware, software, storage) available to help deploy a technical computing cloud environment. The following sections provide a subset of the possible solutions.

Hardware (computational hardware)

IBM HPC and IBM Technical Computing provide flexibility in your choice of hardware and software:

- ▶ IBM System x
- ▶ IBM Power Systems
- ▶ IBM General Parallel File System (GPFS)
- ▶ Virtual infrastructure OpenStack

Software

In addition to this list, IBM Platform Computing provides support to heterogeneous cluster environments with extra IBM or third-party software (Figure 1-3):

- ▶ IBM Platform LSF®
- ▶ IBM Platform Symphony®
- ▶ IBM Platform Computing Management Advanced Edition (PCMAE)
- ▶ IBM InfoSphere BigInsights
- ▶ IBM GPFS
- ▶ Bare Metal Provisioning through xCAT
- ▶ Solaris Grid Engine
- ▶ Open Source Apache Hadoop
- ▶ Third party schedulers

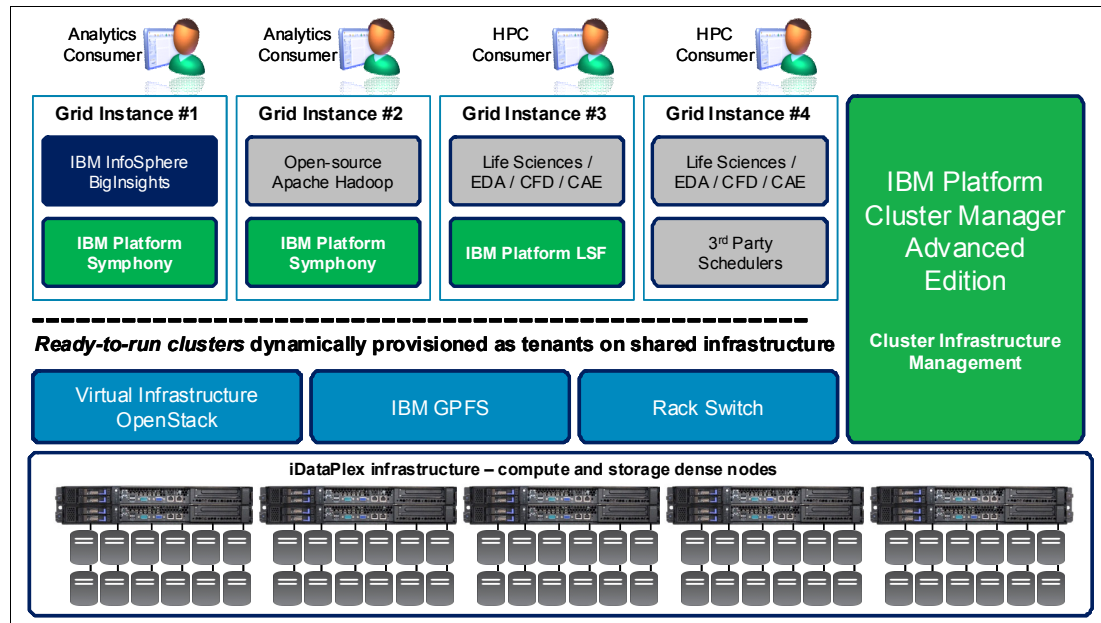


Figure 1-3 Overview of Technical Computing and analytics clouds solution architecture

Networking (high bandwidth, low latency)

IBM Cluster Manager tools help use the bandwidth of the network devices to lower the latency levels. The following are some of the devices supported:

- ▶ IBM RackSwitch™ G8000, G8052, G8124, and G8264
- ▶ Mellanox InfiniBand Switch System IS5030, SX6036, and SX6512
- ▶ Cisco Catalyst 2960 and 3750 switches

Storage (parallel storage and file systems)

IBM Cluster Manager tools use storage devices capable of high parallel I/O to help provide efficient I/O related operations in the cloud environment. The following are some of the storage devices that are used:

- ▶ IBM DCS3700
- ▶ IBM System x GPFS Storage Server

1.1.3 Workloads

Technical computing workloads have the following characteristics:

- ▶ Large number of systems
- ▶ Heavy resource usage including I/O
- ▶ Long running workloads
- ▶ Dependent on parallel storage
- ▶ Dependent on attached storage
- ▶ High bandwidth, low latency networks
- ▶ Compute intensive
- ▶ Data intensive

The next section provides a few technologies that support technical computing workloads.

Message Passing Interface (MPI)

HPC clusters frequently employ a distributed memory model to divide a computational problem into elements that can be simultaneously run in parallel on the hosts of a cluster. This often involves the requirement that the hosts share progress information and partial results by using the cluster's interconnect fabric. This is most commonly accomplished by using a message passing mechanism. The most widely adopted standard for this type of message passing is the MPI standard, which is described on the following website:

<http://www.mpi-forum.org>

IBM Platform MPI is a high-performance and production-quality implementation of the MPI standard. It fully complies with the MPI-2.2 standard, and provides enhancements such as low latency and high bandwidth point-to-point and collective communication routines over other implementations.

For more information about IBM Platform MPI, see the *IBM Platform MPI User's Guide*, SC27-4758-00, at:

<http://www-01.ibm.com/support/docview.wss?uid=pub1sc27475800>

Service-oriented architecture (SOA)

SOA is a software architecture in which the business logics are encapsulated and defined as services. These services can be used and reused by one or multiple systems that participate in the architecture. SOA implementations are generally platform-independent, which means that infrastructure considerations do not get in the way of deploying new systems or

enhancing existing systems. Many financial institutions deploy a range of technologies, so the heterogeneous nature of SOA is particularly important.

IBM Platform Symphony combines a fast service-oriented application middleware component with a highly scalable grid management infrastructure. Its design delivers reliability and flexibility, while also ensuring low levels of latency and high throughput between all system components.

For more information about SOA, see:

https://www14.software.ibm.com/webapp/iwm/web/signup.do?source=stg-web&S_PKG=ov11676DCW03015USEN-Building%20a%20SOA%20infrastructure.pdf

MapReduce

MapReduce is a programming model for applications that process large volumes of data in parallel by dividing the work into a set of independent tasks across many systems. MapReduce programs in general transform lists of input data elements into lists of output data elements in two phases: Map and reduce.

MapReduce is widely used in the data intensive computing such as business analytics and life science. Within IBM Platform Symphony, the MapReduce framework supports data-intensive workload management using a special implementation of service-oriented application middleware to manage MapReduce workloads.

Parallel workflows

Workflow is a task that is composed by a sequence of connected steps. In HPC clusters, many workflows run in parallel to complete a job or to respond to a batch of requests. As the complexity increases, workflows become more complicated. Workflow automation is becoming increasingly important for these reasons:

- ▶ Jobs must run at the correct time and in the correct order
- ▶ Mission critical processes have no tolerance for failure
- ▶ There are inter-dependencies between steps across systems

Clients need an easy-to-use and cost efficient way to develop and maintain the workflows.

Visualization

Visualization is a typical workload in engineering for airplane and automobile designers. The designers create large computer-aided design (CAD) environments to run their 2D/3D graphic calculations and simulations for the products. These workloads demand a large hardware environment that includes graphic workstations, storage, and software tools. In addition to the hardware, the software licenses are also expensive. Thus, the designers are looking to reduced costs, and expect to share the infrastructure between computer-aided engineering (CAE) and CAD.

1.2 Why use clouds?

Implementing a cloud infrastructure can be the ideal solution for companies who do not want to invest in a separate cluster infrastructure for technical computing workloads. It can reduce, among other things, extra hardware and software costs and avoid the extra burden of another cluster administration. Cloud also provides the benefits of request on demand and release on demand after the work is completed, which saves time for deployments and the expenses to a certain extent. For technical computing, the hardware requirements are usually large considering the workloads that it must manage. Although the physical hardware runs better in

HPC environments, evolving virtualization technologies have started to provide room for HPC solutions as well. Using a computing cloud for HPC environments can help eliminate the static usage of the infrastructure. It can also help provide a way to use the hardware resources dynamically as per the computing requirements.

1.2.1 Flexible infrastructure

Cloud computing provides the flexibility to use the resources when required. In terms of a technical computing cloud environment, cloud computing not only provides the flexibility to use the resources on demand, but helps to provision the computing nodes as per the application requirement to help manage the workload. By implementing and using IBM Platform Computing Manager (PCM), dynamic provisioning of the computing nodes with the wanted operating systems is easily achieved. This dynamic provisioning solution helps to better use the hardware resources and fulfill various technical computing requirements for managing the workloads. Figure 1-4 shows the infrastructure of an HPC cloud.

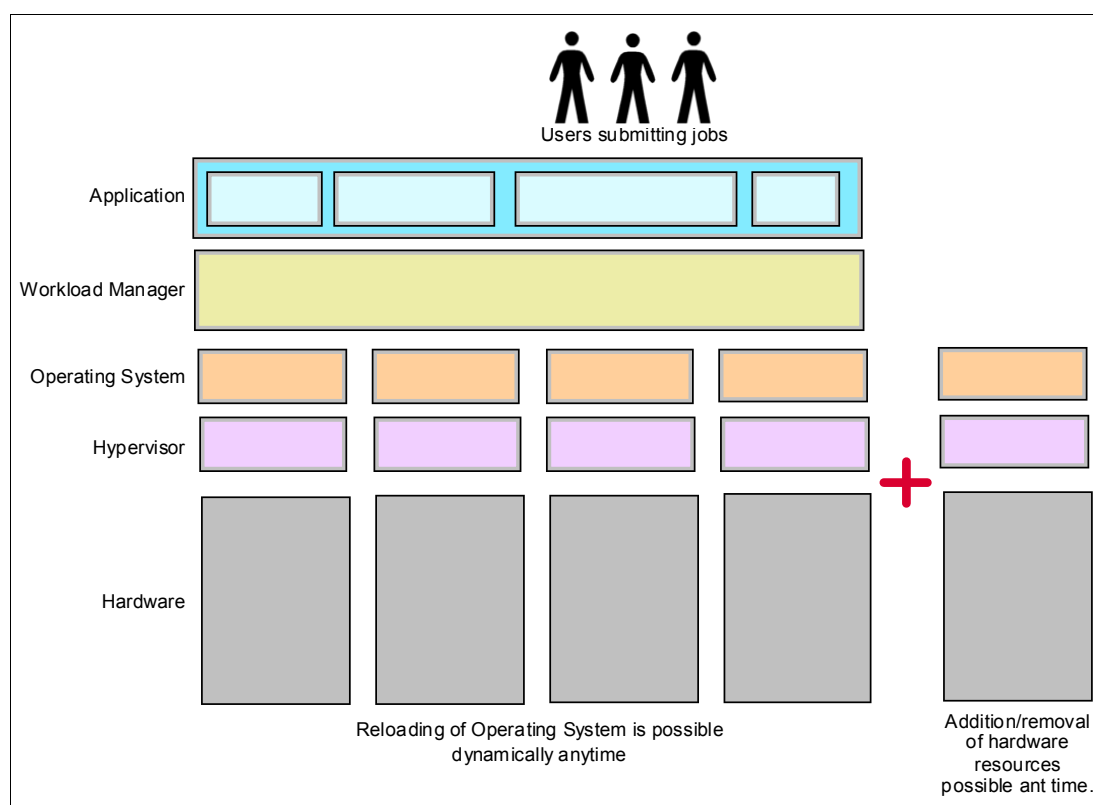


Figure 1-4 Flexible infrastructure with cloud

1.2.2 Automation

Cloud computing can significantly reduce manual effort during installation, provisioning, configuration, and other tasks that were performed manually before. When done manually, these computing resource management steps can take a significant amount of time. A cloud-computing environment can dramatically help reduce the system management complexity by implementing automation, business workflows, and resource abstractions.

IBM PCMAE provides many automation features to help reduce the complexity of managing a cloud-computing environment:

- ▶ Rapidly deployment of multiple HPC heterogeneous clusters in a shared hardware pool.
- ▶ Self-service, which allows users to request a custom cluster, specifying size, type, and time frame.
- ▶ Dynamically grow and shrink (flex up and down) the size of a deployed cluster based on workload demand, calendar, and sharing policies.
- ▶ Share hardware across clusters by rapidly reprovisioning the resources to meet the infrastructure needs (for example, Windows and Linux, or a different version of Linux).

These automation features reduce the time that is required to make the resources available to clients.

1.2.3 Monitoring

In a cloud computing environment, many computers, network devices, storage, and applications are running. To achieve high availability, throughput, and resource utilization, clouds have monitoring mechanisms. Monitoring measures the service and resource usage, which is key for charge back to the users. The system statistics are collected and reported to the cloud provider or user, and based on these figures, dashboards can be generated.

Monitoring provides the following benefits:

- ▶ Avoids outages by checking the health of the cloud-computing environment
- ▶ Improves resource usage to help lower costs
- ▶ Identifies performance bottlenecks and optimizes workloads
- ▶ Predicts usage trend

IBM SmartCloud Monitoring 7.1 is a bundle of established IBM Tivoli infrastructure management products, including IBM Tivoli Monitoring and IBM Tivoli Monitoring for Virtual Environments. The software delivers dynamic usage trending and health alerts for pooled hardware resources in the cloud infrastructure. The software includes sophisticated analytics, and capacity reporting and planning tools. You can use these tools to ensure that the cloud is handling workloads quickly and efficiently.

For more information about IBM SmartCloud Monitoring, see the following website:

<http://www-01.ibm.com/software/tivoli/products/smartcloud-monitoring/>

1.3 Types of clouds

There are three different cloud-computing architectures:

- ▶ Private clouds
- ▶ Public clouds
- ▶ Hybrid clouds

A private cloud is an architecture where the client encapsulates its IT capacities “as a service” over an intranet for their exclusive use. The cloud is owned by the client, and is managed and hosted by the client or a third party. The client defines the ways to access the cloud. The advantage is that the client controls the cloud so that security and privacy can be ensured. Also, the client can customize the cloud infrastructure based on its business needs. A private cloud can be cost effective for a company that owns many computing resources.

A public cloud provides standardized services for public use over the Internet. Usually it is built on standard and open technologies, providing web page, API or SDK for the consumers to use the services. Benefits include standardization, capital preservation, flexibility, and improved time to deploy.

Clients can integrate a private cloud and a public cloud to deliver computing services, which is called hybrid cloud computing. Figure 1-5 highlights the differences and relationships of these three types of clouds.

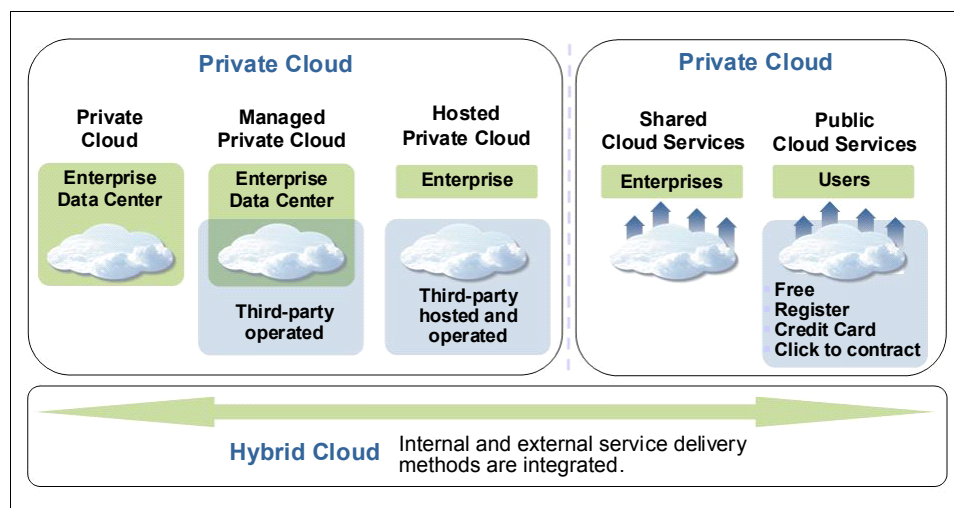


Figure 1-5 Types of clouds

Why an IBM HPC cloud

IBM HPC clouds can help enable transformation of both your IT infrastructure and business. Based on an HPC cloud's potential impact, clients are actively evolving their infrastructure toward private clouds, and beginning to consider public and hybrid clouds. Clients are transforming their existing infrastructure to HPC clouds to enhance the responsiveness, flexibility, and cost effectiveness of their environment. This transformation helps clients enable an integrated approach to improve computing resource capacity and to preserve capital. Eventually the client will access extra cloud capacity by using the cloud models described in Figure 1-5.

In a public cloud environment, HPC must overcome a number of significant challenges as shown in Table 1-1.

Table 1-1 Challenges of HPC in a public cloud

Challenges in a public cloud	
Security	<ul style="list-style-type: none"> ▶ Cloud providers do not provide guarantees for data protection ▶ IP in-flight outside the firewall and on storage devices
Application licenses	<ul style="list-style-type: none"> ▶ Legal agreements (LTUs) can limit licenses to geographic areas or corporate sites ▶ Unlimited licenses can be significantly more expensive
Business advantage	<ul style="list-style-type: none"> ▶ Cloud resources are expensive compared to local resources if used incorrectly ▶ Building and automating business policy for using cloud can be difficult

Challenges in a public cloud	
Performance	<ul style="list-style-type: none"> ► If applications run poorly in a private cloud, the applications will not improve in public clouds
Data movement	<ul style="list-style-type: none"> ► Data must be replicated in the cloud before jobs can run ► Providers charge for data in/out and storage

When using private clouds, HPC might not suffer from the public cloud barriers, but there are other common issues as shown in Table 1-2.

Table 1-2 Issues that a private cloud can address for High Performance Computing (HPC)

Issues	Details
Inefficiency	<ul style="list-style-type: none"> ► Less than fully used hardware ► High labor cost to install, monitor, and manage HPC environments ► Constrained space, power, and cooling
Lack of flexibility	<ul style="list-style-type: none"> ► Resource silos that are tied to a specific project, department, or location ► Dependency on specific individuals to run technical tasks
Delayed time to value	<ul style="list-style-type: none"> ► Long provisioning times ► Limited ability to fulfill peak demand ► Constrained access to special purposes devices (for example, GPUs)

Figure 1-6 shows the IBM HPC cloud reference model.

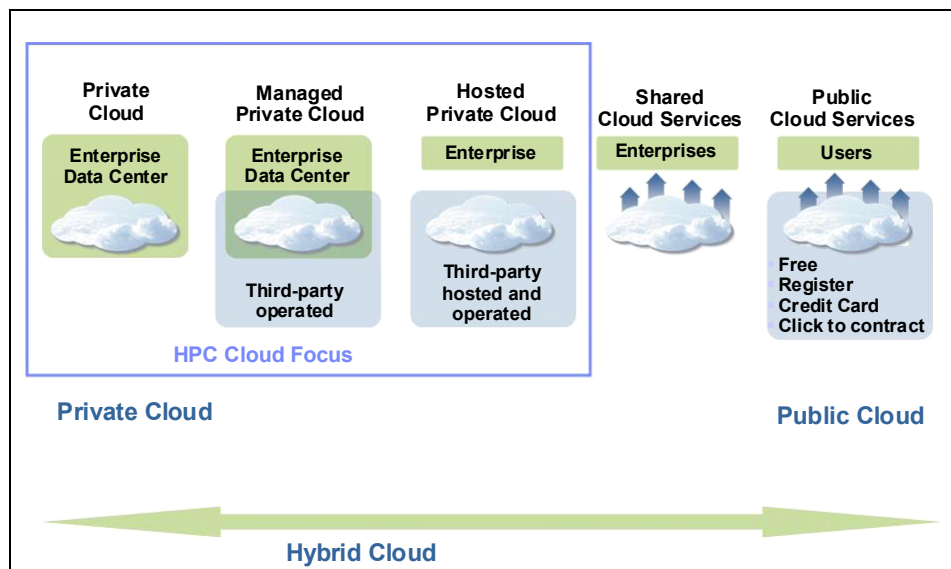


Figure 1-6 IBM HPC cloud

The HPC private cloud has three hosting models: Private cloud, managed private cloud, and hosted private cloud. Table 1-3 describes the characteristics of these models.

Table 1-3 Private cloud models

Private cloud model	Characteristics
Private cloud	Client self hosted and managed
Managed private cloud	Client self hosted, but third-party managed
Hosted private cloud	Hosted and managed by a third party



IBM Platform Load Sharing Facilities for technical cloud computing

This chapter describes the advantages and features of IBM Platform LSF for technical computing clusters workload management in a cloud-computing environment.

This chapter includes the following sections:

- ▶ Overview
- ▶ IBM Platform LSF family features and benefits
- ▶ IBM Platform LSF job management
- ▶ Resource management
- ▶ MultiCluster

2.1 Overview

IBM Platform Load Sharing Facility (LSF) is a powerful workload manager for demanding, distributed, and mission-critical high-performance computing (HPC) environments. Whenever you want to address complex problems, simulation scenarios, extensive calculations, or anything that needs compute power and run them as jobs, submit them to Platform LSF through commands in a technical cloud-computing environment.

Figure 2-1 shows a Platform LSF cluster with a master host (server-01), a master candidate (server-02) host, and other hosts that communicate with each other through the Internet Protocol network.

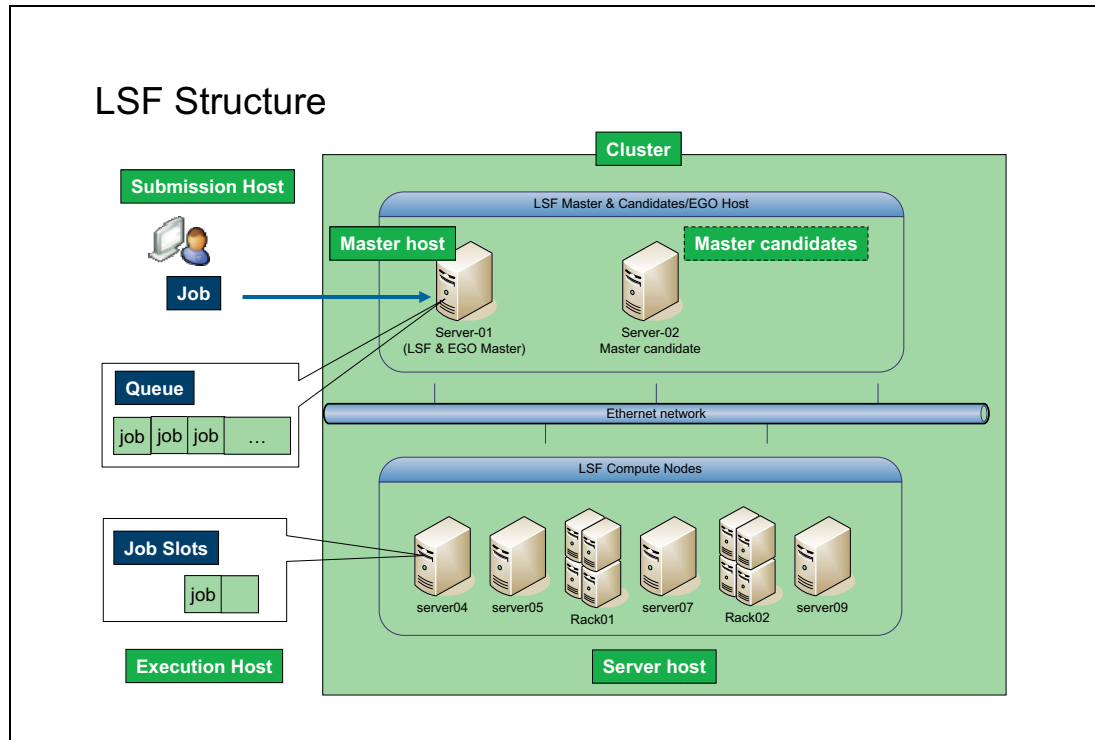


Figure 2-1 IBM Platform LSF cluster structure

The master host is required by the cluster and is also the first host installed. When server-01 fails, server-02 takes over server-01's work as a failover host. Jobs wait in queues until the available resources are ready. The submission host, which can be in a server host or a client host, submits a job with the **bsub** command. A basic unit of work is assigned into a job slot as a bucket in the Platform LSF cluster. Server hosts not only submit but also run the jobs. As shown in Figure 2-1, server04 can act as an execution host and run the job.

2.2 IBM Platform LSF family features and benefits

The Platform LSF family is composed of a suite of products that address many common customer workload management requirements. IBM Platform LSF boasts the broadest set of capabilities in the industry. What differentiates IBM Platform LSF from many competitors is that all of these components are tightly integrated and fully supported. The use of an integrated family also reduces strategic risk because although you might not need a capability today, it is available as your needs evolve. These are the core benefits of an integrated, fully

supported product family. The purpose of the IBM Platform LSF family (Figure 2-2) is to address the many challenges specific to Technical Computing environments.

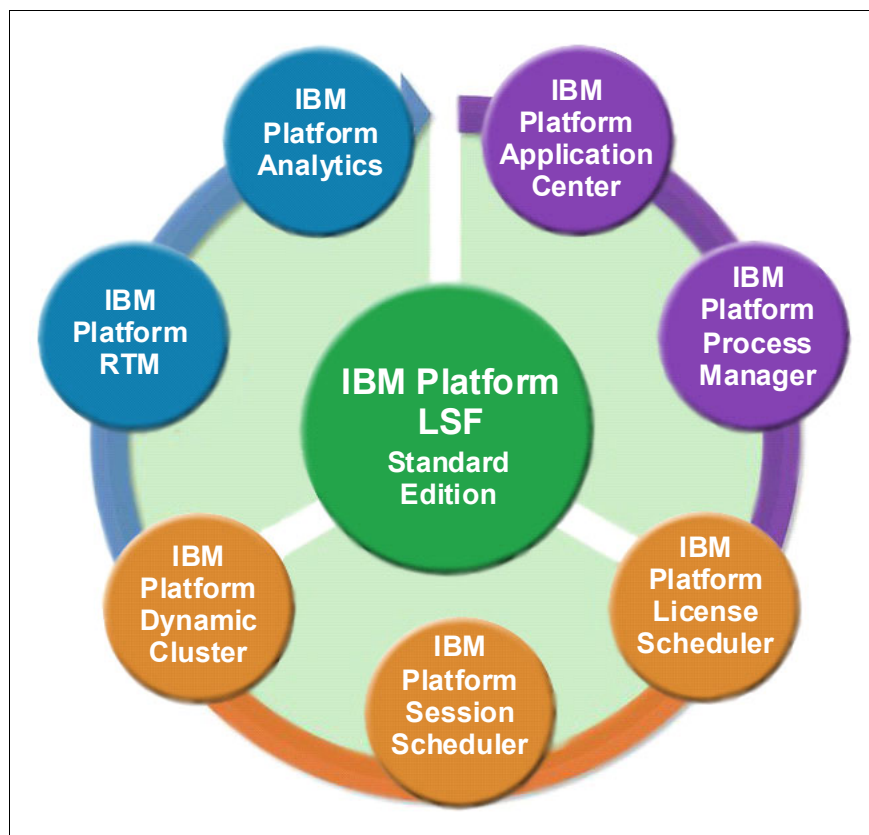


Figure 2-2 IBM Platform LSF product family

The IBM Platform LSF family includes these products:

- ▶ IBM Platform Application Center (PAC)
- ▶ IBM Platform Process Manager (PPM)
- ▶ IBM Platform License Scheduler
- ▶ IBM Platform Session Scheduler
- ▶ IBM Platform Dynamic Cluster
- ▶ IBM Platform RTM
- ▶ IBM Platform Analytics

The following sections describe each optional add-on product in the IBM Platform LSF family.

2.2.1 IBM Platform Application Center (PAC)

IBM Platform Application Center is an optional add-on product to IBM Platform LSF that enables users and administrators to manage applications more easily through a web interface. This add-on product allows cloud to switch environments to run different types of workloads. IBM Platform Application Center is integrated with IBM Platform License Scheduler, IBM Platform Process Manager, and IBM Platform Analytics. Users can access cluster resources locally or remotely with a browser, monitor cluster health, and customize application to meet cloud-computing needs.

IBM Platform Application Center offers many benefits for clients who implement cloud-computing solutions:

- ▶ Easy-to-use web-based management for cloud environments
- ▶ Enhanced security especially for remote cloud users
- ▶ Interactive console support, configurable workflows and application interfaces that are based on role, job notification, and flexible user-accessible file repositories

IBM Platform Application Center offers many benefits for users in cloud:

- ▶ Helps increase productivity
- ▶ Improves ability to collaborate on projects with peers
- ▶ Provides an easier interface that translates into less non-productive time interacting with the help desk
- ▶ Helps reduce errors, which translates into less time wasted troubleshooting failed jobs

2.2.2 IBM Platform Process Manager (PPM)

IBM Platform Process Manager is a powerful interface for designing and running multi-step HPC workflows in a Technical Computing cloud. The process manager is flexible to accommodate complex and real-world workflows. Often similar workflows have submodules shared between flows. Thus, by supporting subflows, modularity is promoted, making flows much easier to maintain.

The process manager enables grid-aware workflows or individual Platform LSF jobs to be triggered based on complex calendar expressions of external events. The process manager can improve process reliability and dramatically reduce administrator workloads with support for sophisticated flow logic, subflows, alarms, and scriptable interfaces.

Process flows can be automated over a heterogeneous, distributed infrastructure. Because the hosts to run individual workflow steps on are chosen at run time, processes automated by using the process manager inherently run faster and more reliably. This is because the process manager interacts with Platform LSF to select the best available host for the workload step.

The IBM Platform Process Manager provides the following benefits for managing workloads in a cloud-computing environment:

- ▶ Provides a full visual environment. This means that flows can be created quickly and easily, and they are inherently self-documenting. Someone else can look at a flow and easily understand the intent of the designer, making workflow logic much easier to manage and maintain.
- ▶ Helps capture repeatable best practices. Processes that are tedious, manual, and error-prone today can be automated, saving administrator time and helping get results faster.
- ▶ Makes it much faster to design and deploy complex workflows, enabling customers to work more efficiently.
- ▶ Enables repetitive business processes such as reporting or results aggregation to run faster and more reliably by making workflows resilient and speeding their execution.
- ▶ Scales seamlessly on heterogeneous clusters of any size.
- ▶ Reduces administrator effort by automating various previously manual workflows.

2.2.3 IBM Platform License Scheduler

IBM Platform License Scheduler allocates licenses based on flexible sharing policies. Platform License Scheduler helps ensure that scarce licenses are allocated in a preferential way to critical projects. It also enables cross-functional sharing of licenses between departments and lines of business.

In many environments, the cost of software licenses exceeds the cost of the infrastructure. Monitoring how licenses are being used, and making sure that licenses are allocated to the most business critical projects is key to containing costs. The Platform License Scheduler can share application licenses according to policies.

The IBM Platform License Scheduler provides many benefits for clients who implement cloud-computing solutions:

- ▶ Improves license utilization. This is achieved by breaking down silos of license ownership and enabling licenses to be shared across clusters and departments.
- ▶ Designed for extensibility, supporting large environments with many license features and large user communities with complex sharing policy requirements.
- ▶ Improves service levels by improving the chances that scarce licenses are available when needed. This is especially true for business critical projects.
- ▶ Improves productivity because users do not need to wait excessive periods for licenses.
- ▶ Enables administrators to get visibility of license usage either by using license scheduler command line tools, or through the integration with the IBM Platform Application Center.
- ▶ Improves overall license utilization, thus removing the practical barriers to sharing licenses and ensuring that critical projects have preferential access to needed licenses.

2.2.4 IBM Platform Session Scheduler

IBM Platform Session Scheduler implements a hierarchical, personal scheduling paradigm that provides a low-latency execution. With low latency per job, Platform Session Scheduler is ideal for running short jobs, whether they are a list of tasks, or job arrays with parametric execution.

Scheduling large numbers of jobs reduces run time. With computers becoming ever faster, the execution time for individual jobs is becoming very short. Many simulations such as designs of experiments or parametric simulations involve running large numbers of relatively short-running jobs. For these types of environments, cloud users might need a different scheduling approach for efficient running of high-volumes of short running jobs.

IBM Platform Session Scheduler can provide Technical Computing cloud users with the ability to run large collections of short duration tasks within the allocation of a Platform LSF job. This process uses a job-level task scheduler that allocates resources for the job once, and then reuses the allocated resources for each task.

The IBM Platform Session Scheduler makes it possible to run large volumes of jobs as a single job. IBM Platform Session Scheduler provides many benefits for clients who implement cloud-computing solutions:

- ▶ Provides higher throughput and lower latency
- ▶ Enables superior management of related tasks
- ▶ Supports over 50,000 jobs per user
- ▶ Particularly effective with large volumes of short duration jobs

2.2.5 IBM Platform Dynamic Cluster

IBM Platform Dynamic Cluster turns static Platform LSF clusters into a dynamic cloud infrastructure. By automatically changing the composition of the clusters to meet ever-changing workload demands, service levels are improved and organizations can do more work with less infrastructure. Therefore, Platform Dynamic Cluster can transform static, low utilization clusters into highly dynamic and shared cloud cluster resources.

In most environments, it is not economically feasible to provision for peak demand. For example, one day you might need a cluster of 100 Windows nodes, and the next day you might need similar sized Linux cluster. Ideally, clusters flex on demand, provisioning operating systems and application environments as needed to meet changing demands and peak times. IBM Platform Dynamic Cluster can dynamically expand resources on demand, which enables jobs to float between available hardware resources.

Platform Dynamic Cluster can manage and allocate the cloud infrastructure dynamically through these mechanisms:

- ▶ Workload driven dynamic node reprovisioning
- ▶ Dynamically switching nodes between physical and virtual machines
- ▶ Automated virtual machines (VMs) live migration and checkpoint restart
- ▶ Flexible policy controls
- ▶ Smart performance controls
- ▶ Automated pending job requirement

The IBM Platform Dynamic Cluster provides many benefits for clients who implement cloud-computing solutions:

- ▶ Optimizes resource utilization
- ▶ Maximizes throughput and reduces time to results
- ▶ Eliminates costly, inflexible silos
- ▶ Increases reliability of critical workloads
- ▶ Maintains maximum performance
- ▶ Improves user and administrator productivity
- ▶ Increases automation, decreasing manual effort

2.2.6 IBM Platform RTM

As the number of nodes per cluster, and the number of clusters increases, management becomes a challenge. Corporations need monitoring and management tools that enable administrator time to scale and manage multiple clusters globally. With better tools, administrators can find efficiencies, reduce costs, and improve service levels by identifying and resolving resource management challenges quickly.

IBM Platform RTM is the most comprehensive workload monitoring and reporting dashboard for Platform LSF cloud environments. It provides monitoring, reporting, and management of clusters through a single web interface. This enables Platform LSF administrators to manage multiple clusters easily while providing a better quality of service to cluster users.

IBM Platform RTM provides many benefits for clients who implementing cloud-computing solutions:

- ▶ Simplifies administration and monitoring. Administrators can monitor both workloads and resources for all clusters in their environment using a single monitoring tool.
- ▶ Improves service levels. For example, you can monitor resources requirements to make sure that Platform LSF resources requests are not “over-requesting” resources relative to what they need and leaving idle cycles.

- ▶ Resolves issues quickly. Platform RTM monitors key Platform LSF services and quickly determine reasons for pending jobs.
- ▶ Avoids unnecessary service interruptions. With better cluster visibility and cluster alerting tools, administrators can identify issues before the issues lead to outages. Examples of issues include a standby master host that is not responding, and a file system on a master host that is slowly running out of space in the root partition. Visibility of these issues allows them to be dealt with before serious outages.
- ▶ Improves cluster efficiency. Platform RTM gives administrators the tools they need to measure cluster efficiency, and ensure that changes in configuration and policies are steadily improving efficiency-related metrics.
- ▶ Realizes better productivity. User productivity is enhanced for these reasons. The cluster runs better, more reliably and at a better level of utilization with higher job throughput because administrators have the tools they need to identify and remove bottlenecks. Administrators are much more productive as well because they can manage multiple clusters easily and reduce the time that they spend investigating issues.

2.2.7 IBM Platform Analytics

HPC managers also need to deal with the business challenges around infrastructure, planning capacity, monitoring services levels, apportioning costs, and so on. HPC managers need tools that translate raw data that are gathered from their environments into real information on which they can base decisions.

IBM Platform Analytics is aimed specifically at business analysts and IT managers because the tool translates vast amount of information collected from multiple clusters into actionable information. Business decisions can be based on this information to provide better utilization and performance of the technical computing environments.

IBM Platform Analytics provides many benefits for clients who implement cloud-computing solutions:

- ▶ Turns data into decision making. Organizations can transform vast amounts of collected data into actionable information based on which they can make decisions.
- ▶ Identifies and remove bottlenecks.
- ▶ Optimizes asset utilization. By understanding the demand for different types of assets exactly, you can use assets more efficiently.
- ▶ Gets more accurate capacity planning. You can spot trends in how asset use is changing to make capacity planning decisions that will intercept future requirements.
- ▶ Generates better productivity and efficiency. By analyzing cluster operations, administrators often find “low-hanging-fruit” where minor changes in configuration can yield substantial improvements in productivity and efficiency.

2.3 IBM Platform LSF job management

This section provides information about how to handle jobs in Platform LSF. The following topics are addressed in this section:

- ▶ Submit/modify jobs
- ▶ Manipulate (such as stop, resume) jobs
- ▶ View detailed job information

2.3.1 Job submission

The command **bsub** is used to submit jobs. **bsub** runs as an interactive command or can be part of a script. The jobs can be submitted using a host to define the jobs and set the job parameters.

If the command runs without any parameters, the job starts immediately in the default queue (usually the *normal* queue) as shown in Example 2-1.

Example 2-1 Submitting the job to the Platform LSF queue

```
bsub demo.sh
Job <635> is submitted to default queue <normal>.
```

```
bjobs
JOBID   USER   STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
635     user1   RUN   normal     HostA       HostB       demo.sh    Jul  3 11:00
```

To specify a queue, the **-q** flag must be added as shown in Example 2-2.

Example 2-2 Specifying a queue to run the job

```
bsub -q priority demo.sh
Job <635> is submitted to queue <priority>.
```

```
bjobs
JOBID   USER   STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
635     user1   RUN   priority   HostA       HostB       demo.sh    Jul  3 11:13
```

To start a job in a suspended state, *the* **-H** flag must be used as shown in Example 2-3.

Example 2-3 Starting a job in a suspended state

```
bsub -H demo.sh
Job <635> is submitted to default queue <normal>.
```

```
bjobs
JOBID   USER   STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
635     user1   PSUSP normal     HostA                               demo.sh    Jul  3 11:00
```

For a list of all flags (switches) supported by the **bsub** command, see *Running Jobs with IBM Platform LSF*, SC27-5307, at the following website or by using the online manual by typing the **man bsub** on the command line:

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

2.3.2 Job status

The command **bjobs** shows the status of jobs defined. Jobs keep changing status until they reach completion. Jobs can have one of following statuses:

Normal state:

PEND: Waiting in queue for scheduling and dispatch

RUN: Dispatched to host and running

DONE: Finished normally

Suspended state:

PSUSP: Suspended by owner or LSF Administrator while pending

USUSP: Suspended by owner or LSF Administrator while running

SSUSP: Suspended by the LSF system after being dispatched

2.3.3 Job control

Jobs can be controlled by using following commands:

The **bsub** command is used to start the submission of a job as shown in Example 2-4.

Example 2-4 Initial job submission

```
bsub demo.sh
Job <635> is submitted to default queue <normal>.

bjobs -d
JOBID  USER  STAT  QUEUE      FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
635    user1  DONE  priority   HostA      HostB      demo.sh   Jul  3 10:14
```

The **bstop** command is used to stop a running job (Example 2-5).

Example 2-5 Stopping a running job

```
bstop 635
Job <635> is being stopped

bjobs
JOBID  USER  STAT  QUEUE      FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
635    user1  USUSP normal     HostA      HostB      demo.sh   Jul  3 10:14
```

The **bresume** command is used to resume a previously stopped job (Example 2-6).

Example 2-6 Starting a previously stopped job

```
bresume 635
Job <635> is being resumed

bjobs
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
635	user1	RUN	normal	HostA	HostB	demo.sh	Jul 3 10:14

The **bkill** command is used to end (kill) a running job (Example 2-7).

Example 2-7 Ending a running job

```
bkill 635
Job <635> is being terminated
```

```
bjobs
No unfinished job found
```

```
bjobs -d
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
635    user1  EXIT  normal HostA      HostB      demo.sh   Jul 3 10:41
```

2.3.4 Job display

The command **bjobs** is used to display the status of jobs. The command can be used with a combination of flags (switches) to check for running and completed jobs. If the command is run without any flags, the output of the command shows all running jobs of a particular user as shown in Example 2-8.

Example 2-8 Output of the bjobs command

```
bjobs
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
635    user1  RUN   normal HostA      HostB      demo.sh   Jul 3 10:14
```

To view all completed jobs, the **-d** flag is required with the **bsub** command (Example 2-9).

Example 2-9 Viewing the completed jobs

```
bjobs -d
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
635    user1  EXIT  normal HostA      HostB      demo.sh   Jul 3 10:41
```

To view details of a particular job, the *job_id* must be specified after the **bsub** command as shown in Example 2-10.

Example 2-10 Viewing details of a job

```
bjobs 635
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
635    user1  EXIT  normal HostA      HostB      demo.sh   Jul 3 10:41
```

To check the details of a particular job, the `-l` flag must be specified after the **bjobs** command as shown in Example 2-11.

Example 2-11 Showing the details of a particular job

```
bjobs -l

Job <635>, User <user1>, Project <default>, Status <EXIT>, Queue <normal>, Comm
and <demo.sh>
Wed Jul  3 10:41:43: Submitted from host <HostA>, CWD <${HOME}>;
Wed Jul  3 10:41:44: Started on <HostB>, Execution Home </u/user1>, Execution
CWD </u/user1>;
Wed Jul  3 10:42:05: Exited with exit code 130. The CPU time used is 0.1 second
s.
Wed Jul  3 10:42:05: Completed <exit>; TERM_OWNER: job killed by owner.

MEMORY USAGE:
MAX MEM: 2 Mbytes;  AVG MEM: 2 Mbytes

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

      adapter_windows      poe nrt_windows
loadSched                   -    -    -
loadStop                    -    -    -

RESOURCE REQUIREMENT DETAILS:
Combined: select[type == local] order[r15s:pg]
Effective: select[type == local] order[r15s:pg]
```

A complete list of flags can be found in the man pages of the **bjobs** command (**man bjobs**).

2.3.5 Job lifecycle

Each job has a regular lifecycle in a Technical Computing cloud. In the lifecycle, the command **bjobs** shows the status of jobs defined. Jobs keep on changing status until they reach completion. The lifecycle process of the job is shown in Figure 2-3.

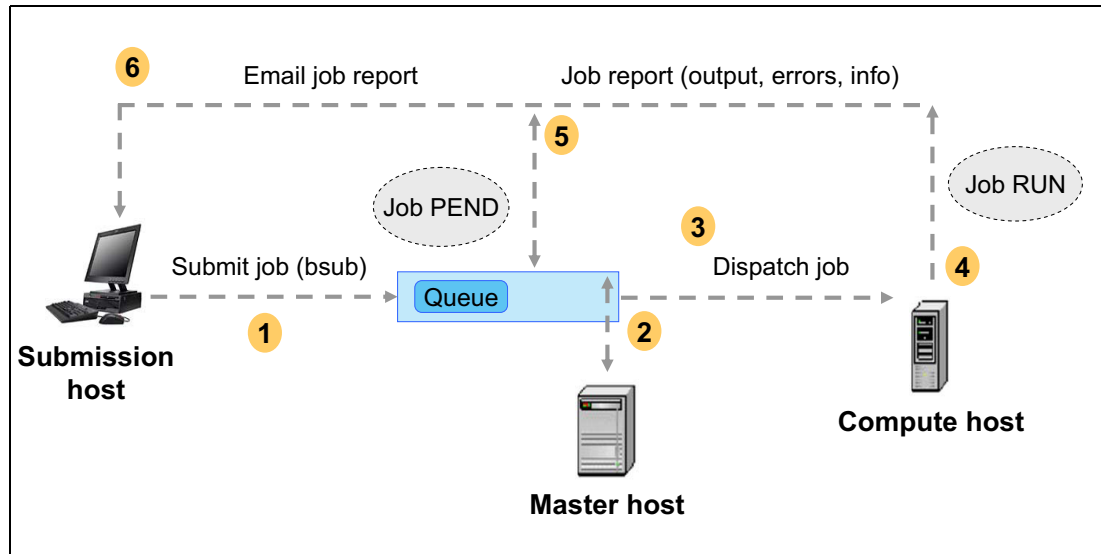


Figure 2-3 Job lifecycle

A host can submit a job by using the **bsub** command. The job's state in a waiting queue is PEND. Then, **mbatchd** at some point sends the jobs to **mbschd** for scheduling so that the job to compute host can be dispatched. When the compute host finishes running, the job is handled by **sbatchd**. There is a job report that indicates success or failure. Finally, the job report is sent by email back to the submission host, including CPU use, memory use, job output, errors, and so on. For more information about job lifecycle, see *Running Jobs with Platform LSF, Version 7.0 Update 6* at:

http://support.sas.com/rnd/scalability/platform/PSS5.1/lsf7.05_users_guide.pdf

2.4 Resource management

Individual systems are grouped into a cluster to be managed by Platform LSF. One system in the cluster is selected as the "master" for LSF. Each subordinate system in the cluster collects its own "vital signs" periodically and reports them back to the master. Users then submit their jobs to LSF and the master decides where to run the job based on the collected vital signs.

Platform LSF uses built-in and configured resources to track resource availability and usage. The LSF daemons on subordinate hosts in the cluster report resource usage periodically to the master. The master host collects all resource usage from all subordinate hosts. Users submit jobs with the resource requirements to LSF. The master decides where to dispatch the job for execution based on the resource required and current availability of the resource.

Resources are physical and logical entities that are used by applications to run. Resource is a generic term, and can include low-level things such as shared memory segments. A resource of a particular type has attributes. For example, a compute host has the attributes of memory, CPU utilization, and operating system type.

Platform LSF has some considerations to be aware of for the resources:

- ▶ Runtime resource usage limits. Limit the use of resources while a job is running. Jobs that consume more than the specified amount of a resource are signaled.
- ▶ Resource allocation limits. Restrict the amount of a resource that must be available during job scheduling for different classes of jobs to start, and which resource consumers the limits apply to. If all of the resource has been consumed, no more jobs can be started until some of the resource is released.
- ▶ Resource requirements. Restrict which hosts the job can run on. Hosts that match the resource requirements are the candidate hosts. When LSF schedules a job, it collects the load index values of all the candidate hosts and compares them to the scheduling conditions. Jobs are only dispatched to a host if all load values are within the scheduling thresholds.

For more information about resource limitations, see *Administering Platform LSF* at:

<http://www-01.ibm.com/support/docview.wss?uid=pub1sc22534600>

2.5 MultiCluster

This section describes the multiclustering features of IBM Platform LSF.

2.5.1 Architecture and flow

Within an organization, sites can have separate, independently managed LSF clusters. LSF MultiCluster can address scalability and ease of administration on different geographic locations.

In a multicluster environment, multiple components (submission cluster mbschd/mbatchd, execution cluster mbschd/mbatchd) work independently and asynchronously. Figure 2-4 shows the architecture and work flow of a MultiCluster.

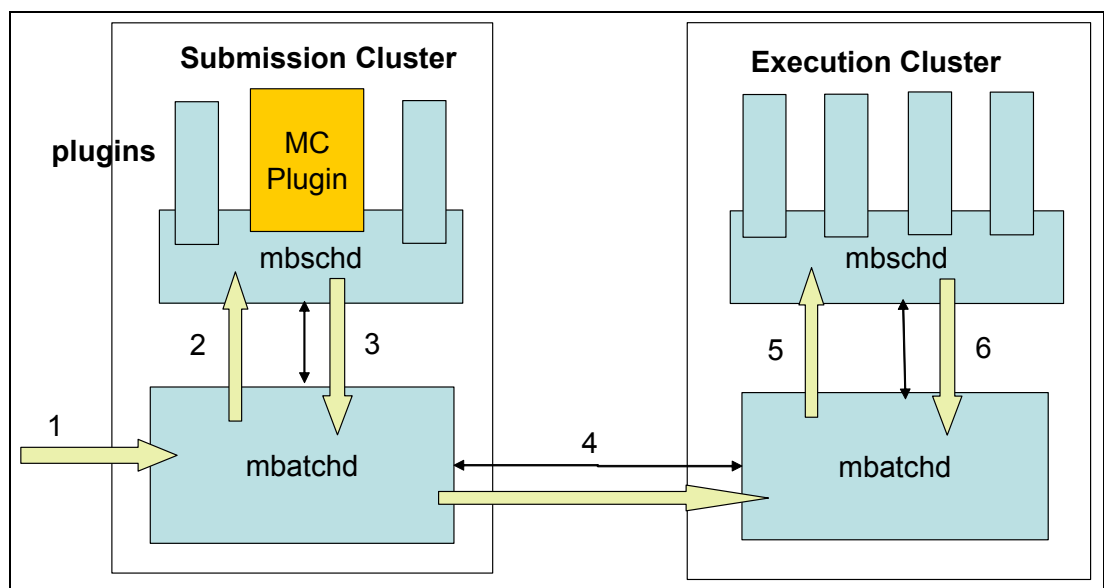


Figure 2-4 MultiCluster architecture and flow

Figure 2-4 on page 26 shows the submission cluster and the execution cluster with mbschd/mbatchd. The following is the workflow in the MultiCluster:

1. The user submits the job to a local submission cluster mbatchd.
2. The local submission cluster mbschd fetches newly submitted jobs.
3. The MultiCluster (MC) plug-in submission cluster makes the decision based on scheduling policies, and mbschd publishes the decision to the submission cluster mbatchd.
4. The submission cluster mbatchd forwards the job to the remote execution cluster mbatchd.
5. The execution cluster mbschd fetches newly forwarded jobs.
6. The execution cluster mbschd and the plug-ins make the job dispatch decision and publish the decision to the execution cluster mbatchd.

Resource availability information includes available slots, host type, queue status, and so on. After this workflow, the execution cluster mbatchd periodically collects its resource availability snapshot and sends it to the submission cluster. The execution cluster mbatchd triggers the call. Then, the submission cluster mbatchd receives resource availability information from the execution cluster mbatchd and keeps them locally until the next update interval to refresh the data. The submission cluster mbschd fetches resource availability information from the submission cluster mbatchd after every scheduling cycle and schedules the jobs based on it.

2.5.2 MultiCluster models

For a Technical Computing clouds environment, IBM Platform LSF MultiCluster provides two different types of share resources between clusters. The following section describes the two types: Job forwarding model and resource leasing model.

Job forwarding model

In this model, the cluster that is starving for resources sends the jobs over to the cluster that has resources to spare. To work together, the two clusters must set up compatible send-jobs and receive-jobs queues. With this model, scheduling of MultiCluster jobs is a process with two scheduling phases. The submission cluster selects a suitable remote receive-jobs queue, and forwards the job to it. The execution cluster then selects a suitable host and dispatches the job to it. This method automatically favors local hosts. A MultiCluster send-jobs queue always attempts to find a suitable local host before considering a receive-jobs queue in another cluster.

Resource leasing model

In this model, the cluster that is starving for resources takes resources away from the cluster that has resources to spare. To work together, the provider cluster must “export” resources to the consumer, and the consumer cluster must configure a queue to use these resources. In this model, each cluster schedules work on a single system image, which includes both borrowed hosts and local hosts.

These two models can be combined. For example, Cluster1 forwards jobs to Cluster2 using the job forwarding model, and Cluster2 borrows resources from Cluster3 using the resource leasing model. For more information about these types and how to select a model, see:

<http://www-01.ibm.com/support/docview.wss?uid=isg3T1016097>



IBM Platform Symphony for technical cloud computing

This chapter presents an overview of IBM Platform Symphony applied to the world of cloud computing. It includes a description of the role of IBM Platform Symphony in a grid environment, and an outline of IBM Platform Symphony's benefits. Also, this chapter includes a description of the IBM Platform Symphony characteristics that make it a good scheduler for Technical Computing cloud environments.

This chapter includes the following sections:

- ▶ Overview
- ▶ Supported workload patterns
- ▶ Workload submission
- ▶ Advanced resource sharing
- ▶ Dynamic growth and shrinking
- ▶ Data management
- ▶ Data management
- ▶ Advantages of Platform Symphony

3.1 Overview

One of the characteristics of a cloud environment is to provide better resource utilization of the hardware within it in the following ways:

- ▶ Allowing multiple workloads to be run on it.
- ▶ Allowing multiple users to access the software within it.
- ▶ Managing the resources with a middleware that is capable of quickly and effectively dispatching users' workloads to the cloud hardware.

Without these characteristics, clouds would not be dynamic nor effective for running most types of workloads, including ones that are close to real-time processing. Thus, the software controlling hardware resources of a cloud must be able to address these points. IBM Platform Symphony is a middleware layer that is able to tackle these points.

In a nutshell, Platform Symphony is a job scheduler that assigns resources to applications. An application sends the grid scheduler a load to be run. The grid scheduler then determines how to best dispatch that load onto the grid. This is where Platform Symphony fits into the overall cloud architecture for technical computing.

IBM Platform Symphony fits well in a cloud-computing environment because it fulfills the need for optimizing its resource utilization. The following are IBM Platform Symphony characteristics:

- ▶ Platform Symphony is based on a service-oriented architecture (SOA), serving hardware resources to applications when they have the need.
- ▶ Platform Symphony provides multi-tenancy support, which means it provides hardware resources to multiple applications simultaneously.
- ▶ Platform Symphony is a low-latency scheduler that can quickly and optimally distribute load to nodes based on workload needs and based on grid nodes utilization levels. This makes Platform Symphony capable of better using the hardware resources of a cloud and increase utilization levels.

All IBM Platform Symphony editions feature low-latency high-performance computing (HPC) SOA, and agile service and task scheduling. The editions range in scalability from one or two hosts for the developer edition to up to 5,000 hosts and 40,000 cores for the advanced edition. The following section explains the different editions:

- ▶ IBM Platform Symphony Developer Edition: Builds and tests applications without the need for a full-scale grid (available for download at no cost).
- ▶ IBM Platform Symphony Express Edition: For departmental clusters, where this is an ideal cost-effective solution.
- ▶ IBM Platform Symphony Standard Edition: This version is for enterprise class performance and scalability.
- ▶ IBM Platform Symphony Advanced Edition: This is the best choice for distributed compute and data intensive applications, including Hadoop MapReduce.

The next sections provide an overview of which types of workloads can be managed by IBM Platform Symphony, how applications interact with it, and some characteristics that makes Platform Symphony an effective scheduler for managing cloud resources. For more information about IBM Platform Symphony, see *IBM Platform Computing Solutions*, SG24-8073.

3.2 Supported workload patterns

IBM Platform Symphony is able to centralize two workload types that were usually managed separately in older HPC grids: Compute intensive and data intensive workloads. Instead of creating different grids for each type of workload, Platform Symphony can manage hardware resources for simultaneous access by both workload types. This is possible because Platform Symphony has software modules that can handle and optimize job execution for both of them. Figure 3-1 shows the high-level architecture of the IBM Platform Symphony components.

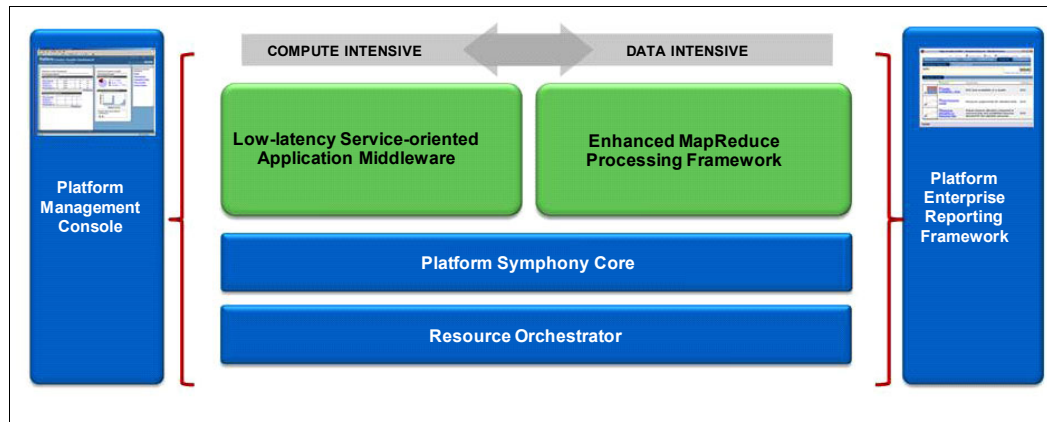


Figure 3-1 Platform Symphony components architecture

There are also other components outlined in Figure 3-1 such as the Platform Management Console and the Platform Enterprise Reporting Framework. This chapter describes Platform Symphony characteristics from the point of view of effectively managing resources in a cloud. For insight into Platform Symphony itself, see *IBM Platform Computing Solutions*, SG24-8073.

The next sections describe in more detail the characteristics of Platform Symphony for compute and data intensive workload types.

3.2.1 Compute intensive applications

Compute intensive workloads use processing power by definition. Two aspects come into play when it comes to optimizing this type of workload:

- Able to quickly provide computational resources to a job.
- Able to scale up the amount of computational resources that are provided to a job.

IBM Platform Symphony uses a different approach than other schedulers when it comes to job dispatching. Most schedulers receive input data from clients through slow communication protocols such as XML over HTTP. Platform Symphony, however, avoids text-based communication protocols and uses binary formats such as Common Data Representation (CDR) that allows for compacting data. This results in shorter transfer rates.

In addition to, and most importantly, Platform Symphony has a service session manager (SSM) that uses a different approach to deal with engines associated to for resource scheduling. Instead of waiting for the engines to poll the session manager for work, the state of each engine is known by the service session manager. Therefore, polling is not needed, which avoids significant delays in the dispatch of a job to the grid. Platform Symphony simply dispatches the job to engines that are available for processing it immediately. This behavior makes Platform Symphony a low-latency scheduler, which is a characteristic that is required by compute intensive workloads. Also, the service session manager itself runs faster as a result of a native HPC C/C++ implementation as opposed to the Java based implementations found in other schedulers.

Figure 3-2 compares IBM Platform Symphony's dispatch model with other scheduler's dispatch models.

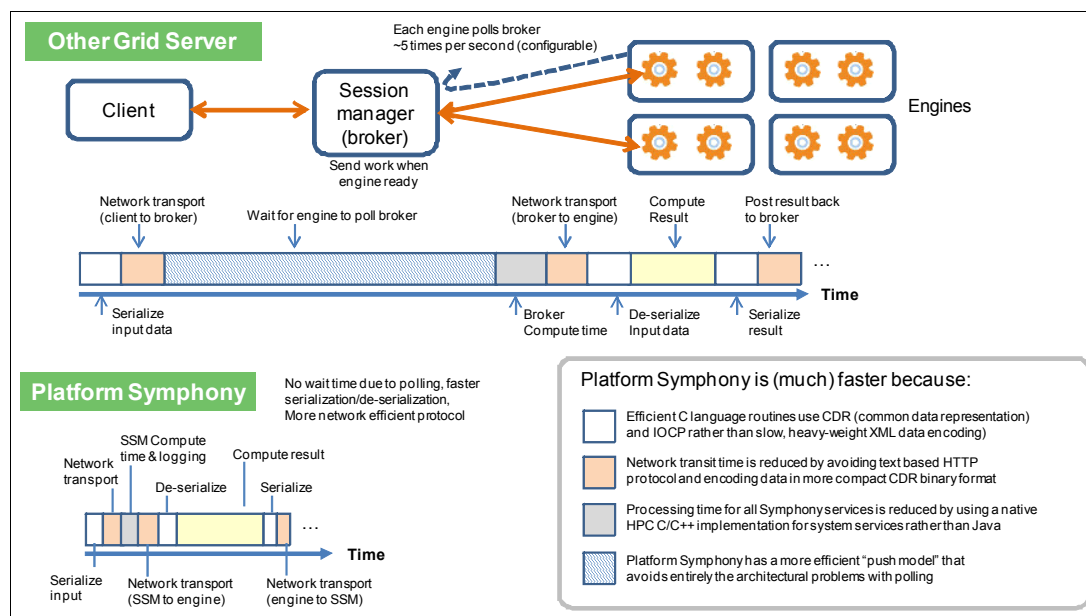


Figure 3-2 Platform Symphony's push-based scheduling versus other poll-based methods

The push-based scheduling allows Platform Symphony to provide low-latency and high throughput service to grid applications. Platform Symphony provides submillisecond responses, and is able to handle over 17,000 tasks per second. This is why it is able to scale much more than other schedulers as depicted in Figure 3-3.

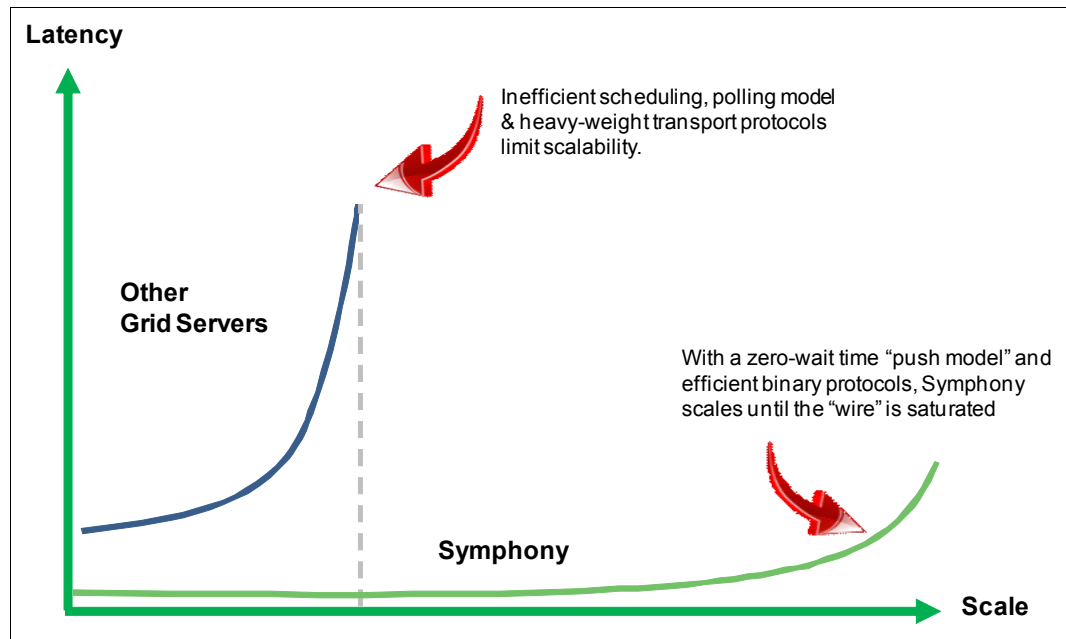


Figure 3-3 Symphony push-based scheduling allows it to scale up more than other schedulers

The second aspect is of great benefit to compute intensive workloads. Platform Symphony is able to scale up to 10,000 processor cores per application, 40,000 processor cores per individual grid, or it can reach up to 100,000 processor cores with its advanced edition version. Platform Symphony can therefore provide quick responses as a scheduler, and can provide application workloads with a large amount of computing power at once. When these two characteristics are combined, applications can compute their results much faster.

Besides low latency and large scaling capabilities of Platform Symphony, the following is a list of characteristics that makes it attractive for managing compute intensive workloads:

- Cost efficient and shared services:
 - Multi-tenant grid solution
 - Helps meet service level agreements (SLAs) while encouraging resource sharing
 - Easy to bring new applications onto the grid
 - Maximizes use of resources
- Heterogeneous and open:
 - Supports AIX, Linux, Windows, Windows HPC, Solaris
 - Provides connectors for C/C++, C#, R, Python, Java, Excel
 - Provides smart data handling and data affinity

HPC SOA model

Symphony is built on top of a low-latency, service-oriented application middleware layer for serving compute intensive workloads as depicted in Figure 3-1 on page 31. In this type of model, a client sends requests to a service, and the service generates results that are given back to the client. In essence, a SOA-based architecture is composed of two logic parts:

- Client logic (the client)
- Business logic (the service)

In this paradigm, there is communication between the client and the business logic layers constantly. The better the communication methods are, the quicker the responses are provided. This is where the ability of Platform Symphony to communicate with clients efficiently as explained in 3.2.1, “Compute intensive applications” on page 31 provides immediate benefits.

Clients can create multiple requests to the service, in which case multiple service instances are created to handle the requests as shown in Figure 3-4.

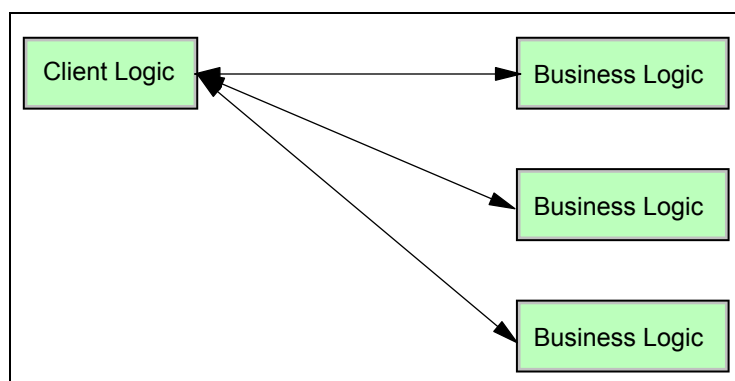


Figure 3-4 The SOA model

Platform Symphony works with this SOA model. Moreover, Platform Symphony can provide this type of service to multiple independent applications that require access to the grid resources. It does so through its SOA middleware, which can manage the business logic of these applications. It dispatches them to the grid for execution through the scheduling of its resources. This characterizes Platform Symphony as a multi-tenancy middleware, a preferred characteristic for grid and cloud environments.

Figure 3-5 shows the relationship among application clients, application business logic, and grid resources for serving the business logic.

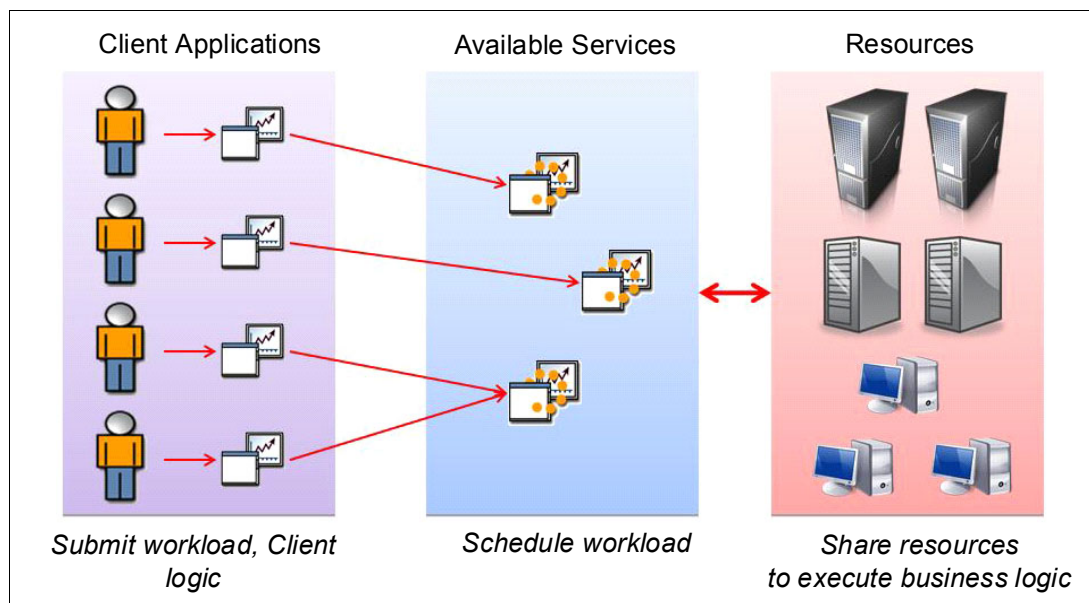


Figure 3-5 Client logic, business logic, and resource layers

Internally, the way that Platform Symphony handles SOA-based applications is shown in the abstraction hierarchy in Figure 3-6.

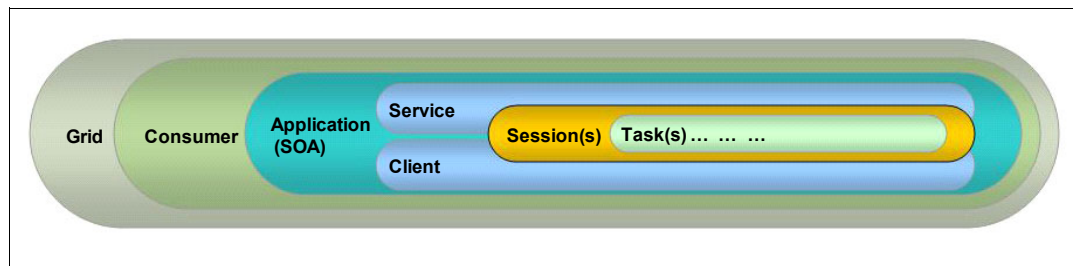


Figure 3-6 Abstraction hierarchy for the Platform Symphony SOA model

The following section is a brief description of each of the abstractions in Figure 3-6:

- Grid** The grid is an abstraction for all of the environment resources, which includes processors, memory, and storage units. Users and applications need to gain access to the grid resources to perform work.
- Consumer** This is the abstraction that organizes the grid resources in a structured way so that applications can use them. An application can only use the resources of the consumer it is assigned to. Consumers can be further organized hierarchically. This organization creates resource boundaries among applications and dictates how the overall grid resources are shared among them.
- Application** Uses resources from the grid through consumers. Each application has an application profile that defines every aspect of itself.
- Client** This is the client logic as presented in Figure 3-4 on page 34. It interacts with the grid through sessions. It sends requests to and receive results from the services.

Service	This is the business logic as presented in Figure 3-4 on page 34. It accepts requests from and returns responses to a client. Services can run as multiple concurrent instances, and they ultimately use computing resources.
Session	Abstraction that allows clients to interact with the grid. Each session has a session ID generated by the system. A session consists of a group of tasks that are submitted to the grid. Tasks of a session can share common data.
Task	The basic unit of computational work that can be processed in parallel with other tasks. A task is identified by a unique task ID within a session that is generated by the system.

3.2.2 Data intensive applications

Data intensive workloads consume data by definition. These data can, and usually are in a cloud or grid environment, be spread among multiple grid nodes. Also, these data can be in the scale of petabytes of data. Data intensive workloads must be able to process all of these data in a reasonable amount of time to produce results, otherwise there is little use for doing so. Therefore, a cloud or grid environment must have mechanisms that efficiently perform and process all of the data in a reasonable amount of time.

As depicted in Figure 3-1 on page 31, Platform Symphony has a component that specializes in serving data intensive workloads. It has a module that is composed of an enhanced MapReduce processing framework that is based on Hadoop MapReduce. MapReduce is an approach to processing large amounts of data in which nodes analyze data that are local to them (the map phase). After the data from all nodes is mapped, a second phase starts (the reduce phase) to eliminate duplicate data that might have been processed on each individual node. Platform Symphony's ability to use MapReduce algorithms makes it a good scheduler for serving data intensive workloads.

As an enterprise class scheduler, Platform Symphony includes extra scheduling algorithms when compared to a standard Hadoop MapReduce implementation. Symphony is able to deploy simultaneous MapReduce applications to the grid, with each one consuming part of the grid resources. This is as opposed to dispatching only one at a time and have it consume all of the grid resources. The Platform Symphony approach makes it easier to run data workloads that have SLAs associated with it. Shorter tasks whose results are expected sooner can be dispatched right away instead of being placed in the processing queue and having to wait until larger jobs are finished.

This integrated MapReduce framework brings the following advantages to Platform Symphony:

- ▶ Higher performance: Short MapReduce jobs run faster.
- ▶ Reliable and highly available rolling upgrades: Uses the built-in highly available components that allow dynamic updates.
- ▶ Dynamic resource management: Grid nodes can be dynamically added or removed.
- ▶ Co-existence of multiple MapReduce applications: You can have multiple applications based on the MapReduce paradigm. Symphony supports the co-existence of up to 300 of them.
- ▶ Advanced scheduling and execution: A job is not tied to a particular node. Instead, jobs have information about its processing requirements. Any node that meets the requirements is a candidate node for execution.

- ▶ Fully compatible with other Hadoop technologies: Java MR, Pig, Hive, HBase, Oozie, and others.
- ▶ Based on open data architecture: Has support for open standards file systems and databases.

For more information about Symphony's MapReduce framework, see Chapter 4, "IBM Platform Symphony MapReduce" on page 59.

Data affinity

Because data in a cloud or grid can be spread across multiple nodes, dispatch data consuming jobs to the nodes on which data is found to be local. This prevents the system from having to transfer large amounts of data from one node to another across the network. This latter scenario increases networking traffic and insert delays in the analysis of data due to data transfers.

Platform Symphony is able to minimize data transfers among nodes by applying the concept of data affinity. This applies to dispatching jobs that are supposed to consume the resulting data of a previous job to the same node of the previous job. This is different from the MapReduce characteristic of having each node process its local data. Here, data affinity is related to dispatching jobs that consume data that are related to one another onto the same node. This is possible because the SSM collects metadata about the data that are being processed on each node of a session.

For more information about data affinity, see 3.6, "Data management" on page 52.

3.3 Workload submission

Platform Symphony provides multiple ways for workload submission:

- ▶ Client-side application programming interfaces (APIs)
- ▶ Commercial applications that are written to the Platform Symphony APIs
- ▶ The symexec facility
- ▶ The Platform Symphony MapReduce client

These methods are addressed in the next sections.

3.3.1 Commercial applications that are written to the Platform Symphony APIs

Some applications use the Platform Symphony APIs to get access to the resource grid. This can be accomplished through .NET, COM, C++, Java, and other APIs. The best example is Microsoft Excel.

It is not uncommon to find Excel spreadsheets created to solve analytics problems, especially in the financial world, and in a shared calculation service approach that makes use of multiple computers. Symphony provides APIs that can be called directly by Excel for job submission. By doing so, calculations start faster and be completed faster.

There are five well-known patterns for integrating Excel with Symphony by using these APIs:

- ▶ Custom developed services: Uses Symphony COM API to call for distributed compute services.
- ▶ Command line utilities as tasks: Excel client calls Platform Symphony services that run scripts or binary files on compute nodes.

- ▶ Excel instances on the grid: Run parallel Excel instances that are called by client spreadsheets or other clients.
- ▶ Excel services by using user-defined functions (UDFs): Web-based client access that uses UDFs to distribute computations to the grid.
- ▶ Hybrid deployment scenarios: Combined UDF with Java, C++ services.

For more information, see the *Connector for Microsoft Excel User Guide*, SC27-5064-01.

3.3.2 The symexec facility

Symexec enables workloads to be called using the Platform Symphony service-oriented middleware without explicitly requiring that applications be linked to Platform Symphony client- and service-side libraries.

Symexec behaves as a consumer within Platform Symphony SOA model. With it, you can create execution sessions, close them, send a command to an execution session, fetch the results, and also run a session (create, execute, fetch, close, all running as an undetachable session).

For more information about symexec, see *Cluster and Application Management Guide*, SC22-5368-00.

3.3.3 Platform Symphony MapReduce client

MapReduce tasks can be submitted to Platform Symphony by either using the command line with the `mrsh` script command, or through the *Platform Management Console* (PMC).

It is also possible to monitor MapReduce submitted jobs by using command line through `soamview`, or also within the PMC.

For more information about how to submit and monitor MapReduce jobs to Symphony, see *User Guide for the MapReduce Framework in IBM Platform Symphony - Advanced Edition*, GC22-5370-00.

3.3.4 Guaranteed task delivery

Platform Symphony is built with redundancy of its internal components. Automatic fail-over of components and applications provide a high level of middleware availability. Even if the Enterprise Grid Orchestrator (EGO) fails, only the new requests for resource allocation are compromised. What had already been scheduled continues to work normally. EGO is a fundamental software piece that Platform Symphony uses to allocate resources in the grid.

A Platform Symphony managed grid infrastructure counts with a Platform Symphony master node that does grid resource management. If it fails, this service fails-over to other master candidate nodes. A similar fail-over strategy can be implemented for the SSM. A shared file system among the nodes facilitates the fail-over strategy, although it is also possible to achieve a degree of high availability without it through the management of previous runtime states.

As for the compute nodes, high availability is ensured by deploying application binary files and configuration to the local disk of the compute nodes themselves.

Figure 3-7 demonstrates the concepts presented.

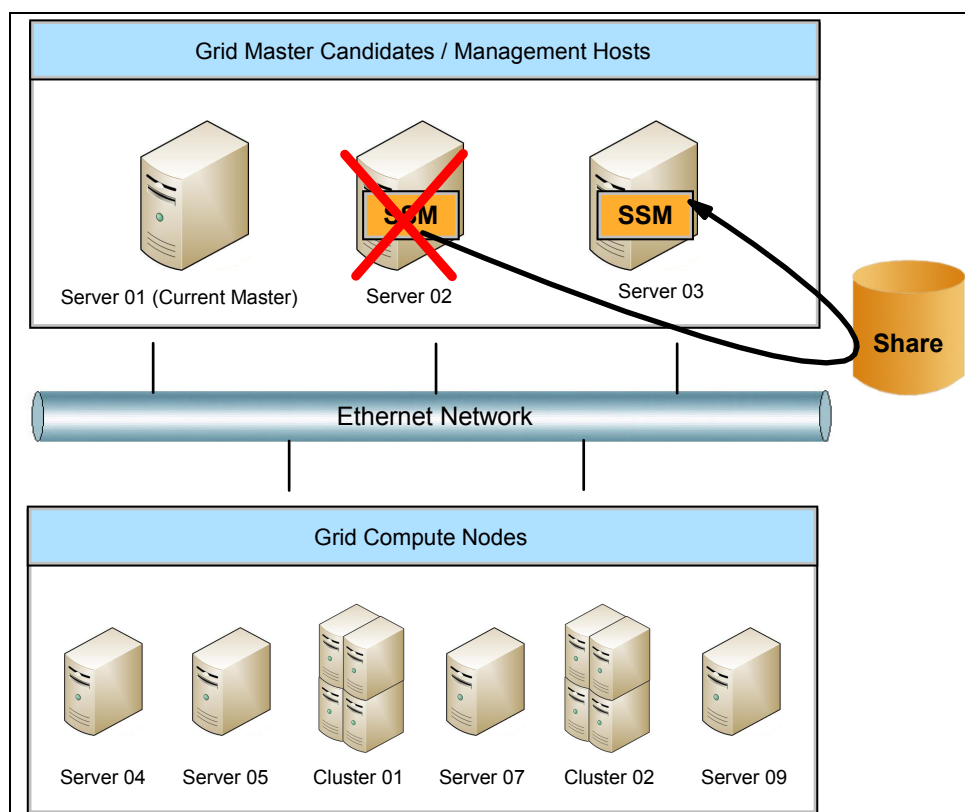


Figure 3-7 Platform Symphony components high availability

All of this component design can be used to ensure a guaranteed task delivery. That is, even if a component fails, tasks are deployed and run by spare parts of the solution.

Guaranteed task delivery is configurable, and depends on whether workloads are configured to be recoverable. If workloads are configured as recoverable, the applications do not need to resubmit tasks even in the case of an SSM failure. Also, reconnecting to another highly available SSM happens transparently and the client never notices the primary SSM has failed. The workload persists on the shared file system.

3.3.5 Job scheduling algorithms

Platform Symphony is equipped with algorithms for running workload submission to the grid. The algorithm determines how much resources and time each task is given. Job scheduling is run by the SSM, which can be configured to use the following algorithms to allocate resources slots (for example, processors) to sessions within an application:

Proportional scheduling

Allocates resources to a task based on its priority. The higher the priority, the more resources a task gets. Priorities can be reassigned dynamically. This is the default scheduling algorithm for workload submission.

Minimum service scheduling

Ensures a minimum number of service instances are associated with an application. Service instances do not allow resources to go below the minimum defined level even if there are no tasks to be processed.

Priority scheduling

All resources are scheduled to the highest priority session. If this session cannot handle them all, the remaining resources are allocated to the second highest priority session, and so on. Sessions with the same priority use creation time as tie-breaker: A newer session is given higher priority.

For more information about the concept of session and related concepts, see “High performance computing (HPC) SOA model” on page 31 and Figure 3-6 on page 33.

Preemption

Sometimes it is necessary to stop the execution of a task and free up its resources for other tasks. This process is called preemption.

Preemption occurs when under-allocated sessions must get resources from over-allocated tasks. It takes into consideration the algorithm in use and changes to that algorithm, and happens immediately. Preempted tasks are queued once again for dispatching. Task preemption is configurable and is not turned on by default.

Note: Task preemption is not turned on by default in the Platform Symphony scheduling configuration.

3.3.6 Services (workload execution)

Workload execution allows you to deploy existing executable files in a cloud grid environment. An execution task is a child process that is run by a Platform Symphony service instance using a command line specified by a Platform Symphony client.

The Platform Symphony execution application allows you to start and control the remote execution of executable files. Each application is made up of an execution service and an executable file that are distributed among compute hosts. The management console implements a workload submission window that allows a single command to be submitted per session to run the executable file.

You can do this by using the GUI interface. Click **Quick Links** → **Symphony Workload** → **Workload** → **Run Executable** from the Platform Symphony GUI, then enter the command with required arguments in the Remote Executable Command field as shown in Figure 3-8.

http://129.40.126.75:18080/soamgui/submitWorkload.do

Run Executable

Use this form to submit a remote command to one of the applications listed. To submit multiple, concurrent commands, open additional forms.

▼ **Session Parameters**

Remote Executable Command

Pre-exec Command

Post-exec Command

Application
 symping6.1 ▼

► **Environment Variables**

Submit **Close**

Figure 3-8 Task execution

The execution service implements the process execution logic as a Platform Symphony service and interacts directly with the service instance manager (SIM). It is used to start the remote execution tasks and return results to the client.

When the command that you submit is processed by the system, the task can get a relevant execution session ID created by the client application. The client application then sends the execution tasks to the execution service. After receiving the input message, the execution service creates a new process based on the execution task data. When the execution task is completed, the exit code of the process is sent back to the client in the execution task status.

If the command execution is successful, a control code message is displayed. If the command execution is not successful, an exception message with a brief description is returned by the system. The interface can also provide the entry of associated pre- and post- commands, and environment variables, and pass their values to the execution service.

3.4 Advanced resource sharing

In a multi-tenant environment, Platform Symphony can provide enterprise level resource management, including enterprise sharing and ownership of these resources. Different lines of business (LOBs), implemented as consumers, can share resources. When one LOB does not need its resources, it can lend them out and others can borrow them. In this way, all LOBs of an enterprise can share and use computing resources efficiently and effectively based on resource plans. Symphony also allows flexible configurations of resource plans.

In a Technical Computing cloud environment, you can define Platform Symphony shared resources in the resource distribution plan. In Figure 3-9, a client works through the SSM to request n slots from the resource manager. Based on the values specified in the resource distribution plan, the service resource manager returns m available slots on the hosts.

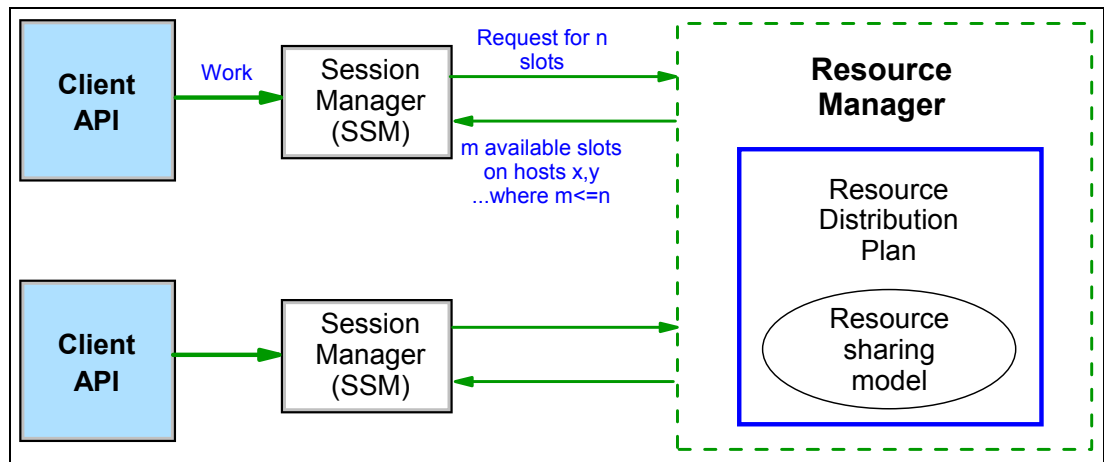


Figure 3-9 Platform Symphony advanced resource sharing

A resource distribution policy defines how many resources an application can use. Resource distribution is a set of rules that defines a behavior for scheduling or resource distribution. Each application has its set of rules. So the resource distribution plan is a collection of resource distribution policies that describes how Platform Symphony assigns resources to satisfy workloads demand. Several resource distribution policies exist. For more information, see *Platform Symphony Foundations - Platform Symphony Version 6 Release 1.0.1*, SC27-5065-01.

The ability of Platform Symphony to lend and borrow resources from one application to another ensures users access to resources when needed. Both lending and borrowing are introduced in the next sections. These operations can happen at the levels of the ownership pool (resources that are entitled to particular consumers) and the sharing pool (resources that are not entitled to any particular consumer, and thus comprise a shared resource pool).

3.4.1 Lending

If a consumer does not need all the processors that it has, it can lend excess processors. For example, if it owns 10 processors but it does not need all of them, it can lend some of them away. This is called *ownership lending* because the consumer lends away what it owns. However, if a consumer is entitled processors from the sharing pool and it does not need all of them, it can also lend them away. This operation is called *share lending* to distinguish it from ownership lending. Basically, consumers can lend processors that they do not need at a specific moment.

3.4.2 Borrowing

If a consumer does not have enough processors and borrows some that belong to other consumers (ownership pool), this is called *ownership borrowing*. Alternatively, if a consumer gets more processors than its share from the sharing pool, this is called *share borrowing*. If a consumer needs processors and there are available ones in both the sharing pool and ownership pool, the order of borrowing is to first borrow from the sharing pool, and then borrow from the ownership pool.

3.4.3 Resource sharing models

Resources are distributed in the cluster as defined in the resource distribution plan, which can implement one or more resource sharing models. There are three resource sharing models:

- ▶ Siloed model
- ▶ Directed share model
- ▶ Brokered share or utility model

Siloed model

The siloed model ensures resource availability to all consumers. Consumers do not share resources, nor are the cluster resources pooled. Each application brings its designated resources to the cluster, and continues to use them exclusively.

Figure 3-10 exemplifies this model. It shows a cluster with 1000 slots available, where application *A* has exclusive use of 150 slots, and application *B* has exclusive use of 850 slots.

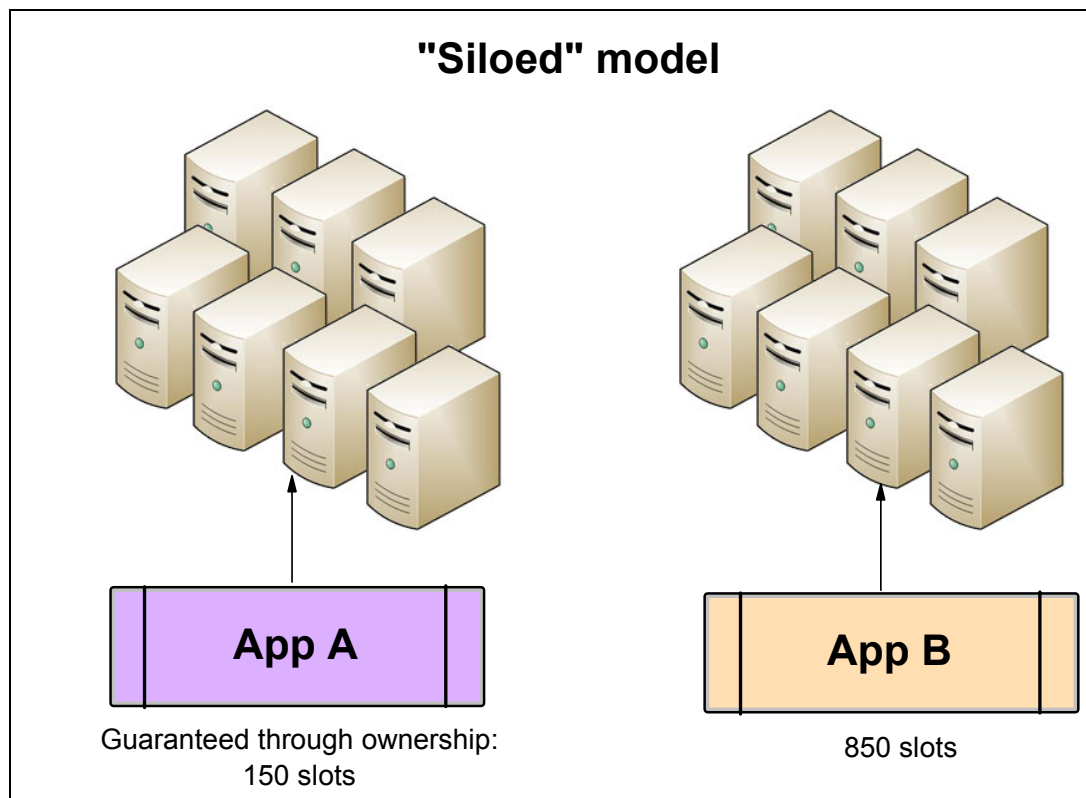


Figure 3-10 Symphony resource sharing: Siloed model

Directed share model

The directed share model is based on the siloed model: Consumers own a specified number of resources, and are still guaranteed that number when they have demand. However, the directed share model allows a consumer to lend its unused resources to sibling consumers when their demand exceeds their owned slots.

Figure 3-11 exemplifies this model. It shows that applications *A* and *B* each owns 500 slots. If application *A* is not using all of its slots, and application *B* requires more than its owned slots, application *B* can borrow a limited number of slots from application *A*.

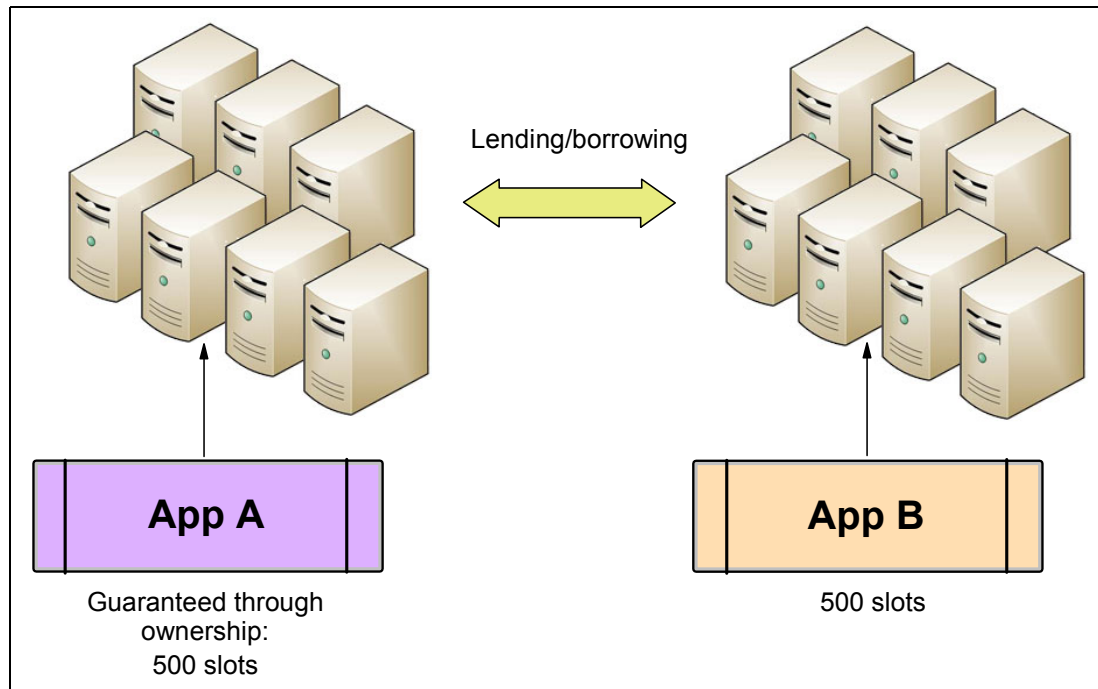


Figure 3-11 Symphony resource sharing: Directed share model

Brokered share or utility model

The brokered share or utility model is based entirely on sharing of the cluster resources. Each consumer is assigned a proportional quantity of the processor slots in the cluster. The proportion is specified as a ratio.

Figure 3-12 is an example of the brokered share model. It shows that application *A* is guaranteed two of every five slots, and application *B* is guaranteed three. Slots are only allocated when a demand exists. If application *A* has no demand, application *B* can use all slots until application *A* requires some.

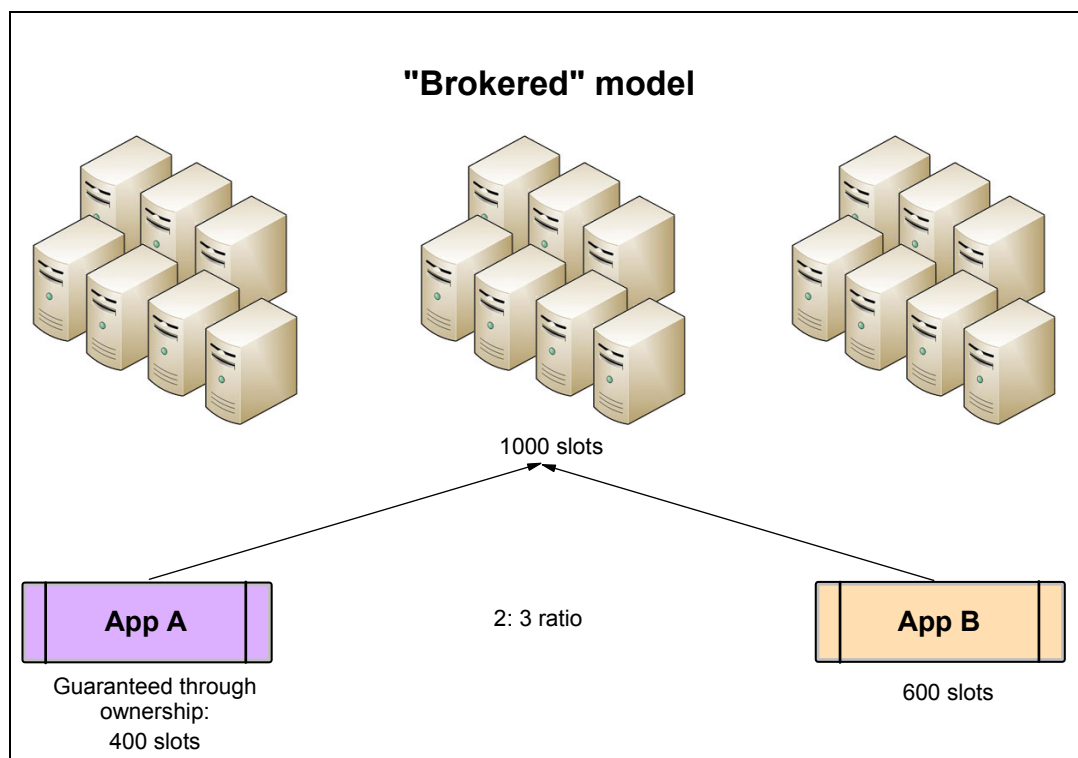


Figure 3-12 Symphony resource sharing: Brokered share or utility model

3.4.4 Heterogeneous environment support

Platform Symphony supports management of nodes running multiple operating systems, such as Linux, Windows, and Solaris. Nodes with these operating systems can exist within the same grid.

Platform Symphony clients and services can be implemented on different operating system environments, languages, and frameworks. Clusters can also be composed of nodes that run multiple operating systems. For example, 32- and 64-bit Linux hosts can be mixed running different Linux distributions, and multiple Microsoft Windows operating systems can be deployed as well. Platform Symphony can manage all these different types of hosts in the same cluster, and control which application services run on each host.

Also, application services that run on top of Linux, Windows, and Solaris can use the same service package and the same consumer. For more information, see Figure 3-6 on page 35. From a hardware perspective, Platform Symphony can be used with multiple hardware platforms.

Table 3-1 lists the hardware, operating systems, languages, and applications supported by Platform Symphony.

Table 3-1 Supported environments and applications in Platform Symphony

Infrastructure hardware and software support	
Hardware support	<ul style="list-style-type: none"> ▶ IBM System x iDataPlex® and other rack-based servers, as well as non-IBM x86 and x64 servers ▶ IBM Power Systems^a
Operating system support	<ul style="list-style-type: none"> ▶ Microsoft Windows 2003, 2003 R2 64-bit, 2008, 2008 R2 64-bit, Vista ▶ Windows 7, Windows HPC Server 2008 ▶ RHEL 4, 5, and 6 ▶ SLES 9, 10, and 11 ▶ PowerLinux supported distributions (RHEL and SLES)
Application support	
Tested applications	<ul style="list-style-type: none"> ▶ IBM GPFS 3.4 ▶ IBM BigInsights™ 1.3, 1.4 and 2.0 ▶ Appistry CloudIQ storage ▶ Datameer Analytics solution ▶ Open source Hadoop applications, including Pig, Mahout, Nutch, HBase, Oozie, Zookeeper, Hive, Pipes, Jaql
Third-party applications that are known to work with Platform Symphony	Murex, Microsoft Excel, Sungard Front Arena, Adaptiv, IBM Algorithmics® Algo® Risk, Oracle Coherence, Milliman Hedge, Alfa, Polysis, Fermat, Numeric, Calypso, Mathworks MATLAB, Quantifico, Tillinghast MoSes, Sophis Risque, Misys, GGY Axis, Openlink, Kondor+
Application and data integration	
Available APIs	<ul style="list-style-type: none"> ▶ C++, C# ▶ .NET ▶ Java ▶ Excel COM ▶ Native binaries

a. Running PowerLinux

3.4.5 Multi-tenancy

Platform Symphony can manage heterogeneous environments, both in terms of hardware and operating systems, and can share grid resources using scheduling algorithms that can provide low-latency and service levels. These characteristics make it a perfect match for being a multi-tenant middleware.

Note: Platform Symphony is a multi-tenant shared services platform with unique resource sharing capabilities.

Platform Symphony provides these capabilities:

- Share the grid among compute and data intensive workloads simultaneously
- Manage UNIX and Windows based applications simultaneously
- Manage different hardware models, simultaneously

Figure 3-13 demonstrates how powerful it is when it comes to sharing grid resources to multiple applications.

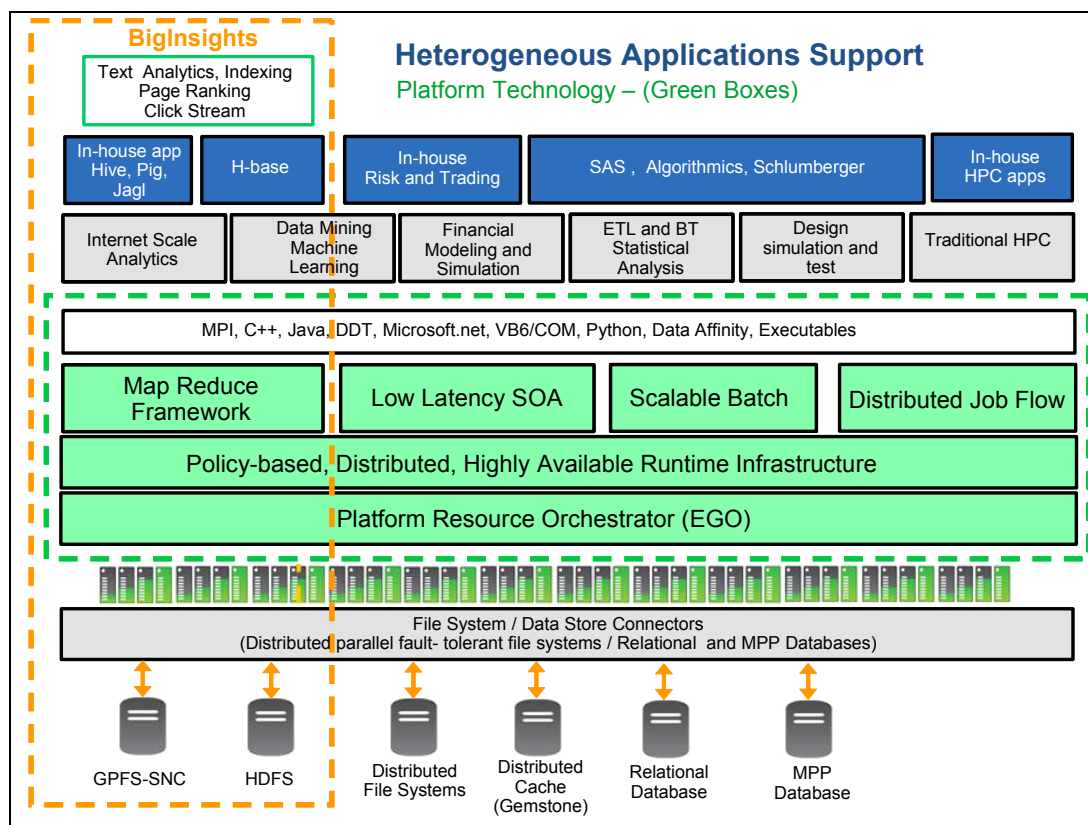


Figure 3-13 Multi-tenant support with Platform Symphony

3.4.6 Resources explained

Resources on the grid are divided into flexible resource groups. Resource groups can be composed of the following components:

- Systems that are owned by particular departments.
- Systems that have particular capabilities. For example, systems that have lots of disk spindles or graphics processing units installed.
- Heterogeneous systems. For example, some hosts run Windows operating system while others run Linux.

As explained in Figure 3-6 on page 35, a *consumer* is something that consumes resources from the grid. A consumer might be a department, a user, or an application. Consumers can also be expressed in hierarchies that are called *consumer trees*.

Each node in a consumer tree owns a share of the grid in terms of resource slots from the resource groups. Shares can change with time. Consumers can define how many slots they are willing to loan to others when not in use and how many they are willing to borrow from others, thus characterizing a resource sharing behavior as explained in 3.4.3, “Resource sharing models” on page 43.

Note: Owners of slots can be ranked.

Applications are associated with each of these consumers, and application definitions provide even more configurability in terms of resources that an application needs to run.

With all of this granularity of resource sharing configuration, organizations can protect their SLAs and the notion of resource ownership. Users can actually get more capacity than they own. Grids generally run at 100% utilization because usage can expand dynamically to use all available capacity.

3.5 Dynamic growth and shrinking

When the need for resources grows and shrinks with processing loads, middleware software that is intended to control these resources must be able to address dynamic changes to grid topology. This avoids under-utilization of resources during low processing periods, and allows for temporary resource assignment to meet peak processing demands.

New nodes can be added to a Platform Symphony-managed grid dynamically without interrupting services. This, however, characterizes a definitive topology change to the grid itself. That is, the added node is now part of the grid, unless it is removed by the grid administrators. Subtracting nodes from a grid follows the same concept. This mechanism allows Platform Symphony to dynamically grow and shrink its grid, a characteristic that is in accordance to a cloud environment, which is also dynamic.

Platform Symphony, however, offers an extra type of dynamism when it comes to adding temporary capacity to its grid. Imagine that you exhausted your grid capacity during a peak processing time and are clearly in need of more resources. Purchasing more hardware to meet this demand is expensive and cannot be done in a timely manner. Instead, you want to make use of idle resources that you already own outside of that grid. With Platform Symphony, you can.

The following mechanisms can be used by Platform Symphony to attend to a peak demand by using existing resources within your environment, or by borrowing extra capacity from a provider:

- ▶ Desktop and server scavenging
- ▶ Virtual server harvesting
- ▶ On-demand HPC capacity

3.5.1 Desktop and server scavenging

Platform Symphony is able to scavenge idle servers and desktops that are not part of its grid. This is a dynamic operation in which these resources, which are not part of the permanent grid, can be requested to process a grid workload. Desktops might be common workstations that are used by office people in its daily tasks, and servers can be web servers, mail servers, file servers, and so on.

The scavenging of extra servers and desktops for grid processing does not compete against the usual jobs that are run on these systems. That is, if a user is actually using its desktop to perform a task, Platform Symphony does not send workloads to it. Similarly, when a web server is actually under load servicing HTTP requests, Platform Symphony does not send workloads to it. When these desktops and servers are busy, they are said to be closed to the Platform Symphony grid and cannot receive grid workloads. After they become idle, their state changes to open for accepting grid workloads. Servers can be configured with a custom load threshold under which it becomes open to grid processing.

Figure 3-14 shows a diagram for Platform Symphony desktop and server scavenging.

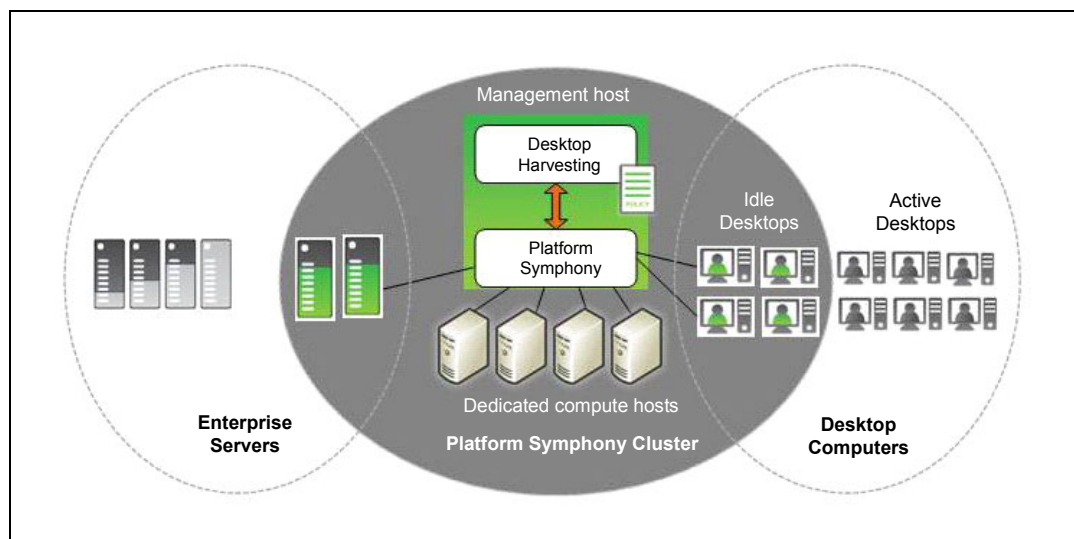


Figure 3-14 Symphony desktop and server scavenging

Desktop and server scavenging has the following advantages:

- ▶ Uses idle capacity of existing resources within your company or institution.
- ▶ Scavenging does not compete for resources when desktops and servers are not idle (open stated or closed state for grid use). Therefore, there is no impact to users or applications that usually run on these desktops and servers.
- ▶ Improves the performance of your Platform Symphony grid.
- ▶ Reduces costs by using existing resources, avoiding new capital expenditure.

3.5.2 Virtual server harvesting

Similarly to desktop and server scavenging, Platform Symphony is also able to harvest virtual servers into the grid. Again, this happens as a temporary resource addition to the grid.

Virtual servers can receive grid jobs when its usual processing is below a custom threshold, or when the virtual server is idle only. You might have different virtual environments as depicted in Figure 3-15. Also, you can either choose to create more virtual servers managed by their original clusters or pools.

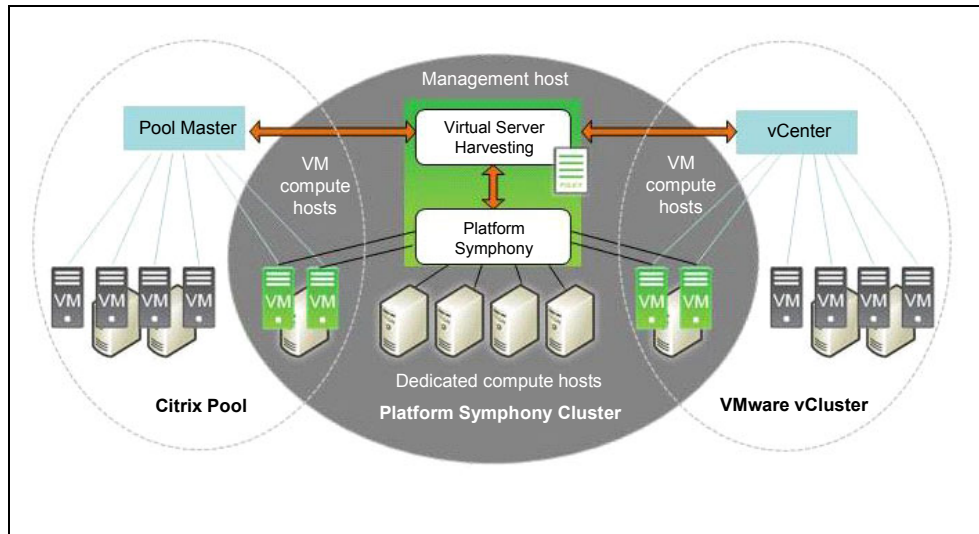


Figure 3-15 Symphony virtual server harvesting

Virtual server harvesting provides these benefits:

- ▶ Gives the grid access to idle cycles from virtual server infrastructures.
- ▶ Harvesting causes minimal impacts to existing virtual machine environments.
- ▶ Avoids costly duplication of infrastructure.
- ▶ Increases grid capacity without capital expenditure.
- ▶ Improves grid service levels.

3.5.3 On-demand HPC capacity

If your HPC cluster peaks at a particular time, Platform Symphony allows you to use external HPC resources from providers in an on-demand fashion. You can think of on-demand HPC capacity as the act of instantiating and connecting extra resources to your existing infrastructure to serve your peak demands. This concept is also called *cloud bursting*.

You can have your infrastructure ready to connect to a cloud resource provider at all times, but set up a policy to use this on-demand extra capacity only when a certain threshold processing level is reached. This can be automatically managed by a threshold policy.

Figure 3-16 depicts the idea of on-demand HPC capacity.

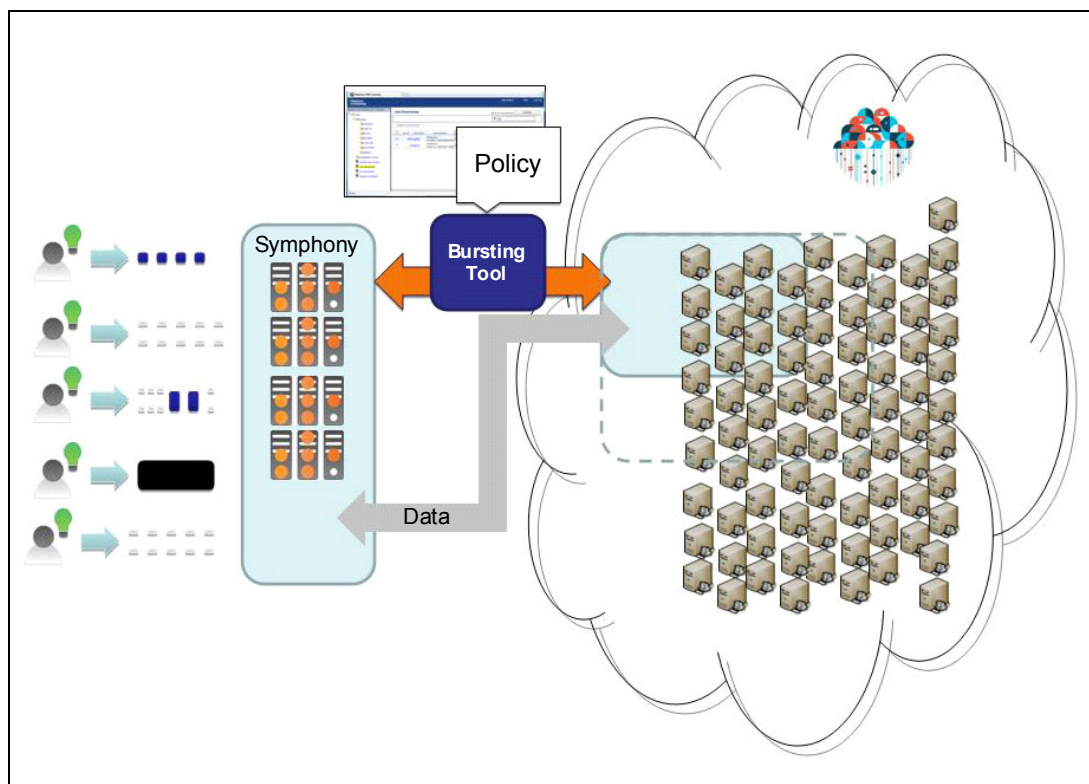


Figure 3-16 Platform Symphony using on-demand HPC capacity

Platform Symphony has plug-ins that allow an easy integration of your internal grid with external HPC capacity. Moreover, no changes are required to the job submitting processes. After you burst your grid with extra cloud resources, Platform Symphony is able to use them transparently.

On-demand HPC capacity also provides the following advantages:

- ▶ Minimize capital expenses: No need to acquire extra hardware for short peak demands.
- ▶ Ensure service level agreements: Jobs can be serviced on time by quickly using extra grid capacity.
- ▶ Scale endlessly: Ability to scale as much resources as needed with cloud burst.
- ▶ Capacity planning becomes less critical.
- ▶ Existing infrastructure can be reshaped while still meeting processing demands.
- ▶ Capital expenses become operational expenses.

Figure 3-17 shows a grid workload behavior that can make good use of on-demand HPC resources.

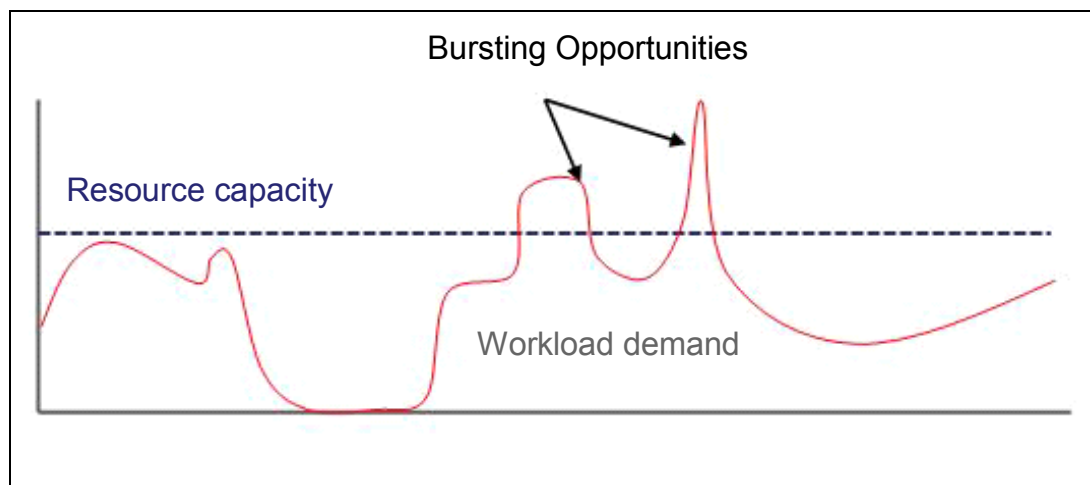


Figure 3-17 Workload opportunities for bursting existing grid capacity

Using external resources for a grid offers some challenges because the extra resources are not local to your infrastructure:

- ▶ Security
 - Do the providers offer data protection?
 - Your network IP addresses on third-party infrastructure are outside of your firewall.
- ▶ Application licenses
 - Legal agreements might limit technology use to certain geographic locations only or to corporate sites, preventing a more free choice of where you can burst your environment to.
- ▶ Performance
 - Applications must be able to efficiently use general, multi-purpose external resources.
- ▶ Data movement
 - As depicted in Figure 3-16 on page 51, you must exchange data between your site and the on-demand provider's site.

Even after the challenges mentioned, using on-demand HPC capacity is a reality with Platform Symphony.

3.6 Data management

Workload schedulers focus on dispatching tasks to compute hosts and transferring data either directly to the compute hosts or delegating data retrieval to the service. The time that is required for this data transfer from various sources to where the work is being processed can lead to inefficient use of processor cycles and underutilization of the resource.

Platform Symphony has a mechanism called data-aware scheduling, or data affinity, to optimize the scheduling of jobs to nodes where data is found to be local. It is a feature available in IBM Platform Symphony Advanced Edition.

3.6.1 Data-aware scheduling

The data-aware scheduling (data affinity) feature allows Platform Symphony to intelligently schedule application tasks and improve performance by taking into account data location when dispatching tasks. By directing tasks to resources that already contain the required data, application run times can be significantly reduced. In addition, this feature can help to meet the challenges of latency requirements for real-time applications.

Note: Data-aware scheduling is a feature available in the Advanced Edition version of Platform Symphony.

With the data-aware scheduling feature, you can specify a preferential association between a task and a service instance or host that already possesses the data that is required to process the workload. This association is based on the evaluation of a user-defined expression that contains data attributes capable of being collected. The evaluation of this expression is carried out against the data attributes of each service instance available to the session. Typically, a data attribute is an identifier for a data set that is already available to a service instance before it processes the workload.

Figure 3-18 illustrates the concept of data-aware scheduling at the task level. The data preference expression is evaluated and it is determined that a task in the queue prefers to run on a service instance where the service instance already possesses *Dataset1*. The SSM collects metadata (service attributes) from all the resources available to the session at that moment. Service B with *Dataset1* is available and, because it is the best match for that task according to the specified preference, the task is then dispatched to Service B.

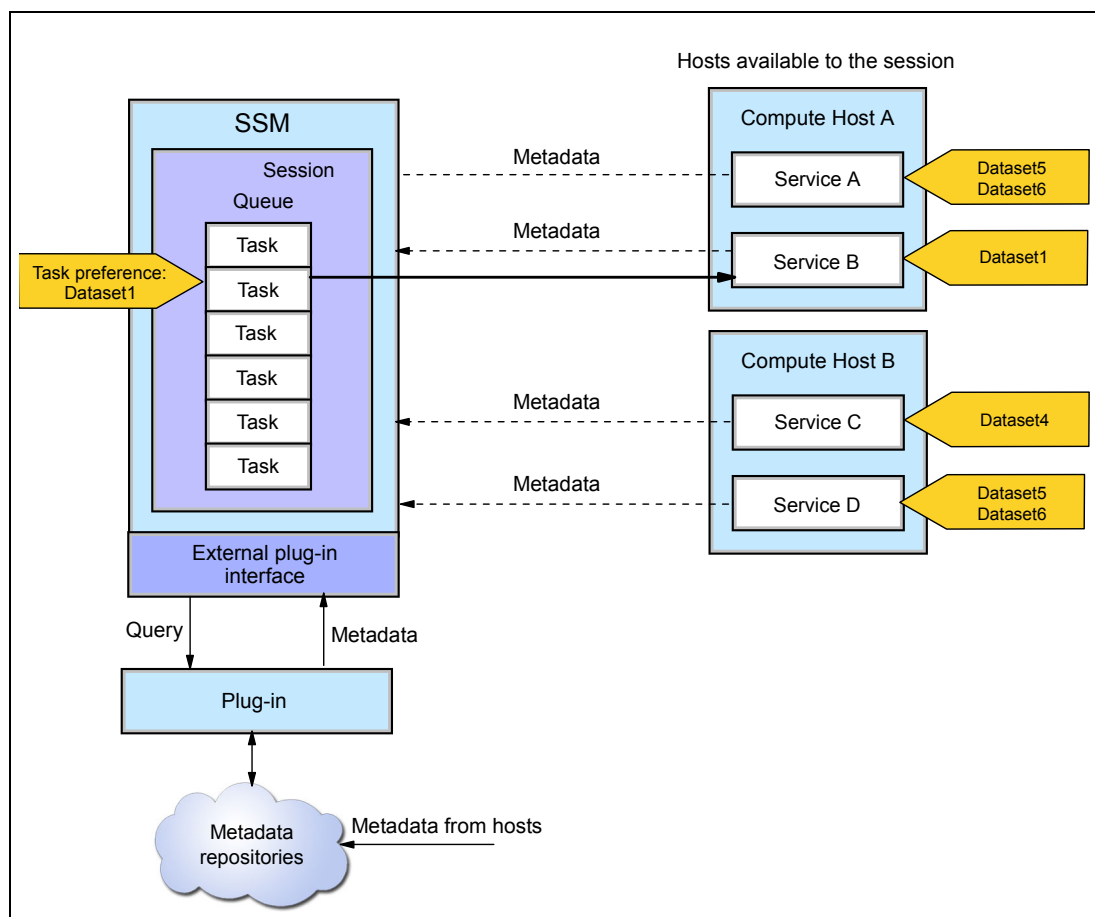


Figure 3-18 Data-aware scheduling

The following is a list of benefits of the data-aware scheduling provided by Platform Symphony:

- ▶ Elimination of data bottlenecks by intelligently placing workload on systems that are physically close to the required data.
 - More flexible scaling of data-intensive applications with improved performance.
 - Reduced application run times.
 - Improved utilization through deployment of applications on enterprise grids that previously were difficult to run in a shared environment.
 - Reduced reliance on expensive network storage hardware.
- ▶ Service level data publishing.
- ▶ Use of previously calculated results to speed up subsequent calculations.
- ▶ Tasks that are matched with data.
- ▶ Interfacing with data caching and distributed file system solutions.

3.7 Advantages of Platform Symphony

Platform Symphony is an enterprise class scheduler for HPC, technical computing, and cloud-computing environments.

In addition, there are other advantages of Platform Symphony:

- ▶ Get higher quality results faster:
 - Starts and runs jobs fast
 - Scales very high
- ▶ Lower costs:
 - By increasing resource utilization levels
 - Easier to manage
 - Simplifies application integration
- ▶ Better resource sharing:
 - Both compute and data intensive workloads can use the same resources
 - Provides a sophisticated hierarchical sharing model
 - Multi-tenancy model for applications
 - Provides harvesting and multi-site sharing options
- ▶ Smarter data handling:
 - Through an optimized and low-latency MapReduce implementation
 - Consideration of data locality when scheduling tasks
 - Adapts to multiple data sources

This chapter focuses on the product characteristics that are related to cloud and grid environments. The following advantages of Platform Symphony are as detailed in this publication:

- ▶ Advanced monitoring
 - Node status, grid component status, application service failures, consumer host under allocated, client communication status, session status (aborted, paused, resumed, priority change), and failed tasks.
- ▶ Full reporting
 - Symphony provides nine built-in standard reports based on the following information: Consumer resource allocation, consumer resource list, consumer demand, orchestrator allocation events, resource attributes, resource metrics, session attributes, task attributes, session property, and session history. Also, users can create their own custom reports.
- ▶ Analytics and metrics
- ▶ Full high availability and resiliency at all levels
 - Client, middleware, schedulers, and service instances.

For more information about Platform Symphony's software components and architecture, and its monitoring, reporting, metrics, and high availability features, see *IBM Platform Computing Solutions*, SG24-8073.

3.7.1 Advantages of Platform Symphony in Technical Computing Cloud

Platform Symphony is able to manage grid resources for multiple applications that are running within a grid. However, let us know move one step further. It can also host a grid with multiple applications for multiple customers, and provide requires a complete isolation of its grid environment for each customer from the other tenants (other customers).

You can use the IBM Platform Cluster Manager Advanced Edition (PCM-AE) to deploy separate grid environments to serve distinct tenants, isolating one from another. For more information, see 5.2, “Platform Cluster Manager - Advanced Edition capabilities and benefits” on page 90. Each cluster then has its own Platform Symphony scheduler that manages the resources of that particular cluster grid only. Platform Symphony can be integrated with PCM-AE as Platform Symphony’s installation and setup can be managed by a post-installation script during PCM-AE cluster deployment.

Also, cluster clouds must be able to dynamically reprovision resources to technical computing cluster environments. This can be accomplished with PCM-AE as described in 5.2, “Platform Cluster Manager - Advanced Edition capabilities and benefits” on page 90, and is called cluster flexing. With that, you can dynamically grow or shrink your cluster. Whichever middleware is used within a cluster to control the use of its resources must also be capable of following PCM-AE’s provisioning and reprovisioning operations. Platform Symphony is flexible to dynamically add or remove nodes from its grids.

In summary, Platform Symphony is not only able to manage a grid of computing resources, but is also ready to use and deploy inside a higher level of resource organization: A cloud of clusters.

3.7.2 Multi-core optimizer

The multi-core optimizer is an IBM Platform Symphony add-on product that can make most out of multi-core servers within Technical Computing cloud environments.

What can the multi-core optimizer do in a cloud multi-core environment? It cannot only reduce capital and operating expenses through running multiple I/O-intensive tasks per core. It must also efficiently matching resources with non-uniform workloads on non-uniform hardware. Moreover, it improves the performance and scalability of applications by reducing I/O and memory contention in cloud multi-core environments. This can help improve the performance and scalability of applications for these reasons:

- Optimize utilization in mixed environments. The multi-core optimizer dynamically maps services onto slots, and also allows applications to specify a service to slot ratio. Therefore, you can share heterogeneous environments among any mix of application architectures. This includes single-threaded, multi-threaded, data-intensive, I/O-intensive, compute-intensive, and so on. For example, an application that runs eight threads can be assigned eight slots.

- ▶ Reduce data and memory contention. Sometimes, the same data must be sent multiple times to a host, which places a burden on host memory and network bandwidth. This causes greater I/O contention with multiple cores on a processor, and is not good for cloud multi-core environments. The multi-core optimizer extends the common data optimization feature in Platform Symphony to optimize the distribution of common data and common data updates so that only one copy of the data is sent to each multi-core host serving the same session. This process reduces the burden on memory and network bandwidth.
- ▶ Oversubscribe slots to improve utilization. The multi-core optimizer improves overall cluster utilization by enabling intelligent over-scheduling of low priority work to available cores. It preempts the lower priority tasks currently running when a high priority task comes in to ensure that high priority tasks can reclaim the cores as needed so SLAs are met.



IBM Platform Symphony MapReduce

This chapter describes Platform Symphony MapReduce and an outline of its benefits while running data-intensive Hadoop MapReduce workloads inside the Platform Symphony environment.

This chapter included the following sections:

- ▶ Overview
- ▶ Key advantages for Platform Symphony MapReduce
- ▶ Key benefits

4.1 Overview

This section introduces the MapReduce technology and the Hadoop architecture, and describes the IBM Platform Symphony MapReduce framework. This framework allows you to run data-intensive Hadoop MapReduce workloads inside the Platform Symphony environment. It works at high levels of performance and shared resource utilization, in a secure multi-tenant fashion, through a nice on-demand self-service web interface. This cloud-computing-specific function is provided by two sophisticated components: A scheduling engine and a resource manager that is used by an advanced management console.

4.1.1 MapReduce technology

A Hadoop MapReduce computing environment is built on top of two core components:

- ▶ Hadoop Distributed File System (HDFS) for data storage
- ▶ Hadoop MapReduce for data processing.

These two components provide affordable but performance distributed processing of large amounts of data on physical infrastructure made of low-cost commodity hardware. Such a grouping of nodes in the same data center or site and contributing together to the same computing purpose is usually referred to as a *cluster*. Because the infrastructure can scale out to thousands of nodes, the risk of hardware failures is high. The design addresses this situation by including automatic detection and recovery during hardware failures.

The Hadoop Distributed File System (HDFS)

HDFS is a specialized distributed file system that is optimized for files in the range between hundreds of megabytes and gigabytes, or even terabytes in size. It stores the data files on local disks in a specific format by using large file system blocks that are typically 64 MB or 128 MB. The goal behind this format is the optimization of the data processing while running the MapReduce algorithm. Consider, for example, a 10 GB file in an HDFS file system that has a 128 MB block size and is uniformly spread over 100 nodes. This file might be processed at today's usual sequential sustained read speed of 70-80 MBps in less than 2 seconds (assuming fast enough processors, so no CPU-bound workload). For a specific application with a given average input file size and a wanted latency, you can estimate the required number of nodes for your cluster.

The data redundancy results from replication of these blocks on different nodes. The usual replication factor is three, and the nodes are chosen in such a way that not all three of them are in the same rack. HDFS implements a rack awareness feature for this purpose.

For more information about the architecture and execution environment of an HDFS file system, see 4.1.2, "Hadoop architecture" on page 62.

MapReduce algorithm

MapReduce implements a distributed algorithm that consists of separate fragments of work, each one running on a node in the cluster and processing a slice of data, preferably on that node, at the HDFS level. The processing involves phases, the most representative of which are the map phase and the reduce phase. Each of these phases consists of multiple parallel and similar fragments of work, called map tasks and reduce tasks.

Figure 4-1 shows an example of a MapReduce data flow that starts by reading its input from the HDFS and ends by writing the output results to the same file system.

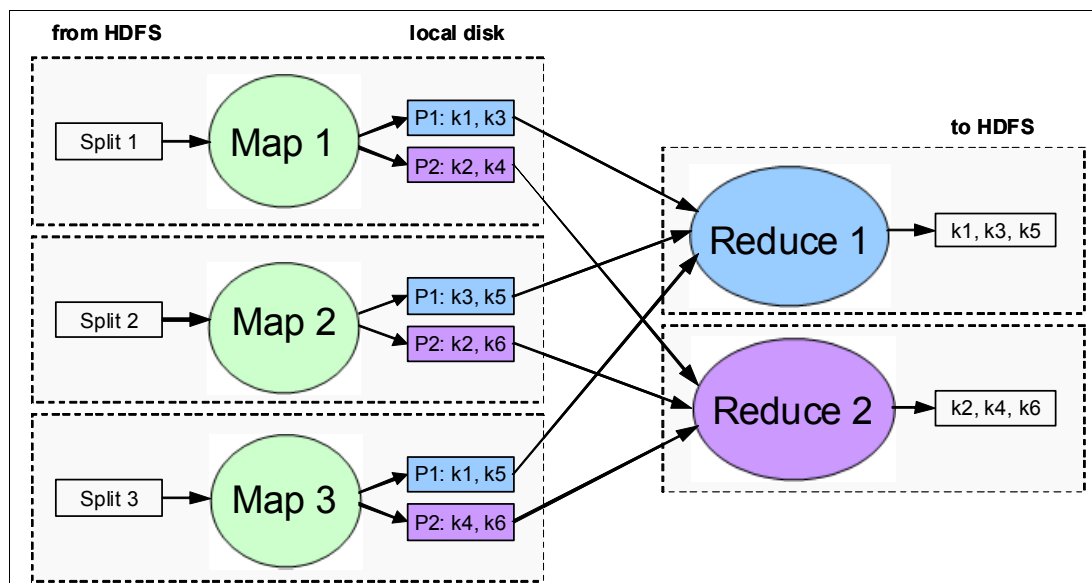


Figure 4-1 MapReduce data flow

The data in one input block (input split or just split in MapReduce terminology) does not depend on the data in any other block. Map tasks are instantiated on each node containing an input split, normally one task for each split. There might not be a perfect overlap between an input split and an HDFS block, but the framework does its best to match them. The small portion of the split that might be on a different node is transferred locally when needed.

A map function parses the records in the split and outputs an intermediate key-value pair for each processed record. The developer must incorporate the logic for the map function, considering the record format and the processing purpose. The output of the map function is written to the local file system (not to HDFS) for performance reasons.

The output of each map is partitioned to determine which part goes to each reduce task. By default, the framework hashes the keys to select the destination reduce task. All the intermediate pairs that have a particular key are sent to the single reduce task to which the hash function has associated the respective key.

Local aggregation of the intermediate outputs might also be specified by the user in a *combiner operation*. This minimizes the network traffic in the subsequent shuffle stage.

The number of reduce tasks is determined by the Hadoop framework. The series of intermediate key-value pairs can now be sent to the reduce tasks. This constitutes the shuffle stage.

Now that all intermediate data is partitioned and transferred to the reduce function, an extra merge operation is run by the framework before the data enters the final reduce tasks. As key-value pairs with the same key might have been delivered from different maps, this merge operation regroups and resorts them. A secondary sort by value might be specified here, before the final reduction, with a so-called comparator function.

The reduce tasks consolidate results for every key according to a user implemented logic, and these results are finally written to the HDFS.

4.1.2 Hadoop architecture

In this section, the two core components of the Hadoop MapReduce environment are put together in a functional architecture diagram. It also mentions some higher-level applications that make the environment truly usable in concrete situations.

Hadoop MapReduce execution environment

The execution environment for the involved tasks is presented in Figure 4-2. It introduces the entities that are involved and their operation in the workflow. A MapReduce job is a self-contained unit of work that a user wants to be performed on a set of input data, usually one or more files, which are stored in the HDFS file system.

The job is initially specified by the user, and consists of the input data, the MapReduce application, an output directory, and possibly extra non-default Hadoop framework configuration information. The job is submitted to the MapReduce framework, which is a combination of the master node and the compute nodes. The master node runs a process called JobTracker that takes the request from the user and coordinates the job execution flow process. The JobTracker schedules other processes on compute nodes called TaskTrackers. As detailed in “MapReduce algorithm” on page 60, Hadoop runs the job by dividing it into two sets of tasks: Map tasks and reduce tasks. The TaskTracker processes start, each on its own compute node, and with its associated map and reduce tasks. They monitor the local data processing tasks, and send progress reports to the JobTracker.

The JobTracker monitors the overall progress of the execution flow. If a TaskTracker fails, the JobTracker reschedules its task to a different TaskTracker. The JobTracker maintains its state on the local drive.

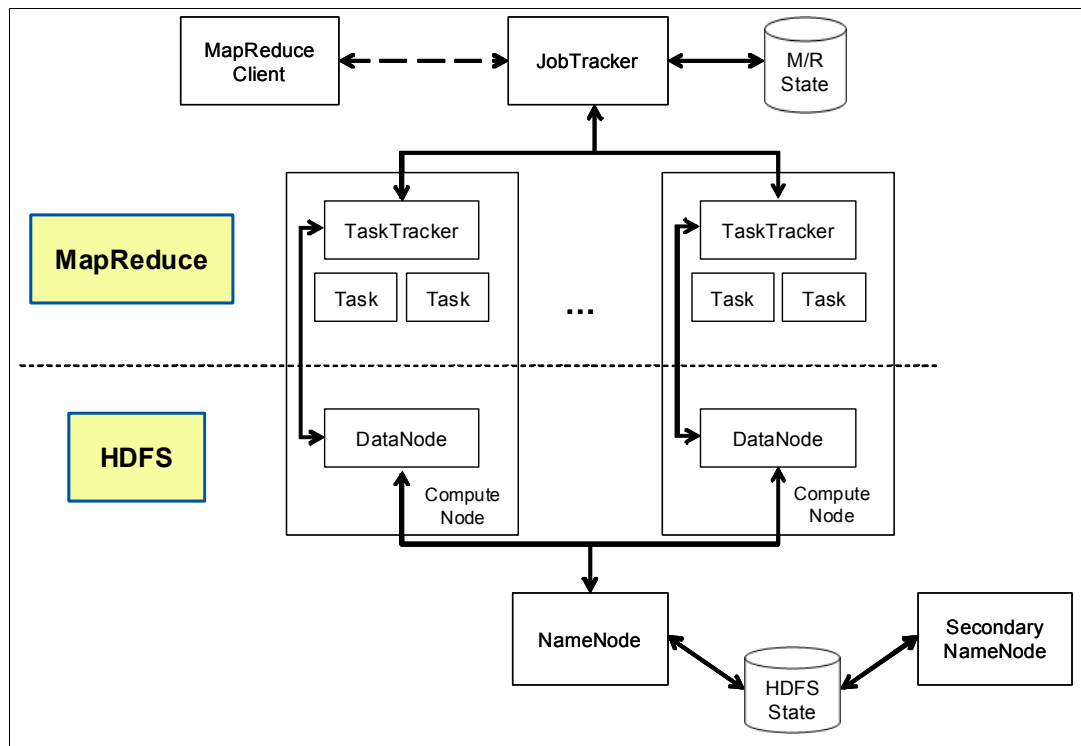


Figure 4-2 MapReduce execution environment

The Hadoop framework initially processes the arguments of a job and identifies the nodes where the HDFS blocks containing the bulk input data are. Data splits are derived from the

considered blocks on the identified nodes, and presented to the input of the map tasks. As explained in “MapReduce algorithm” on page 60, there might not be a perfect match between a MapReduce split and an HDFS data block. Limited network traffic might appear to feed the map task with the data fragments in the split that is not stored locally. The overall effect of the data locality is a significant optimization because valuable cluster network bandwidth is preserved.

This is the typical way that the framework schedules the map tasks on the chosen nodes. There might be situations when all of the three nodes that store the HDFS block replicas of an input split are already running map tasks for other splits. In this case, the JobTracker tries to allocate a node as close as possible to one of the replicas behind the input split, preferably in the same rack. If this is not possible, an available node in a different rack might be used, resulting in some inter-rack network data transfer.

In a typical Hadoop deployment, the JobTracker and NameNode processes run on the same node. If there is a lot of data or lots of files in the HDFS, separate the processes because the NameNode needs dedicated system memory. Similarly, for the SecondaryNameNode, you can typically run it on the same node as the NameNode. However, if memory requirements for the NameNode are high, it is more appropriate to choose a separate node for the SecondaryNameNode. The SecondaryNameNode does not play a direct failover role. It actually provides a service to manage HDFS state files in case recovery is needed.

High-level Hadoop applications

The MapReduce framework is written in Java programming language. Using it at the low level of the map and reducing Java class methods is a meticulous and time consuming activity. It is similar to assembly language, where you must write a lot of code to implement simple operations. However, technologies using this framework at a higher level of abstraction have been developed. The following open source tools are supported with Hadoop MapReduce in the Platform Symphony MapReduce framework: Pig, Hive, Hbase, and Zookeeper. Figure 4-3 shows them on top of the basic Hadoop MapReduce core components.

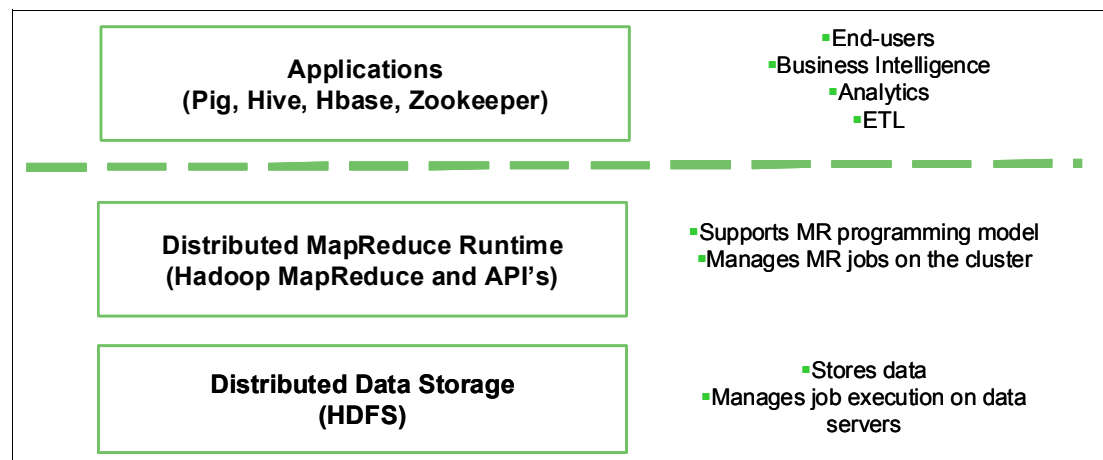


Figure 4-3 Hadoop system

4.1.3 IBM Platform Symphony MapReduce framework

The MapReduce framework is depicted in Figure 4-4 along with the other components of the Platform Symphony high-level architecture. The Platform Symphony MapReduce framework has been added as a core component and has been tightly integrated with the existing standard core components: SOA framework, scheduling engine, resource orchestrator, and management console.

Before adding the MapReduce functions, the standard Symphony product exposed its low-latency service-oriented architecture (SOA) framework to interactive compute-intensive applications. These were the kind of applications it was originally targeted to support. These are still valid and supported, but additionally, the same SOA framework can also support MapReduce data-intensive applications, which are similar in their workflow structure. The SOA framework implements this common workflow structure by running on top of the scheduling engine, which gets computing resources from the resource orchestrator. These core components are augmented to become a complete enterprise environment with the monitoring, administering, and reporting features of the management console. The core components are colored green in Figure 4-4 to make them more visible. For more information about all these components, see Chapter 3, “IBM Platform Symphony for technical cloud computing” on page 29.

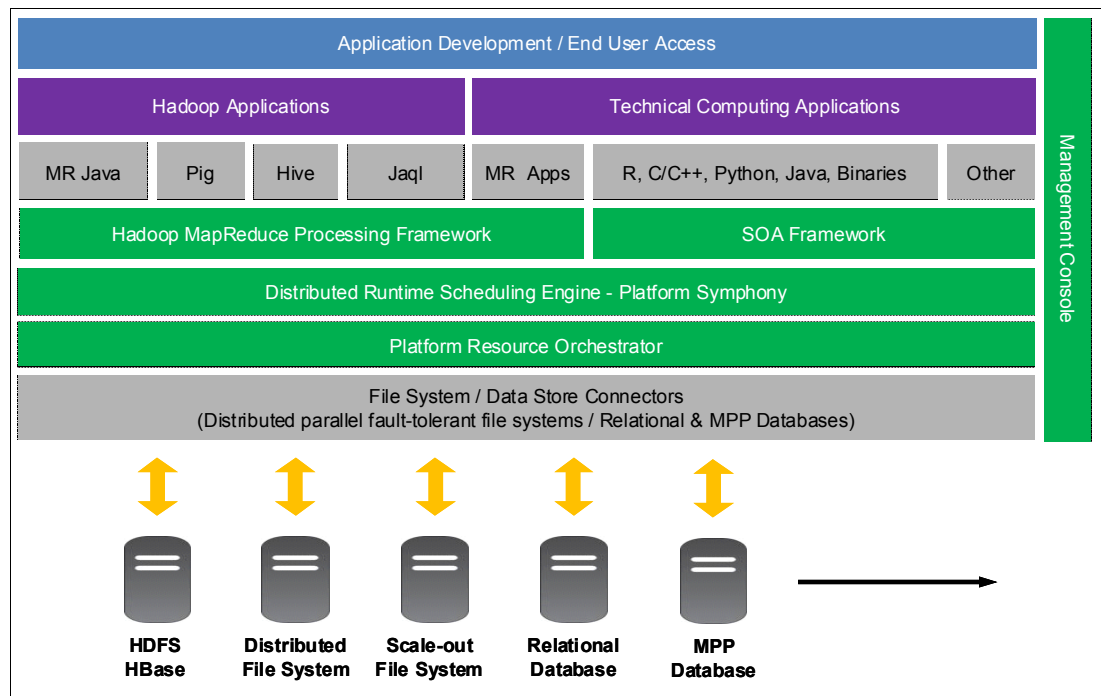


Figure 4-4 Platform Symphony MapReduce framework

With this new MapReduce layer, Platform Symphony can integrate and run external high-level Hadoop-related applications. These applications run on top of the MapReduce framework. These applications are not delivered with the Platform Symphony installation bundle, and must be installed separately. Platform Symphony supports both the Apache Hadoop open software bundle and commercial distributions that are based on it such as IBM InfoSphere® BigInsights.

At the bottom of Figure 4-4, you can also see a distinct layer of data storage connectors that are needed for the integration of various distributed file systems and other sources of data. The MapReduce data-intensive applications are going to use these systems while processing

their considerable amounts of input data. The storage connectors integrate Platform Symphony with different file and database systems such as HDFS, GPFS, Network File System (NFS), and the local file system.

Integration with the underlying SOA middleware

Figure 4-5 shows the Platform Symphony MapReduce framework at a first level of detail, revealing its consistency with the Hadoop MapReduce logic flow. The important capability here is that more application managers, which each correspond to a JobTracker, can coexist in the same infrastructure. This means that distinct MapReduce job queues can run simultaneously and share infrastructure resources, each of them with its own set of MapReduce jobs. Another aspect highlighted in Figure 4-5 is that the various components involved in the case of a data-intensive MapReduce application must implement more functionality compared to the standard SOA compute-intensive case. This is required to provide access, when and where needed, to the data in the distributed shared storage.

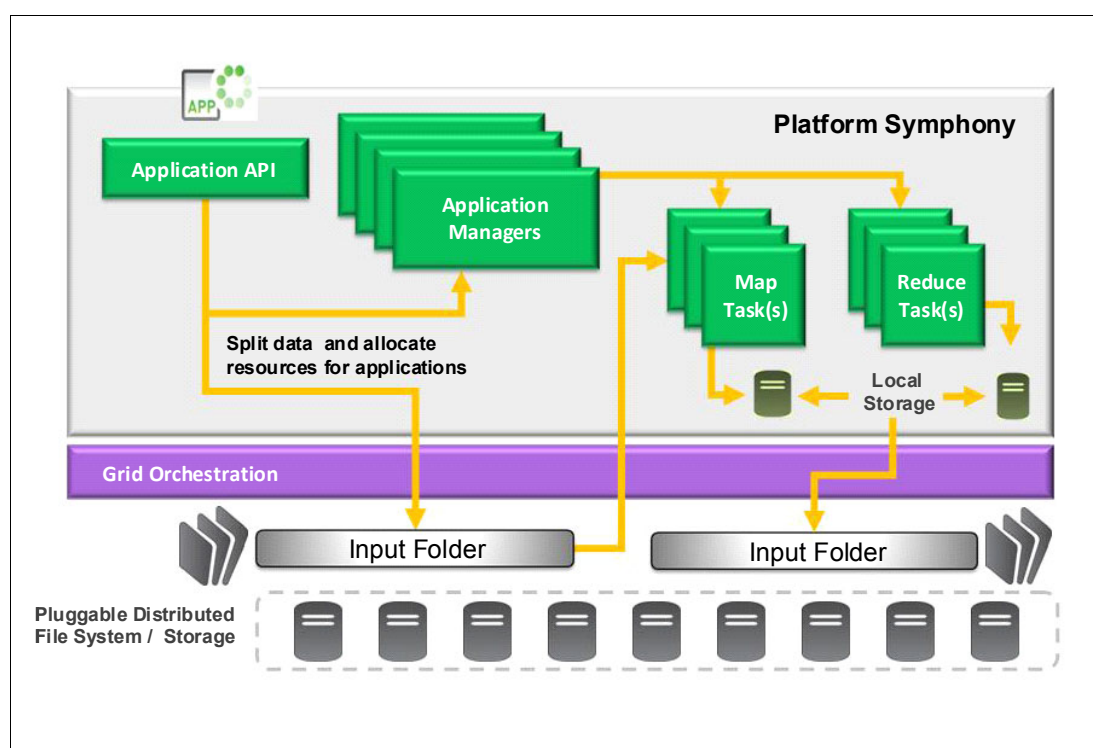


Figure 4-5 Platform Symphony MapReduce framework

Each application manager corresponds to a distinct application registered on a distinct consumer in the hierarchy of consumers that are configured on the grid. Application manager is just another term that is used inside the MapReduce framework for the service session manager (SSM) at the SOA level. The map and reduce tasks are also simple tasks at the SOA level, each served by its own instance of the MapReduce service.

There are many SOA framework terms and notions, only some of which are covered in this chapter. Figure 4-6 depicts a generic view of the SOA framework with its main components. To run an application, a user submits a job to the grid through the client component of this application. The client-side code uses the SOA API to locate the session manager of the application. For this, the API contacts a system service or session director that is aware of all the configured applications in the grid. If not already available, the session manager is instantiated by the service director. There is only one session manager per application. The client communicates from now on with the session manager.

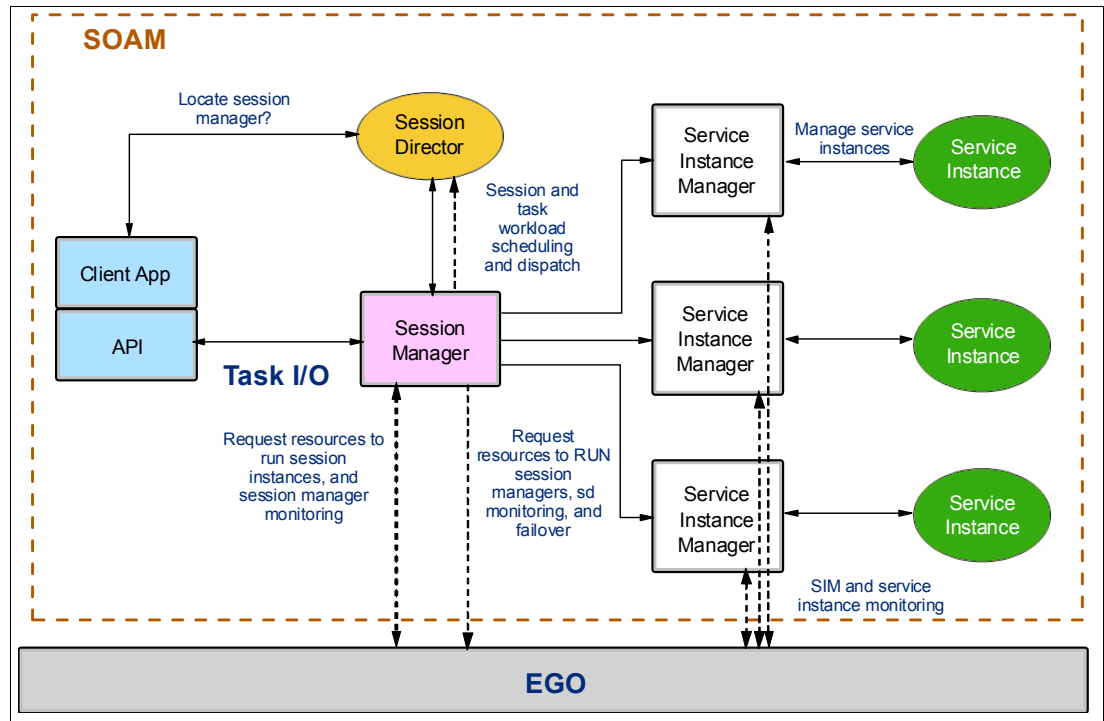


Figure 4-6 Service-oriented application middleware components

A session is created for the particular job submitted by the client. The client gets a handle for this session, and through this handle the client code requests a number of tasks to be scheduled on the grid. The objects that are created inside the SOA framework to manage a particular session are shown in Figure 4-7.

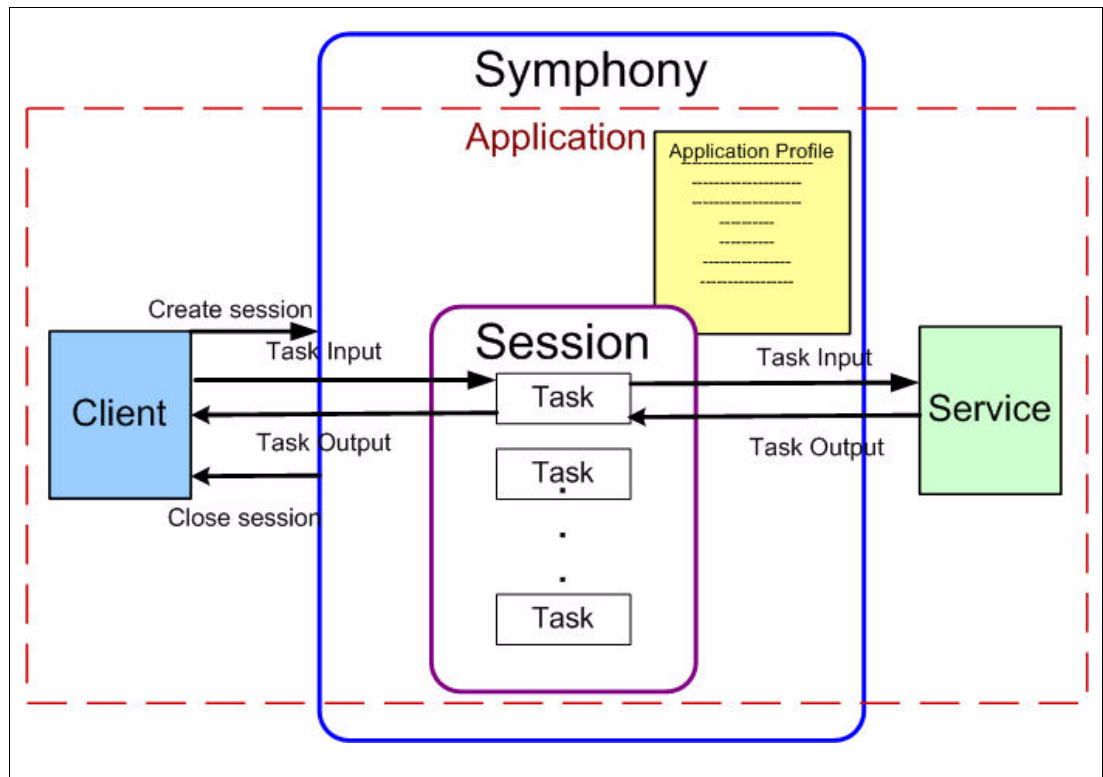


Figure 4-7 SOA application objects

The session manager transmits the scheduling request to the scheduling engine, which decides according to the resource distribution policy and other scheduling attributes associated with the application, the CPU slots, and their locations on the compute nodes, for the requested tasks. The session manager is then able to dispatch a corresponding number of session instance managers (SIM) and service instances of the service that is associated with the application involved. There is one service instance per task, that is locally monitored on the hosting compute node by an associated SIM. The session manager manages and monitors all the SIMs associated with any of its sessions. If another job of the same application is submitted in parallel, a new session is created by the same session manager, and the whole task scheduling process is repeated. Tasks in multiple sessions then compete for the resources that are allocated to the same application. A scheduling framework is available for the sessions of an application. This is a separate scheduling layer on top of the overall resource distribution layer that schedules the resources of the whole grid to the active applications registered by all the defined consumers.

Platform Symphony MapReduce work and data flow

The execution workflow of a MapReduce job in a Platform Symphony environment is shown in Figure 4-8. It has similarities and the differences compared with the Apache Hadoop approach as detailed in “Hadoop MapReduce execution environment” on page 62. Notice the SOA framework specific to Platform Symphony in which the client side of the MapReduce application interacts through a well-defined interface with the service component of the same MapReduce application and MRService deployed in the cluster. The mapping of the standard Hadoop MapReduce entities to the SOA middleware objects playing equivalent roles is also similar. The SSM is in the role of the JobTracker, while various SIMs act as TaskTrackers. Multiple instances of the Platform Symphony MapReduce service run concurrently on compute nodes in the cluster.

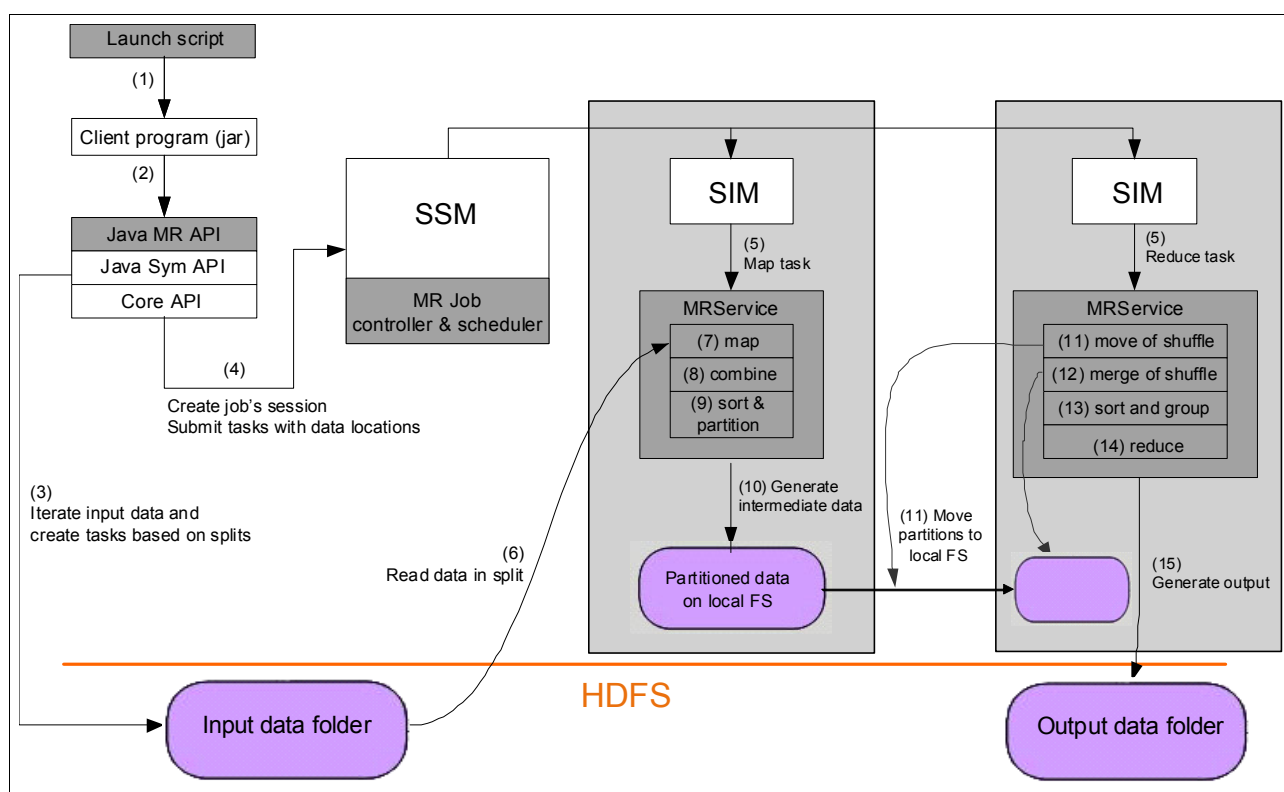


Figure 4-8 Symphony MapReduce work and data flow

When a MapReduce job is submitted by a client, preliminary processing is performed in the client's local environment to identify the splits of the input files and the required map tasks. A session is then created by the session manager for that job, consisting of map tasks that are dispatched on compute nodes according to the split locations previously identified. Reduce tasks are also dispatched in the session, as required.

The computation entities providing either map or reduce logic, which is based on the received task type, are MRService service instances. The SSM creates one SIM for each task, and the SIM then instantiates a corresponding MRService instance. A SIM is responsible for managing the lifecycle of a service instance and for routing client messages between the session manager and its associated service instance.

Each MRService service instance loads the application .jar file received from the client and implements the logic for the configured task type. For the map task, the MRService reads data records from the split that is passed as task input, then starts the application Mapper (and optionally Combiner) functions. It finally stores its intermediate data output to local files.

The intermediate data are partitioned depending on the way the intermediate keys are associated with the reduce tasks.

For the reduce task, the MRService instance asks the MapReduce shuffling service (MRSS) to get intermediate map output data for the related keys. It then merges and sorts the received data. The MapReduce shuffling service is a special MapReduce data transfer daemon that runs as a Symphony service on each host. The daemon optimizes local memory and disk usage to facilitate faster shuffling processes for map task output data to local and remote reduce tasks.

The reduce task then applies the user-defined Reducer logic to this merged and sorted data. The result is written to the output destination specified in the reduce task argument.

API adapter technology

The MapReduce framework allows Apache Hadoop MapReduce compatible applications to use Platform Symphony scheduling engine without any changes in the application code or any recompilation. This is obtained by inserting an application adapter layer in the original Hadoop MapReduce software stack as shown in Figure 4-9. The Hadoop MapReduce application calls the same Hadoop MapReduce and Hadoop Common Library APIs.

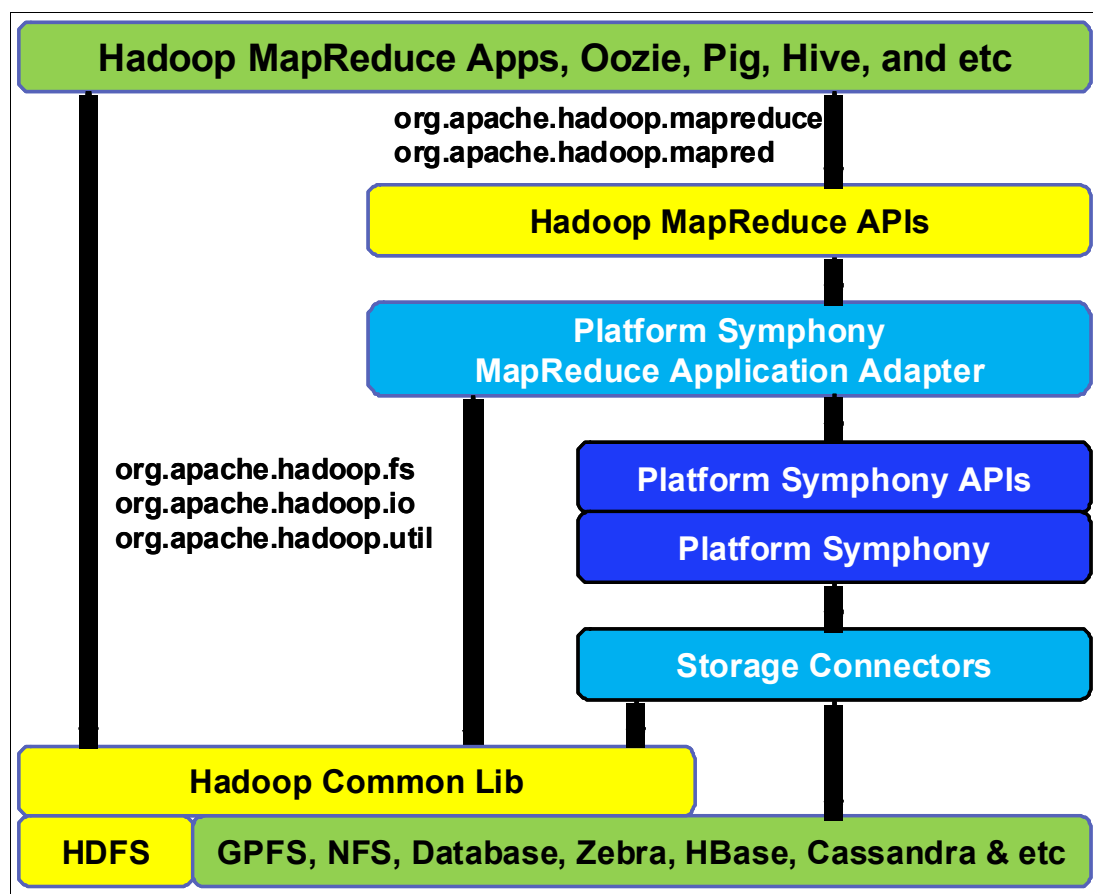


Figure 4-9 MapReduce application adapter

To route these calls through the MapReduce Application Adapter, preliminary configuration is needed before starting the applications. Normally, perform this configuration at installation time. The *User Guide for the MapReduce Framework*, GC22-5370-00, recommends that you completely install and configure the Hadoop distribution before you install Platform Symphony on the cluster nodes. This simplifies the steps of this integration configuration.

The initial step consists of some settings in the shell environment, and then at the MapReduce application level inside Platform Symphony.

As a second step, you might have to remove some system or platform dependencies at the Hadoop application level. Application classes and Hadoop base classes might come packaged into the same .jar (or .war) file. The Java class loader looks first at the classes in the .jar files that are already loaded in the JVM together with application classes. Therefore, you must repackage the application in its own .jar file, separating the Hadoop classes. This applies, for example, to Pig and Oozie applications.

For a complete list of the supported Apache Hadoop applications and their detailed repackage steps, see *Integration Guide for MapReduce Applications*, SC27-5071-00. The Platform Symphony documentation also contains details about integration with some commercial Hadoop distributions, including IBM InfoSphere BigInsights and Cloudera CDH. For examples of Apache Hadoop and IBM InfoSphere BigInsights integration with Platform Symphony see Chapter 5 - IBM Platform Symphony in *IBM Platform Computing Solutions*, SG24-8081, and Chapter 2 - Integration of IBM Platform Symphony and IBM InfoSphere BigInsights in *IBM Platform Computing Integration Solutions*, SG24-8081.

Figure 4-9 on page 69 shows that both the standard Hadoop stack and the application adapter use the Hadoop common library for HDFS, and for other file systems such as GPFS or NFS if configured. The MapReduce framework in Platform Symphony also provides storage connectors to integrate with different file systems and databases.

Figure 4-10 shows when and where in the workflow of a submitted MapReduce job the various portions of the code are used: From the application itself, from the core Hadoop components, or from the Platform Symphony application adapter.

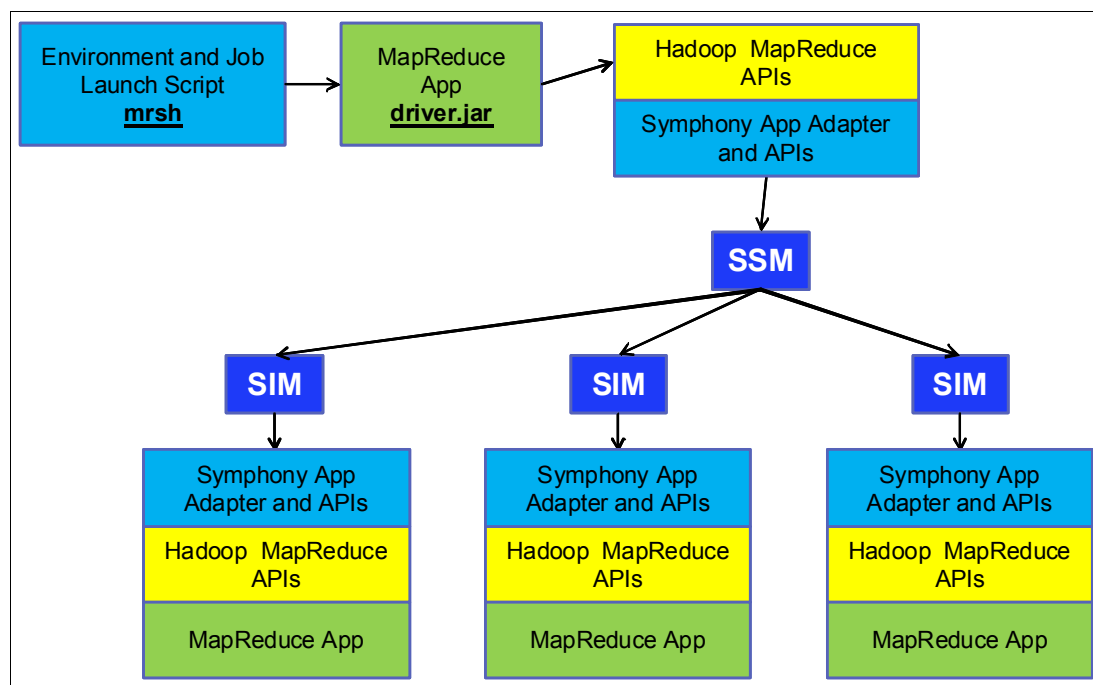


Figure 4-10 Application adapter in the MapReduce workflow

4.2 Key advantages for Platform Symphony MapReduce

This section describes some of the advantages of the Symphony MapReduce framework compared to the standard open source Apache Hadoop environment, and other commercial distributions of the Hadoop environment. It also describes the advantages from a cloud-computing perspective because the Platform Symphony product offers multi-tenancy, resource sharing, low-latency scheduling, and heterogeneous features. For more information about these topics in the larger context of the Platform Symphony product, see 3.7.1, “Advantages of Platform Symphony in Technical Computing Cloud” on page 56. This section focuses on features that are directly related to the MapReduce component. The following key features make Platform Symphony MapReduce a good MapReduce framework option:

- ▶ **Performance:** A low latency architecture allows a much higher performance for certain workloads (jobs).
- ▶ **Dynamic resource management:** Slot allocation changes dynamically based on job priority and server thresholds, loaning, and borrowing.
- ▶ **Sophisticated scheduling engine:** A fair share scheduling scheme with 10,000 priority levels can be used for multiple jobs of the same application. Also, pre-emptive and resource threshold-based scheduling is available with runtime change management.
- ▶ **Management tools:** Platform Symphony provides a comprehensive management capability for troubleshooting alerting and tracking jobs, and rich reporting capabilities.
- ▶ **Reliability:** Platform Symphony makes all MapReduce and HDFS-related services highly available such as name nodes, job trackers, and task trackers.
- ▶ **Application lifecycle:** There is support for rolling upgrades for Platform Symphony. Also, multiple versions of Hadoop can coexist in the same cluster.
- ▶ **Open:** Platform Symphony supports multiple APIs and languages. It is fully compatible with Java, Pig, Hive, and other MapReduce applications. Platform Symphony also supports multiple data sources, including HDFS and GPFS.

The following sections present these features in more detail.

4.2.1 Higher performance

Significant performance improvement for the Symphony MapReduce framework is expected for most of the MapReduce jobs when compared with the open source Hadoop framework, especially for the short-run jobs. This improvement is based mainly on the low latency and the immediate map allocation plus the job startup design features of the SOA framework. Platform Symphony also has some performance optimizations for specific components of the Symphony MapReduce framework such as the intermediate data shuffling operation run between the map and reduce phases.

For more information about performance comparison tests and benchmarks that are run on IBM Platform Symphony, see:

<http://www-03.ibm.com/systems/technicalcomputing/platformcomputing/products/symphony/highperfhadoop.html>

Low latency

Low-latency is one of the key features of the Platform Symphony MapReduce framework. It comes directly from the approach that is used in SOA, which was designed as a low latency architecture for managing compute-intensive applications. The demand for such an architecture came from the financial services space, particularly from the investment banking.

Such analytics infrastructure was needed to run jobs such as Monte Carlo simulations, risk analysis, and credit validations, in a more interactive, near real-time manner.

The latency of the open source Hadoop engine is determined by the broker logic it uses. The broker, which is the JobTracker, runs as a service that is waiting for notifications and requests from the individual TaskTrackers.

A heartbeat mechanism is implemented between the JobTracker and its monitored TaskTrackers. All TaskTrackers must signal to the JobTracker that they are alive by sending periodical heartbeat messages to the JobTracker. The JobTracker declares a TaskTracker as 'lost' if it does not receive heartbeats during a specified time interval (default is 10 minutes). Tasks in progress that belong to that TaskTracker are then rescheduled.

Extra information is piggybacked on the heartbeat messages for various other purposes. For example, task status information is conveyed from the TaskTracker to the JobTracker. When task reports progress, status flags and progress counters are updated inside the TaskTracker. These flags and counters are sent more or less frequently through these heartbeat messages to the JobTracker. The JobTracker uses them to produce a global view with the status of all the running jobs and their constituent tasks.

A TaskTracker indicates when one of its tasks finishes and whether it is ready to receive a new task through the same heartbeat messages. If the JobTracker allocates a task, it communicates this to the TaskTracker by a heartbeat return value.

In initial versions of open source Hadoop, the task trackers sent heartbeats at 10 second intervals. Later versions made the heartbeat interval variable. It became dependent on the size of the cluster and had to be calculated dynamically inside the JobTracker. A minimum heartbeat interval limit of 3 seconds was enforced by this computation. For larger clusters, the computed interval was longer, as intended. But for short run map tasks, this still resulted in an underused cluster. To improve latency on small clusters, the default minimum heartbeat interval has been lowered, starting with release 1.1, from 3 seconds to 300 ms. Reducing this interval even more is problematic, though, because the higher the heartbeat rate, the more processor load the JobTracker must handle, so it might become unresponsive at high load.

A new option was also introduced to allow the TaskTrackers to send an out of band heartbeat on task completion. For jobs with short tasks, the JobTracker can get flooded with too many heartbeat completions, so there is an out of band heartbeat option to mitigate too many completion heartbeats.

Overall, these heartbeat interval changes were meant to improve the scheduling throughput. But the polling logic and the minimum limit of the heartbeat interval determine the job latency. The heartbeat exchange between the JobTracker and the TaskTrackers uses the HTTP protocol as a communication vehicle, and slow, heavy-weight XML data for encoding text messages. This affects the scalability because the JobTracker cannot respond if the heartbeat interval is too small.

Therefore, the polling window approach used in Hadoop has this disadvantage of wasted time, which becomes more critical in the case of short run jobs. The polling window takes a significant amount of time to allocate work to the individual tasks or the individual TaskTrackers, compared with the duration of the task itself.

The Platform Symphony deal with this problem in a much more efficient way. Symphony uses a different approach, called the push model, which is an interrupt-based type of logic. When a service instance completes a piece of work, the SSM immediately pushes a new piece of work to that instance. This results in milliseconds of response time in terms of latency. Platform Symphony encodes data in a more compact binary format and uses an optimized proprietary binary protocol to communicate between the SSM and the service instances.

Being implemented in C++, the protocol allows fast workload allocation and minimal processor load to start jobs when compared to the Hadoop open source approach, which is written in Java.

For more information about this core feature of the SOA framework, see Figure 3-2 on page 32 and 3.2.1, “Compute intensive applications” on page 31.

Optimized shuffling

For the shuffling phase, MapReduce framework uses a Symphony service that runs on each host as a special MapReduce data transfer daemon (shuffling service-MRSS). The daemon optimizes local memory and disk usage to facilitate faster shuffling processes for map task output data to local and remote reducer tasks. This optimized shuffling works especially well for jobs where map output data can be kept mostly in memory and retrieved by reducers before the next rounds of mapper tasks generate more output.

The shuffling optimization has potential of improving the things by this memory leveraging on the individual nodes in three related areas. The map tasks basically run on the input split data and have as output an intermediate data set. This intermediate data set is kept in memory as much as possible instead of running it out to the local disk, reducing the number of I/O cycles that are associated with writes. If it is too large for the available memory, it goes to disk. When, during the shuffle phase, this data must traverse the network, they is retrieved from memory rather than from disk. Similarly, on the reducer side, instead of writing them to disk, they are stored in memory, if possible. When, finally, the reducer reads these data, they are there in memory, again saving I/O cycles.

This works well for a significant number of MapReduce jobs because their map outputs are not large compared with the current affordable memory sizes in typical technical computing nodes. The generated output tends to be much smaller, in the order of GB, for example, or even less. As shown in Figure 4-11, intermediate data is kept in memory if possible, contributing to the overall performance advantages brought by fast low-latency scheduling and data movement. All these explain how Platform Symphony can improve the speed of even workloads like TeraSort that are not overly sensitive to scheduling latency.

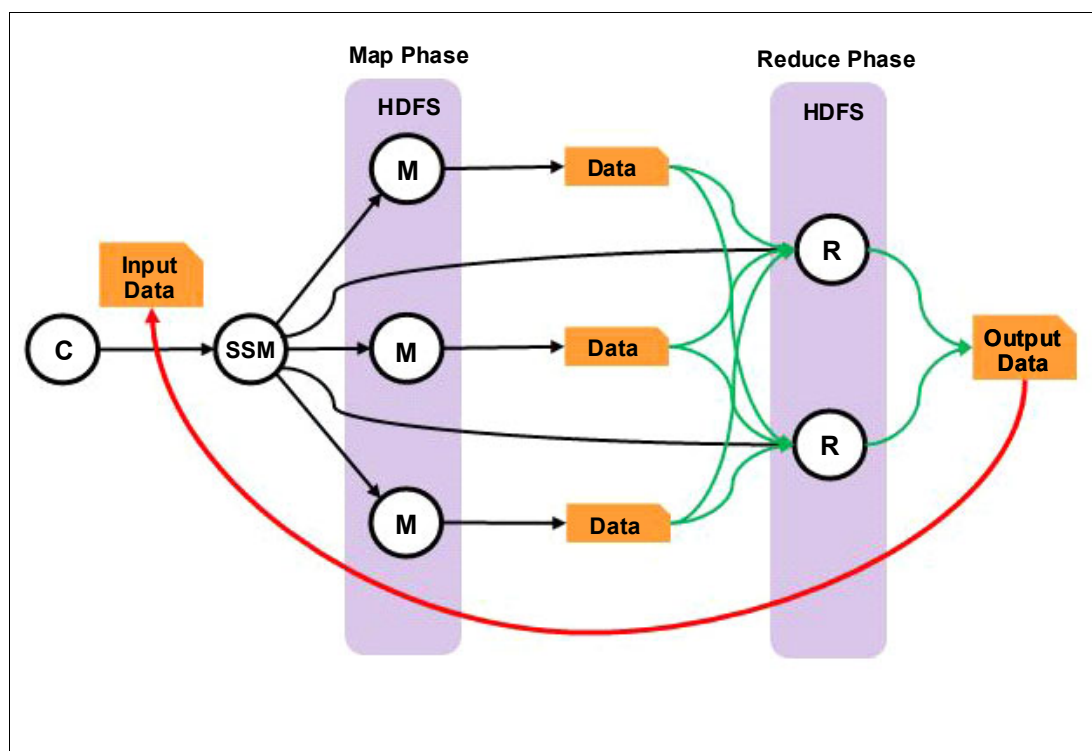


Figure 4-11 Speeding up iterative MapReduce patterns

There exists, also, a class of MapReduce jobs where the overall output is minimal as well and can be kept in memory. Loop-invariant data is kept in memory, if possible, between iterations, further improving the performance speed. As an example, Platform Symphony behaves well on workloads like K-Means that are iterative and use the output of one MapReduce workload as inputs to the next.

The last class of MapReduce jobs are those that repeatedly read common data input splits. By caching the input splits at the first iteration of a job, subsequent iterations reuse the cached input split as input. This process avoids fetching the same data from the HDFS repeatedly. Analytics applications that support iterative algorithms and interactive data mining benefit from this feature.

Data input splits of a map task are mapped to system memory and to a file on the local disk (specified as the `PMR_MRSS_CACHE_PATH` environment variable). If this cache is not accessed for a time period (specified as `PMR_MRSS_INPUTCACHE_CLEAN_INTERVAL`) or exceeds the maximum memory limit (specified as `PMR_MRSS_INPUTCACHE_MAX_MEMSIZE_MB`), the least recent split and its associated local disk files are deleted. You must adapt these specified parameters to the available memory. To enable a job to get its input split from the cache, you must set these caching options in the shuffle service (mrss) definition file before you submit the job.

4.2.2 Improved multi-tenant shared resource utilization

In Platform Symphony Advanced Edition, up to 300 MapReduce runtime engines (job trackers) can coexist and use the same infrastructure. Users can define multiple MapReduce applications and associate them with resource consumers by “cloning” the default MapReduce application. Each application has its separate and unique Job Tracker (SSM). When multiple SSMs are instantiated, they are balanced on the available management nodes. Furthermore, inside each application, simultaneous job management is possible because of the special design that implements sophisticated scheduling of multiple sessions on the resources that are allocated for an application. This function is obtained by separating the job control function (workload manager) from the resource allocation and control (Enterprise Grid Orchestrator). The new YARN, Apache Hadoop 2, has a similar feature, but this release is still in alpha stage. The stable release of Hadoop MapReduce offers only one Job Tracker per cluster.

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

Users and security

To allow users to use resources when running their applications in a managed way, Platform Symphony implements a hierarchical model of consumers. This tree of consumers allows association of users and roles on one hand with applications and grid resources on the other. Policies for the distribution of resources among multiple applications run by different users can be configured this way to share the resources in the grid.

MapReduce applications, and other non-MapReduce applications such as standard SOA compute-intensive applications, inside Platform Symphony can use the same infrastructure. In addition, a multi-head installation of both Platform LSF and Platform Symphony is supported. This installation allows batch jobs from LSF, and compute-intensive and data-intensive applications from Symphony to share the hardware grid infrastructure.

A security model is enforced for the authentication and authorization of various users to the entitled applications and to isolate them when they try to access the environment. You can create user accounts inside the Platform Symphony environment, as shown in Figure 4-12, then assign to them either predefined or user created roles. User accounts include optional contact information, a name, and a password.

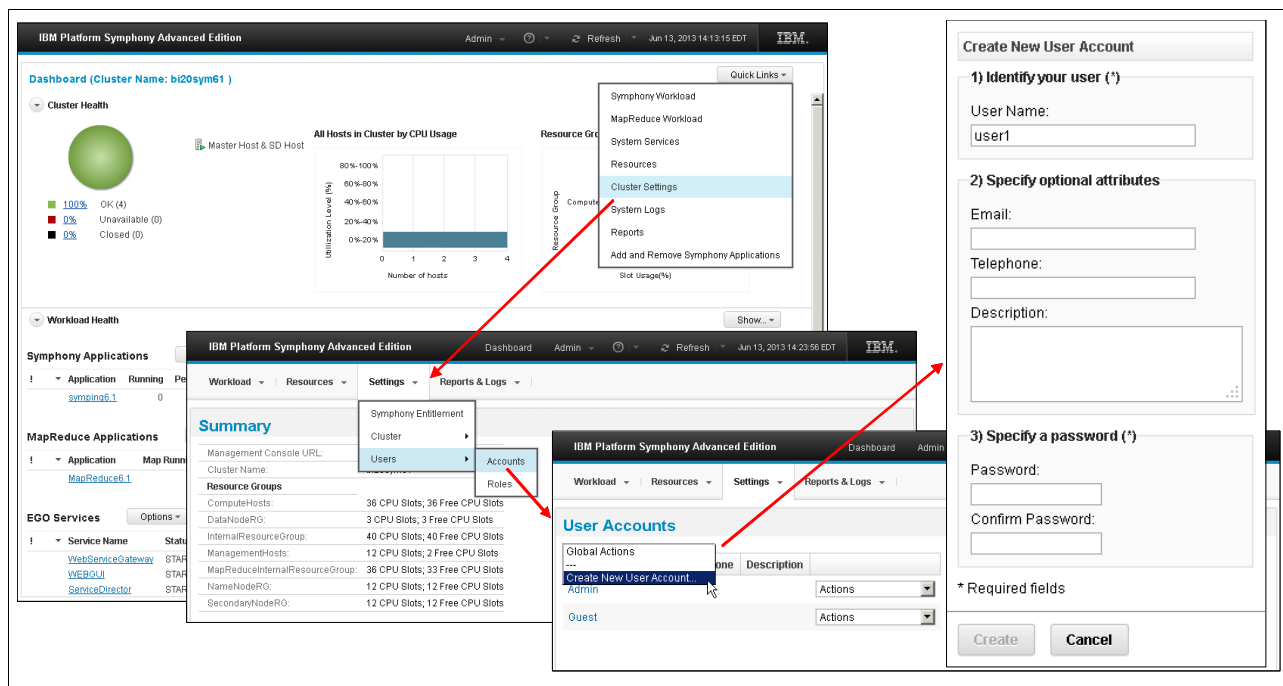


Figure 4-12 Creating user accounts

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

- ▶ **Cluster administrator**
A user with this role can perform any administrative or workload-related task, and has access to all areas of the Platform Management Console and to all actions within it.
- ▶ **Cluster administrator (read only)**
This user role allows read-only access to any cluster information, but cannot perform any add, delete, or change action.
- ▶ **Consumer administrator**
Users with this role are assigned to a top-level consumer in the consumer hierarchy, and can administer all subconsumers in that branch of the tree.
- ▶ **Consumer user**
Consumer users are assigned to individual consumers on the tree, and have access and control only over their own workload units.

You can also create customized user roles and use them in addition to the predefined roles. You can choose from a set of predefined permissions and apply them to new or existing user roles. There are cluster-wide permissions or per consumer permissions, for one or more consumers in the hierarchy as shown in Figure 4-13.

Figure 4-13 Creating user roles

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

Sharing resources

An application can be used only after it is registered and enabled. You can only register an application at a leaf consumer (a consumer that has no subconsumers). Only one application can be enabled per consumer.

Before you can register an application, you must create at least one consumer, and deploy the service package of the application to the intended consumer. You can deploy the service package to a non-leaf consumer so that all applications registered to child leaf consumers are able to share the service package. A service package is created that puts all developed and compiled service files and any dependent files associated with the service in a package.

Resource distribution plan

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

You must initially assign bulk resources to consumers in the form of resource groups to simplify their management. Later you can change this assignment. Resource groups are logical groups of hosts. A host in a resource group is characterized by a number of slots. The number of slots is a variable parameter. When you choose a value for it, the value must express the degree of specific workload that the host is able to serve. A typical slot assignment is, for example, the allocation of one slot per processor core.

After it is created, a resource group can be added to each top-level consumer to make it available for all the other subconsumers underneath. Figure 4-14 shows an example of a consumer tree with all its top-level consumers and their assigned resource groups and users. Platform Symphony provides a default top-level consumer, MapReduceConsumer, and a leaf-consumer, MapReduceversion, which in this example is MapReduce61.

IBM Platform Symphony Advanced Edition

DashboardAdminRefreshJun 16, 2013 05:05:49 EDT

WorkloadResourcesSettingsReports & Logs

Consumers

bi20sym61

ManagementServices

EGOManagementServices

SymphonyManagementService

ComputeServices

MapreduceComputeServices

SymTesting

Symping61

SampleApplications

SOASamples

EclipseSamples

SymExec

SymExec61

ClusterServices

EGOClusterServices

SymphonyClusterServices

MapReduceConsumer

MapReduce61

HDFS

NameNodeConsumer

SecondaryNodeConsumer

DataNodeConsumer

Consumers

Global Actions

Consumer Name	Resource Group	Administrators	Users	
ClusterServices	InternalResourceGroup			Actions
ComputeServices	MapReduceInternalResourceGroup	Admin	Admin	Actions
HDFS	DataNodeRG , NameNodeRG , SecondaryNodeRG	Admin	Guest , Admin	Actions
ManagementServices	ManagementHosts	Admin	Admin	Actions
MapReduceConsumer	ComputeHosts , ManagementHosts	Admin	Guest , Admin	Actions
SampleApplications	ComputeHosts , ManagementHosts			Actions
SymExec	ComputeHosts , ManagementHosts			Actions
SymTesting	ComputeHosts , ManagementHosts			Actions

Expand All

Collapse All

Figure 4-14 Consumer tree

The concepts that are used inside a resource distribution plan, as shown in Figure 4-15, are ownership, borrowing and lending, sharing, reclaiming of borrowed resources, and rank:

- **Ownership:** The guaranteed allocation of a minimum number of resources to a consumer.
- **Borrowing and lending:** The temporary allocation of owned resources from a lending consumer to a consumer with an unsatisfied demand.
- **Sharing:** The temporary allocation of unowned resources from a “share pool” to a consumer with an unsatisfied demand.
- **Reclaiming:** Defines the criteria under which the lender reclaims its owned resources from borrowers. The policy can specify a grace period before starting the resource reclaim, or the policy can specify to stop any running workload and reclaim the resources immediately.
- **Rank:** The order in which policies are applied to consumers. Rank determines the order in which the distribution of resources is processed. The highest ranking consumer receives its resources first, borrows resources first, and returns borrowed resources last.

Resource Plan

Resource Group: ComputeHosts Time Intervals and Settings

► Slot allocation policy

Consumer	Owned Slots	Consumer Rank	Lend Limit	Borrow Limit	Share Ratio	Limit
bi20sym61	36					
SymTesting	2	0			1	
Symping61	1	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	
Total	1	-	-	-	-	-
Balance	1	-	-	-	-	-
SampleApplications	0	0			1	
SOASamples	0	0	<input type="checkbox"/>	<input type="checkbox"/>	1	
EclipseSamples	0	50	<input type="checkbox"/>	<input type="checkbox"/>	1	
Total	0	-	-	-	-	-
Balance	0	-	-	-	-	-
SymExec	0	0			1	
SymExec61	0	0	<input type="checkbox"/>	<input type="checkbox"/>	1	
Total	0	-	-	-	-	-
Balance	0	-	-	-	-	-
MapReduceConsumer	2	0			1	
MapReduce61	1	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	
Total	1	-	-	-	-	-
Balance	1	-	-	-	-	-
Total	4	-	-	-	-	-
Balance	32	-	-	-	-	-

► Expand All
▼ Collapse All

Apply Revert Import... Export...

Lend Details:
Symping61 (ComputeHosts, 00:00-24:00)
Total lend limit:
Lend to these consumers
Consumers to lend to Lend / Limit
bi20sym61
 SymTesting
 Symping61
 SampleApplications
 SOASamples
 EclipseSamples
 SymExec
 SymExec61
 MapReduceConsumer
 MapReduce61

Borrow Details:
MapReduce61 (ComputeHosts, 00:00-24:00)
Total borrow limit:
Borrow from consumers
Consumers to borrow from Borrow / Order
bi20sym61
 SymTesting
 Symping61
 SampleApplications
 SOASamples
 EclipseSamples
 SymExec
 SymExec61
 MapReduceConsumer
 MapReduce61

► Expand All
▼ Collapse All

Apply Revert Close

Figure 4-15 Resource plan

The first allocation priority is to satisfy each consumer's reserved ownership. Remaining resources are then allocated to consumers that still have demand. Unused owned resources from consumers willing to lend them are then allocated to demanding consumers that are entitled to borrow. The resource orchestrator then allocates the unowned resources from the share pool to consumers with unsatisfied demand and entitled to this type of resources. The resources from the “family” pool (any unowned resources within a particular branch in the consumer tree) are allocated first. After the family pool is exhausted, the system distributes

resources from other branches in the consumer tree. The free resources in the shared pools are distributed to competing consumers according to their configured share ratio. A consumer that still has unsatisfied demand and has lent out resources, reclaims them back at this stage. Owned resources are reclaimed first, followed by the entitled resources from the shared pool currently used by consumers with a smaller share-ratio. This is the default behavior. The default behavior can be changed such that owned resources are recalled first before trying to borrow from other consumers.

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

As shown in Figure 4-15 on page 79, each resource group must have its own plan. Also, you can define different resource plans for distinct time intervals of the day, allowing you to better adapt them to workload patterns. At the time interval boundary, the plan change might determine important resource reclaiming.

For more information about sharing resources, see 3.4, “Advanced resource sharing” on page 42. For in-depth coverage, see *Cluster and Application Management Guide*, SC22-5368-00.

MapReduce scheduling policies

In addition to the sophisticated hierarchical sharing model of resources among consumers and applications, you can also schedule resources of an application among the multiple sessions that can run simultaneously, and also, at the session level, among the tasks of a session. For more information about the general scheduling approach within an application of the Platform Symphony SOA framework, see 3.3.5, “Job scheduling algorithms” on page 39. Two of the SOA framework scheduling policies are used by MapReduce applications: Priority scheduling and proportional scheduling. These policies are further tailored for MapReduce workload, and are introduced here with some of these specific MapReduce extensions.

After it is allocated to a running MapReduce application, its resources are further split into the map pool and the reduce pool. This split is based on the map-to-reduce ratio parameter that is specified in the application profile. Each pool is divided among the jobs (sessions) of the application depending on the configured policy type:

- ▶ For priority scheduling, the job with the highest priority gets as many slots from each pool as it can use before other jobs get resources. The allocation then continues with the next jobs, in decreasing order of their priority.
- ▶ For proportional scheduling, each job gets a share in each pool that is proportional to its relative priority compared to the priorities of the other jobs.

Up to 10,000 levels of prioritization are supported for the jobs of an application, 1 being the lowest priority level, and 5000 the default value. The priority can be modified while the job is running from the console GUI or from the command line. Hadoop’s priorities (`mapred.job.priority`) are directly mapped to values in this range as follows:

```
VERY_LOW=1
LOW=2500
NORMAL=5000
HIGH=7500
VERY_HIGH=10000
```

At each scheduling cycle, resources that are already assigned might be reassigned from one job to another based on a preemption attribute of each job, which can take a true or a false value. If true, the session can reclaim resources from other overusing sessions. If the

deserved share of resources of a demanding job is not available and the preemption value is false (the default), that job must wait for a current running task to finish before resources can be served.

When a preemptive job reclaims resources and the system finds multiple sessions or tasks that can provide slots, even after considering preemption ranks or session priorities, other attributes can be used to decide which task to interrupt to reclaim a resource. You can use the “enableSelectiveReclaim” and “preemptionCriteria” attributes. Jobs notified that their tasks will be interrupted get the chance to clean up and finish its involved tasks in a configurable timeout interval. If this grace period expires, the tasks are ended and requeued to pending state.

For more information about this scheduling topic and related attributes and policies, see the Version 6 Release 1.0.1 product manuals of the *User Guide for the MapReduce Framework and Platform Symphony Reference*, GC22-5370-00.

Data locality

The map tasks input data splits (in more replicas, usually three) are pseudo-randomly distributed on the HDFS nodes, which are also compute nodes in the Platform Symphony cluster. The nodes themselves are commonly spread across many racks and, can be spread across different data centers, as shown in Figure 4-16.

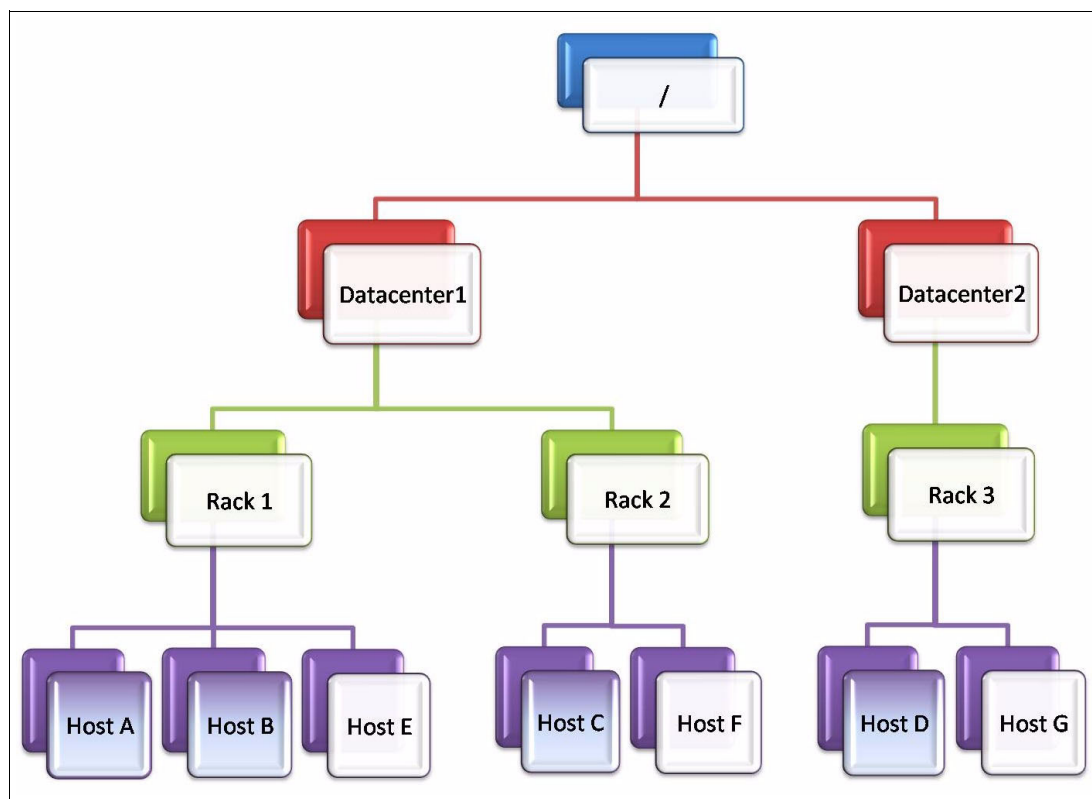


Figure 4-16 Network topology tree

Communication between two nodes in the same rack goes usually through a top rack switch. Communication between nodes in distinct racks might involve expensive bandwidth in external core switches and other networking resources. If you organize hosts while scheduling resources in such a way that the traffic between nodes in the same rack is favored over the traffic between nodes in different racks, performance and resource utilization improves. Platform Symphony implements this function through a rack-aware scheduling

feature. A network topology of the cluster in the form of a tree (Figure 4-16 on page 81) is derived from physical location information that is provided by user-defined topology scripts. This tree provides a network “distance” concept between compute nodes as the sum of their distances to their closest common ancestor node in the tree.

When a MapReduce job is submitted, the SSM produces a list of preferred hosts that consists of those that store the replicas of the input data splits. This information is submitted to Enterprise Grid Orchestrator (EGO) along with the appropriate allocation request of slots for the map tasks. For each input split and preferred host apart, a check is then made by EGO, looking for available slots. If they are not available, EGO checks for slots on hosts that are closest to the preferred host by searching breadthwise across the preferred host's parent in the topology tree. If slots are still not available, it looks up the parent's parent, and so on. In the same request, the SSM also provides preferences for reduce tasks to be placed on the same resources as related map tasks, or as close as possible. This minimizes the network traffic during shuffling of intermediate data.

EGO tries to satisfy all the allocation requests from multiple applications by taking into account extra data locality preferences in the case of the MapReduce applications. Upon receiving the allocated slots, the SSM schedules the map tasks with the “closest” input data locality using the same “distance” between hosts. At the same time, the SSM also tries to schedule reduce tasks evenly among allocated hosts, and place reduce tasks closer to related map tasks.

When reclaiming resources from a MapReduce application, SSM first tries to reclaim resources from a map task with the farthest input data distance or, depending on the ratio between the total number of map and reduce tasks, from a reduce task running farther from any related map task.

After the tasks are dispatched to the hosts, use the console GUI to view the type of match, whether node-local, rack-local, or off-rack, on the task's Summary tab.

Interactive management

All the management tasks can be performed from either the web-based Platform Management Console GUI or from the command line.

This section addresses some of the most interesting management tasks, emphasizing the high level of interactivity with the running jobs and applications. For example, the resource distribution plan that was introduced in “Resource distribution plan” on page 78 can be modified online, while applications are running their workloads. The EGO resource manager takes care of all changes and controls all the reassignment and reclaiming tasks that might be performed.

Another related feature is the administrative control of running jobs. Actions like Suspend, Resume, and Kill can be performed at Job and Task level, on top of the usual Monitoring, Monitoring, Troubleshooting, and Reporting features. Even the priority attributes of running MapReduce jobs can be changed online, making the environment highly interactive when this kind of prioritization requirement is needed.

4.2.3 Improved scalability

The scalability of an enterprise MapReduce grid computing environment is an important feature as business requirements usually grow. You must make sure that the continuously increasing workload does not saturate your application, middleware, and hardware infrastructure. However, just building a larger cluster is not enough because you also need to be able to keep the cluster fully busy and to operate all these resources efficiently. The

following critical performance and scalability limits are supported by the Platform Symphony 6.1 as a whole, and by its MapReduce runtime engine:

- ▶ 10,000 physical servers per cluster
- ▶ 40,000 cores per cluster
- ▶ 10,000 cores per application
- ▶ 100,000 cores in multicluster configurations
- ▶ 40,000 service instances (concurrent tasks) per cluster
- ▶ 1,000,000 of pending tasks per cluster
- ▶ 1 ms delivered latency for grid services
- ▶ 17,000 tasks per second as enabled throughput
- ▶ 1,000 instances per second as rate of service instances reallocation
- ▶ 300 MapReduce applications (SSMs, JobTrackers) per cluster
- ▶ 1000 concurrent jobs per MapReduce application
- ▶ 10,000 discrete priorities for the jobs of a MapReduce application
- ▶ 20,000 slots per SSM

These values are taken from the product manuals and the availability announcement of IBM Platform Symphony 6.1 software product:

http://www-01.ibm.com/common/ssi/rep_ca/6/897/ENUS212-426/ENUS212-426.PDF

4.2.4 Heterogeneous application support

Platform Symphony provides an open application architecture that is based on the SOA framework that allows compute-intensive and data-intensive applications to run simultaneously in the same cluster. Up to 300 separate MapReduce applications (Job Trackers) can share the grid resources with other types of distributed applications, simultaneously. This allows customers to use both existing and new resources, and maximize their IT infrastructure while maintaining a single management interface.

The MapReduce framework in Platform Symphony is built on an open architecture that provides 100% Hadoop application compatibility for Java based MapReduce jobs. The application adapter technology that is built into the product delivers seamless integration of Hadoop applications with Platform MapReduce so that jobs built with Hadoop MapReduce technology (Java, Pig, Hive, and others) require no changes to the programming logic for their execution on a grid platform. For more information about the integration of third-party applications to the MapReduce framework, see “API adapter technology” on page 69.

This Symphony MapReduce open architecture also provides methods for using multiple file system types and database architectures. Storage connectors, as they are called, allow integration with IBM General Parallel File System (GPFS), Network File System (NFS), local file system, and other distributed file system types and data types. In addition, for MapReduce processes, the input data source file system type can be different from the output data source file system. This provides support for many uses, including extract, transformation, and load (ETL) workflow logic.

Well-documented SOA APIs allow seamless integration of applications that are written in other languages and software environments:

- ▶ C++
- ▶ C#, .NET
- ▶ Java™
- ▶ Excel COM
- ▶ Native binaries

Platform Symphony also supports other languages and environments that are commonly used in distributed computing, including Python. Plug-ins that are provided for Microsoft Visual Studio Professional support compilation-free integration of .NET assemblies. Developers can use a step-by-step wizard to integrate and test applications end-to-end without needing to be expert in the Platform Symphony APIs. For Java developers, the Eclipse integrated development environment (IDE) is supported as well.

For more information about the integration with various languages and development environments, check the product manual Application Development Guide.

4.2.5 High availability and resiliency

A Platform Symphony cluster is a mission critical environment in most cases. The grid should never go down as a whole. Components such as management nodes or compute nodes that run critical system services or application workloads might fail, so cluster resiliency mechanisms such as automatic failure recovery must be available.

Automatic failure recovery can be configured for all the workload management (service-oriented application middleware) and resource management (EGO) components that run on master hosts. The master host fails over to an available master candidate host in a list of master candidates. To work properly, the master candidate failover requires that the master host and the master candidates share a file system that stores the cluster configuration information. The shared file system also must be highly available, such as a Highly Available Network File System (HA-NFS) or an IBM GPFS file system.

Figure 4-17 presents in more detail some of the involved entities and a recovery approach for the failure case of a host running a service instance. For an in-depth presentation for the roles and functions of the involved system daemons, see *Platform Symphony Version 6 Release 1.0.1 Application Development Guide*, SC27-5078-01.

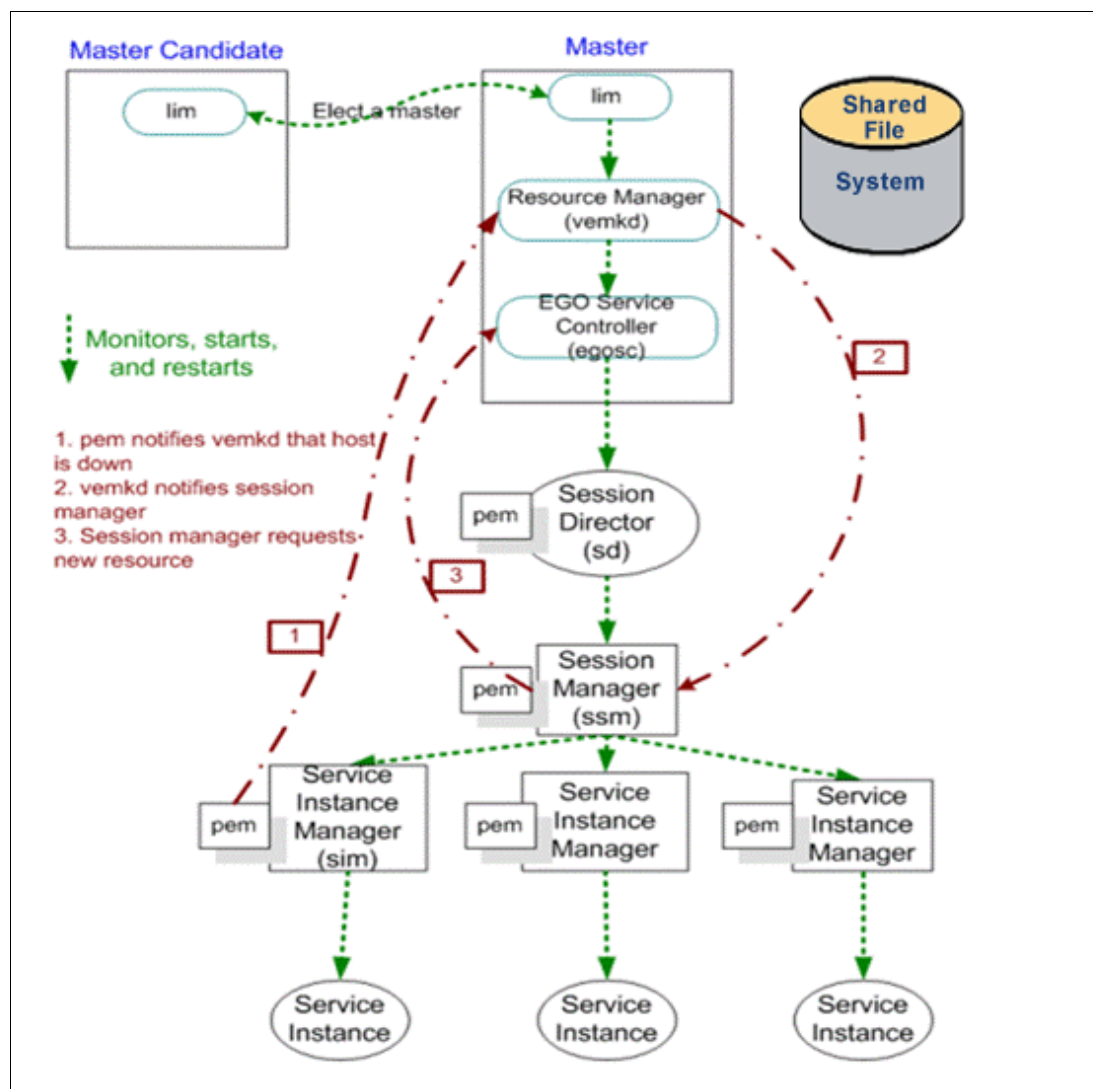


Figure 4-17 System level resiliency

Part of this recoverability is in place for the Symphony MapReduce framework as well. Open source Hadoop itself has by design recovery mechanisms for a failed TaskTracker. However, the JobTracker in the stable releases is still a single point of failure (SPOF). In Platform Symphony, the whole MapReduce framework runs on top of the SOA framework. The Application Manager (SSM) and the service instances replace the JobTracker and the TaskTrackers. The high availability of these two components is embedded in the Platform Symphony design.

Within the MapReduce framework, the HDFS component can also be configured as highly available by automating a failover process for all the involved daemons on any of the nodes (the NameNode, the SecondaryNameNode, and the data nodes). The HDFS daemons are integrated as MapReduce framework system services into the EGO system. The EGO system monitors them and restarts them on the same or another management host if a failure occurs. A built-in EGO DNS service running a Linux 'named' daemon is used to register any

system service name as a well-known host name mapped to the IP address of the host where the service instance is running.

The NameNode is configured with such a well-known host name on which it accepts requests from the SecondaryNameNode, data nodes, and HDFS clients like MapReduce applications. When a NameNode daemon fails over from a failed host to a working host, the EGO DNS service reassigns the well-known NameNode host name to the IP address of the working host. This way, all of the involved entities can ask for NameNode services at the same well-known host name. The whole failover mechanism is shown in Figure 4-18.

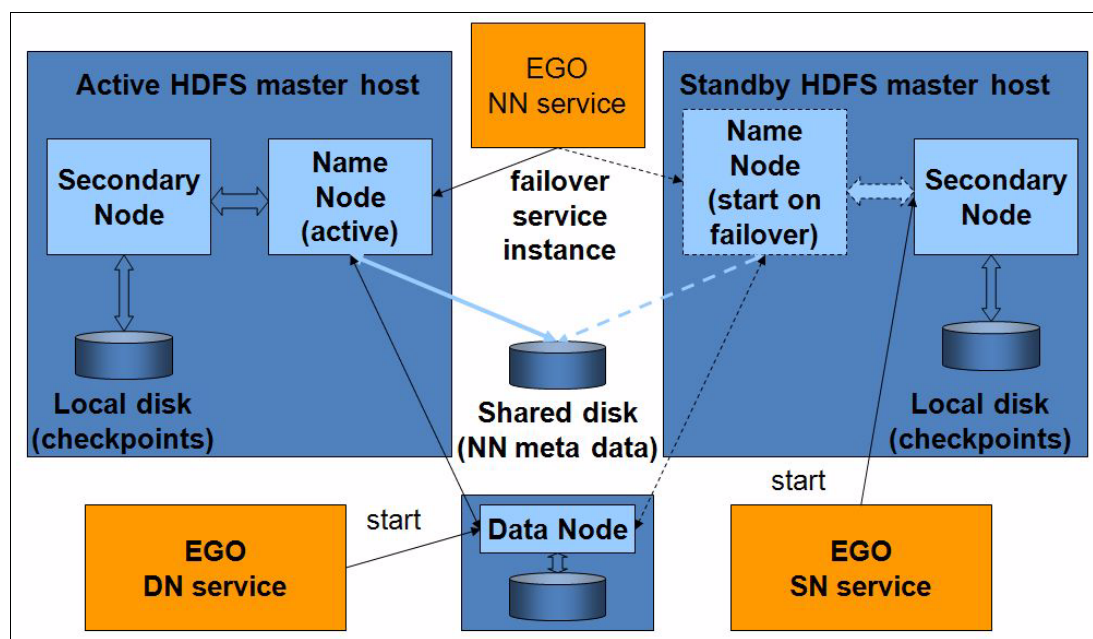


Figure 4-18 HDFS high-availability setup

The disk space that is required by the NameNode for its metadata files must be provided as a shared file system. Make this system highly available as either an HA-NFS or an IBM GPFS file system. Although this HDFS failover logic maintains the metadata on the shared disk and the node is brought up quickly, the HDFS NameNode service must go through the whole recovery logic. The HDFS recovery logic can take a long time (minutes or even more) because each individual DataNode must reregister its data chunks with the NameNode. The the NameNode must build a namespace image into memory for performance reasons and resynchronize the persistent state that it keeps on the shared disk. This is a limitation of the current Hadoop HDFS design. The problem might be solved by avoiding the centralized metadata management, which might also eliminate the need for the failover logic.

A good candidate for such an approach is GPFS with its new FPO feature, which has been recently announced. For more information about the new GPFS FPO feature, see 6.4.2, “File Placement Optimizer (FPO)” on page 129.

4.3 Key benefits

This section provides a summary of the key benefits of using the Platform Symphony technology:

- ▶ High resource utilization
 - Single pool of shared resources across applications
 - Eliminates silos and single purpose clusters
- ▶ Performance
 - Low latency architecture
 - Many jobs across many applications simultaneously
- ▶ Flexibility
 - Compatible with open source and commercial APIs
 - Supports open source and commercial file systems
- ▶ Reliability and availability
 - Ensures business continuity
 - Enterprise-class operations
- ▶ Scalability
 - Extensive customer base
 - 40000 cores per cluster/300 simultaneous MapReduce applications
- ▶ Manageability
 - Ease of management, monitoring, reporting, and troubleshooting
- ▶ Predictability
 - Drives SLA-based management

Maybe the most significant benefit from these is the resource utilization. You can use Platform Symphony to organize the computing resources in shared pools and then allocate guaranteed portions of them to specific applications. Furthermore, you can borrow and lend in specified amounts from these allocated resources between specified applications. You can also use a shared pool approach that allows resource utilization by multiple applications according to configurable share ratios. All of these factors enable a sophisticated shared services model among applications and lines of business, which drives a high degree of resource utilization. This maximizes the use of assets on the grid, so you obtain an important efficiency benefit from a business perspective.

Performance is a key differentiator from other Hadoop distributions. The low-latency architecture and the sophisticated scheduling engine bring higher levels of performance and deeper analysis, which provide a business advantage. The scheduler allows you to do more with less infrastructure. Platform Symphony enables significant TCO reductions and better application performance.

In terms of flexibility, Platform Symphony offers heterogeneous application support by providing an open application architecture. This is based on the SOA framework, and allows compute-intensive and data-intensive applications to run simultaneously in the same cluster. The MapReduce framework in Platform Symphony is built on an open architecture that provides 100% Hadoop application compatibility for Java based MapReduce jobs. This Symphony MapReduce open architecture also provides methods for using multiple file system types and database architectures. With well-documented APIs, Platform Symphony

integrates well with various languages and development environments. All these contribute to reducing development costs and new application deployment time, saving time and improving the overall quality.

In an enterprise environment, where mission critical implementation is the rule, the reliability and high availability features are common ingredients. With the automatic failure recovery for all the workload management (service-oriented application middleware) and resource management (EGO) components, Platform Symphony fulfills these mandatory requirements for an enterprise product. All MapReduce framework services, including the HDFS daemons, are integrated within the automatic failure recovery functionality. They can be easily deployed in resilient and highly available configurations.

The scalability of an enterprise MapReduce grid computing environment is an important feature because business requirements usually grow. You must make sure that the continuously increasing workload does not saturate your grid. Platform Symphony has an extended customer base, with huge enterprise deployments in many cases. The product has grown and matured over many years, with new features and scalability limits being developed as responses to customer requirements for increasingly compute-intensive and data-intensive workloads.

The management tasks can be performed both from the web-based Platform Management console GUI, and from the command line. The GUI is a modern, web-based portal for management, monitoring, reporting, and troubleshooting purposes. It offers a high level of interactivity with the running jobs. For example, you can Suspend, Resume, and Kill jobs and tasks, and even change the priority of a running job.

Predictability is an important feature when you want to ensure that specific jobs will run according to an SLA. If you have 1000 jobs running at once, you might care about how all of them interact and are run so that you meet the SLA. You want to avoid the situation of a dominant job consuming all the resources, or many small jobs being stuck behind a large job. Platform Symphony, through its ownership and guaranteed allocation of a minimum number of resources to a consumer, handles this effectively. This drives predictability, which gives you the ability to manage and fulfill sophisticated SLA requirements when allocating and using resources.

In conclusion, shared services for big data are possible now. Hadoop is just another type of workload for a Platform Symphony grid. You do not have to deploy separate grids for different applications anymore. Platform Symphony MapReduce enables IT data centers running shared-service grids to achieve improved resource utilization, simplified infrastructure management, faster performance, and high availability. This provides reduced infrastructure and management costs, and an increased return on infrastructure investments.



IBM Platform Cluster Manager - Advanced Edition (PCM-AE) for technical cloud computing

This chapter describes aspects of PCM-AE related to technical cloud computing.

This chapter includes the following sections:

- ▶ Platform Cluster Manager - Advanced Edition capabilities and benefits
- ▶ Architecture and components
- ▶ PCM-AE managed clouds support
- ▶ PCM-AE: a cloud-oriented perspective

5.1 Overview

A characteristic of a cloud environment is that it allows users to serve themselves when it comes to resource provisioning. You can have, for example, a smart cloud entry environment to manage a set of IBM Power or x86 servers in such a way that users can create their own environment (logical partitions or virtual machines) to run their workloads. Clouds automate day-to-day tasks such as system deployment, leaving IT specialists time to perform more challenging work such as thinking about solutions to solve business problems in the company.

If resource provisioning can be done at the level of a single system, how about automating whole cluster deployments? Moreover, how about automating provisioning tasks for highly demanding and specialized cluster environments such as high performance computing (HPC)? How much time can you save your IT administrators team by having users trigger the deployment of HPC clusters dynamically? You can do all of this with the implementation of IBM Platform Cluster Manager - Advanced Edition (PCM-AE).

Deployment automation and a self-service consuming model are just a small part of the advantages that PCM-AE can provide you. For more information about all of PCM-AE capabilities along with the benefits you get from them, see 5.2, “Platform Cluster Manager - Advanced Edition capabilities and benefits” on page 90.

IBM Platform Cluster Manager - Advanced Edition is a product that manages the provisioning of complex HPC clusters and cluster resources in a self-service and flexible fashion.

This chapter is not intended to be a comprehensive description of PCM-AE. Rather, it focuses on PCM-AE aspects that are of interest for technical computing cloud environments. For information about how to implement a cloud environment managed with PCM-AE, see *IBM Platform Computing Solutions*, SG24-8073. For a detailed administration guide, see *Platform Cluster Manager Advanced Edition Administering*¹, SC27-4760-01.

5.2 Platform Cluster Manager - Advanced Edition capabilities and benefits

PCM-AE can greatly simplify the IT management of complex HPC clusters and be the foundation of building HPC environments as a service. The following are the capabilities and benefits of PCM-AE:

- Manage multi-tenancy HPC environments

PCM-AE can create and manage isolated HPC environments running on your server farm. You can use it to deploy multiple HPC workloads, each running a particular application and used by a different group of users as shown in Figure 5-1 on page 91. Each environment can be accounted for individually in terms of resource usage limits and resource usage reporting.

¹ <http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC27-4760-01>

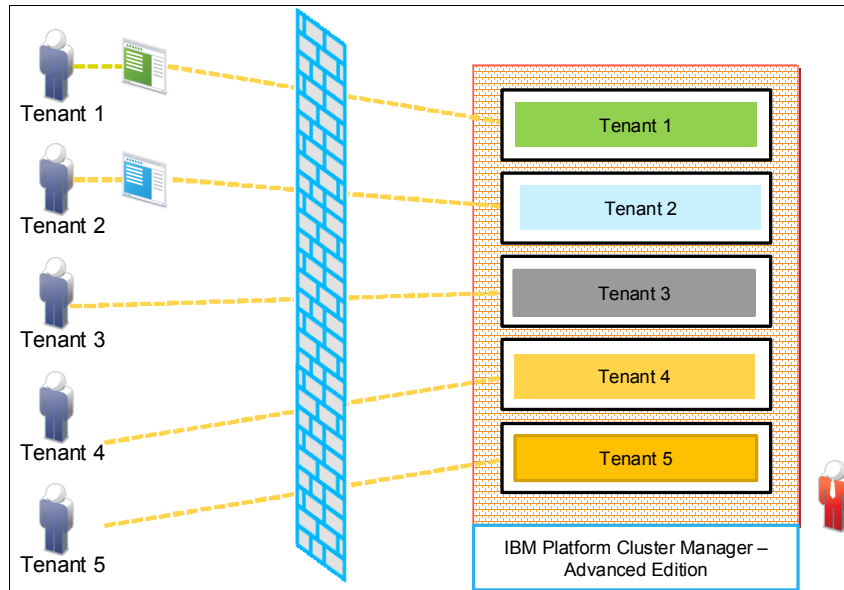


Figure 5-1 Multi-tenancy characteristics of IBM PCM-AE

► Support for multiple HPC products

PCM-AE can help manage IBM HPC products such as IBM Platform Symphony, IBM Platform LSF, and IBM InfoSphere BigInsights to name a few. PCM-AE also supports third-party products such as Grid Engine, PBS Pro, and Hadoop. These characteristics allow you to consolidate multiple workload types under the same hardware infrastructure as depicted in Figure 5-2.

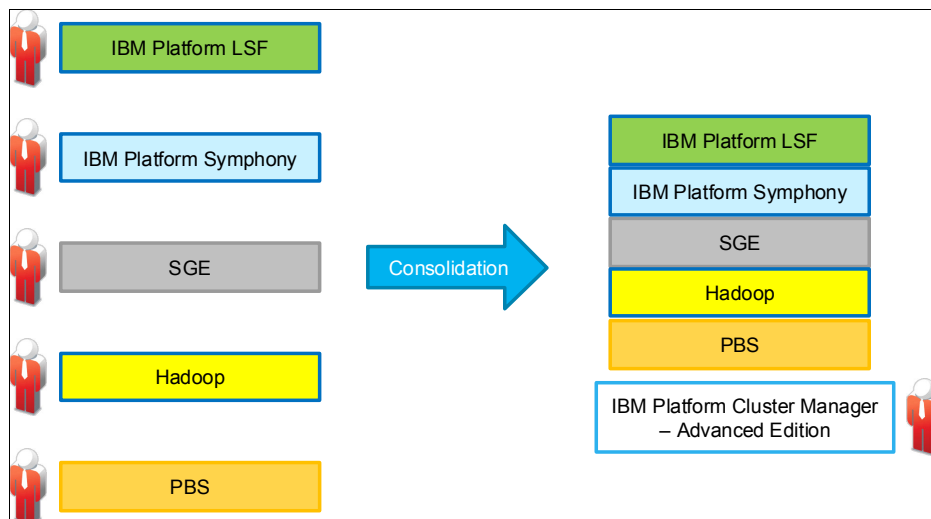


Figure 5-2 Consolidating multiple HPC clusters under a PCM-AE managed environment

► On-demand and self-service provisioning

Users can dynamically create clusters within PCM-AE based on cluster definitions that are published by PCM-AE cloud administrators. This process requires no deep understand of the cluster setup process. Also, there is little or no need for paperwork and approval flows to deploy a cluster because resource usage limits and policies are defined on a per tenant basis. Users can help themselves freely according to the rules and limits established for them. This helps reduce operational costs.

- Use of physical and virtual resources

You can choose whether to deploy an HPC cluster with underlying physical servers composing your cluster infrastructure, or you can provision on top of a virtualized layer, or use a mixed approach. Some larger servers with generous amounts of processor and memory can be good candidates for hosting clusters in a virtual fashion with the use of virtual machines. You can maximize the consolidation level of your infrastructure as a whole with virtualization, or you can truly isolate workloads by engaging only physical servers. Either way, PCM-AE can serve your goals.

- Increased server consolidation

Having multiple technical computing cluster infrastructures that are isolated as silos might result in a considerable amount of idle capacity when looking at overall resources. With PCM-AE, these silos' infrastructure can be managed in a centralized, integrated way that allows you to use these little islands of capacity that alone would not be sufficient for deploying a new cluster.

- Fast and automated provisioning

With a self-service based approach and predefined cluster definitions, all of the provisioning process can be automated. This reduces the deployment time from hours (or days) to minutes.

- Cluster scaling

As workloads increase or decrease, you can adjust the amount of resources that are assigned to your cluster dynamically as depicted in Figure 5-3 on page 93. This feature is commonly called *cluster flexing*, and can happen automatically based on utilization thresholds that you define for a particular cluster. This allows you to optimize your server utilization. For more information, see 5.5.3, "Cluster flexing" on page 100. Also, temporary cluster deployments or recurring cluster deployments that are based on scheduled workload processing are supported.

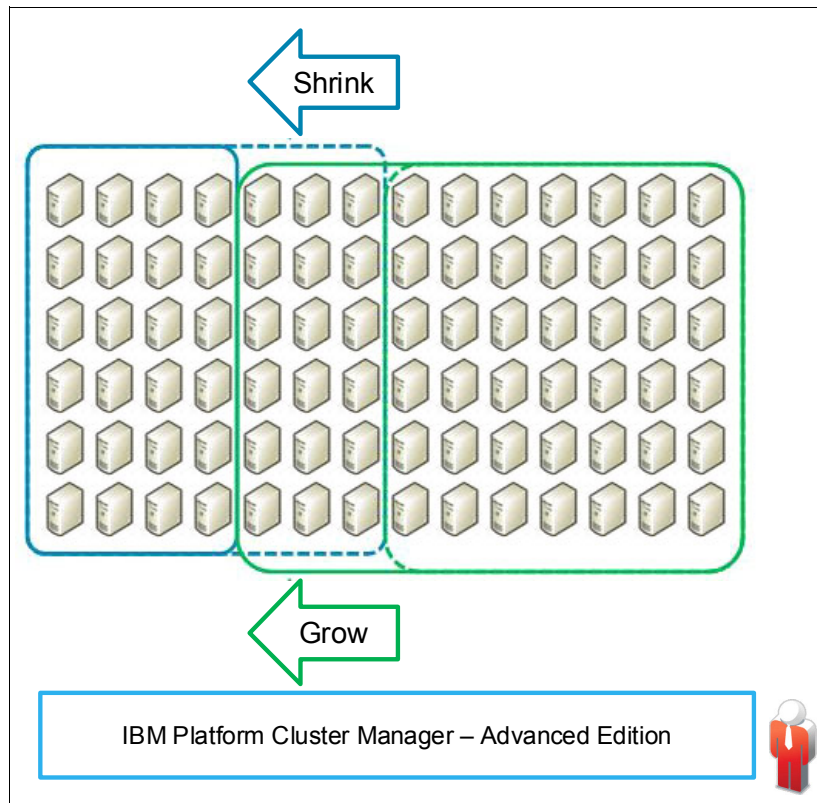


Figure 5-3 Dynamically resizing of clusters

► Shared HPC cloud services

This helps use other HPC cloud infrastructures to handle peak demands. If you face an unpredictable high resource utilization in your PCM-AE cloud environment and need to extend it, you can use IBM Smart Cloud public cloud infrastructure to add extra resources to your environment to meet your peak demand.

In addition, during low utilization periods, you can lease excess capacity to other entities. This is a good use case for universities that sparsely consume their HPC cloud resources, and is depicted in Figure 5-4. You can even have a completely isolated PCM-AE environment to serve external consumers.

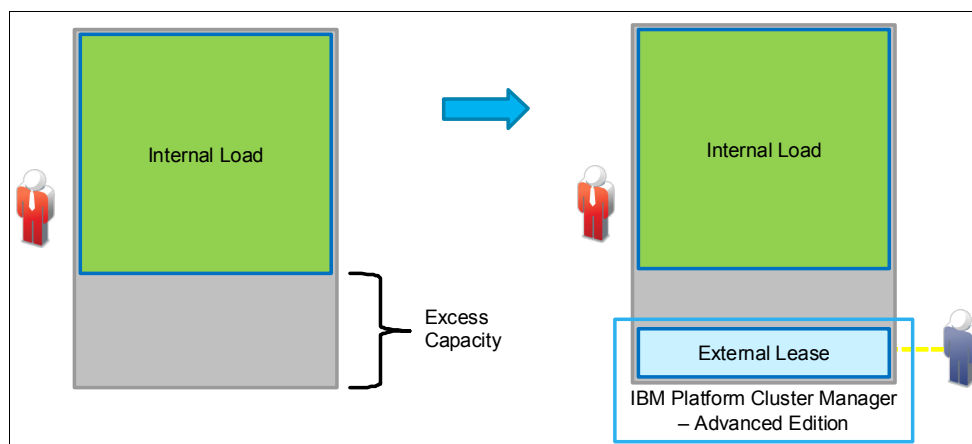


Figure 5-4 Leasing excess capacity with PCM-AE

5.3 Architecture and components

PCM-AE has its own particular internal software components architecture, and it uses other software components to create a manageable cloud infrastructure environment. Figure 5-5 depicts the hardware and software components of a PCM-AE environment. They can be classified into three distinct components: hardware, PCM-AE external software components, and PCM-AE internal software components.

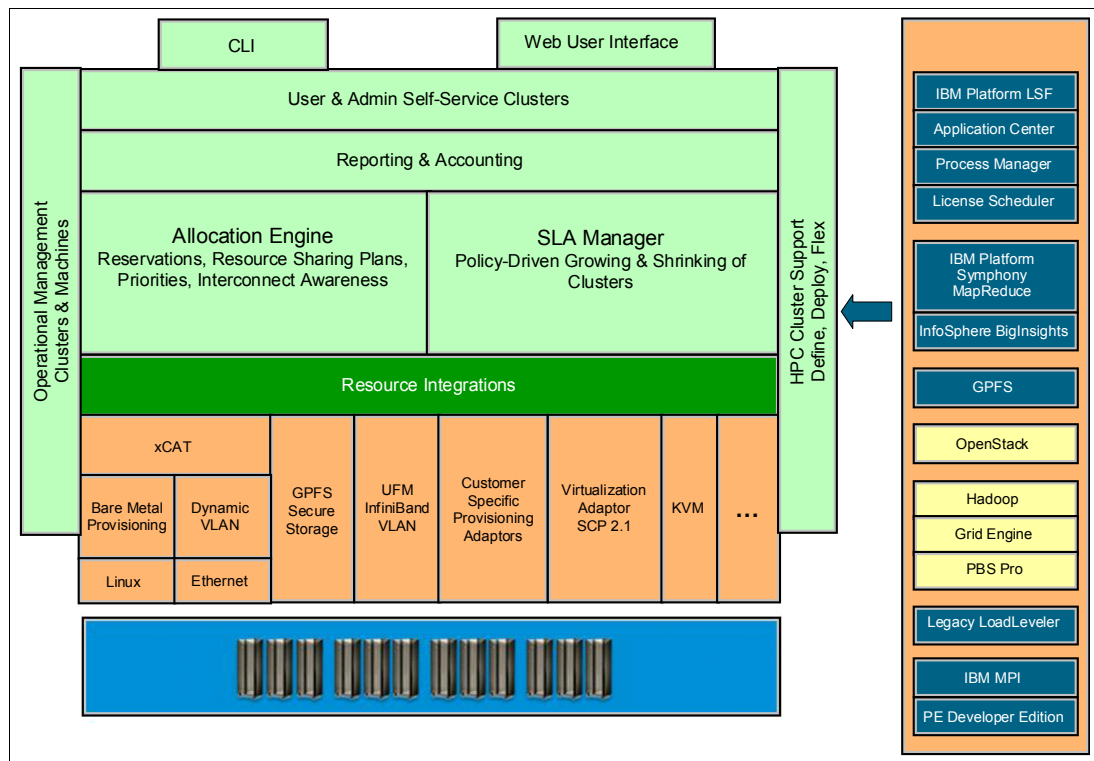


Figure 5-5 PCM-AE software components architecture

This section describes the architecture of Figure 5-5 in a bottom-up approach.

5.3.1 Hardware

The components with a blue background are the hardware: Servers, switches, and storage units. As explained in 5.2, “Platform Cluster Manager - Advanced Edition capabilities and benefits” on page 90, servers can be used as physical boxes or can be further virtualized by software at upper layers.

5.3.2 External software components

The components in orange are PCM-AE external components that can be used with PCM-AE to provide resource provisioning. They manage machine or virtual machine allocation, network definition, and storage area definition. As Figure 5-5 illustrates, you can use xCAT to perform bare metal provisioning of servers with Linux along with dynamic VLAN configuration. Also, KVM is an option as a hypervisor host managed by PCM-AE.

Another external software component, IBM General Parallel File System (GPFS), can be used to manage storage area network disks that can be used, for example, to host virtual

machines that are created within the environment. GPFS plays an important role when it comes to Technical Computing clouds because of its capabilities as explained in 6.1, “Overview” on page 112. Parallel access to data with good performance makes GPFS a good file system for many HPC applications. Also, the ability of GPFS to access remote network shared disks by using the NSD protocol provides flexibility in the management of the cloud.

PCM-AE can also manage and provision clouds using the Unified Fabric Manager platform with InfiniBand. PCM-AE can be used to provide a private network among the servers while maintaining multi-tenant cluster isolation by using VLANs.

PCM-AE is a platform that can offer support for the integration of other provisioning software, including eventual custom adapters that you might already have or need in your existing environment.

5.3.3 Internal software components

The components in green in Figure 5-5 on page 94 are PCM-AE’s internal software components.

Besides the resource integrations layer that allows PCM-AE to use external software components, PCM-AE includes these software components:

- ▶ Software to control the multi-tenancy characteristic of a PCM-AE based cloud (user accounts)
- ▶ A service level agreement component that defines the rules for dynamic cluster growth or shrinking
- ▶ Allocation engine software that manages resource plans, prioritization and how the hardware pieces are interconnected
- ▶ Accounting and reporting software to provide feedback on cluster utilization and resources consumed by tenants
- ▶ Software that handles operational management of the clouds (cluster support and operational management). This allows you to define, deploy and modify clusters, and also to visualize existing clusters and the servers within them.

5.4 PCM-AE managed clouds support

PCM-AE supports many computing and analytics workloads. If the cloud hardware and the provisioned cluster operating system can handle the workload requirements, you can have your workload managed by PCM-AE.

Table 5-1 shows a list of supported hardware and operating systems for cluster deployment in a PCM-AE managed cloud.

Table 5-1 PCM-AE hardware and software support, and environment provisioning support

Infrastructure hardware and software support	
Hardware support	<ul style="list-style-type: none"> ▶ IBM System x iDataPlex ▶ IBM Intelligent Cluster™ ▶ Other rack-based servers ▶ Non-IBM x86-64 servers
Supported provisioning software for bare-metal servers	<ul style="list-style-type: none"> ▶ xCAT 2.7.6 ▶ IBM support for xCAT V2 (suggested)

Infrastructure hardware and software support	
Supported provisioning software for virtual servers	<ul style="list-style-type: none"> ▶ KVM on RHEL 6.3 (x86 64-bit) ▶ IBM SmartCloud Provisioning 2.1 ▶ vSphere 5.0 with ESXi 5.0
Network infrastructure support	<ul style="list-style-type: none"> ▶ IBM RackSwitch G8000, G8124, G8264 ▶ Mellanox InfiniBand Switch System IS5030, SX6036, SX6512 ▶ Cisco Catalyst 2960 and 3750 switches
PCM-AE Master node OS and software requirements	<ul style="list-style-type: none"> ▶ RHEL 6.3 (x86 64-bit) ▶ MySQL, stand-alone 5.1.64 or Oracle 11g Release 2 or Oracle 11g XE
Environment provisioning support	
Bare-metal-provisioned systems (by xCAT)	<ul style="list-style-type: none"> ▶ RHEL 6.3 (x86 64-bit) ▶ KVM on RHEL 6.3 (x86 64-bit) ▶ CentOS 5.8 (x86 64-bit)
Virtually provisioned systems (by KVM, SmartCloud, vSphere)	<ul style="list-style-type: none"> ▶ RHEL 6.3 (x86 64-bit) ▶ Microsoft Windows 2008 (64-bit)
Provisioned storage clients	<ul style="list-style-type: none"> ▶ IBM GPFS V3.5 client node

PCM-AE supports most of today's workload managers, including:

- ▶ IBM Platform LSF 8.3, or later
- ▶ IBM Platform Application Center 8.3
- ▶ IBM Platform Symphony 5.2, or later
- ▶ IBM InfoSphere BigInsights 1.4, or later

5.5 PCM-AE: a cloud-oriented perspective

This section provides an overview of some of PCM-AE's common operations such as cluster definition, cluster deployment, cluster flexing, and cluster metrics. This section is intended to present and demonstrate the product from a practical user point of view. It is not a comprehensive user administration guide.

5.5.1 Cluster definition

PCM-AE cluster provisioning is based on a self-service approach. This means that a BigData consumer can create a cluster and start using its analytics application for report generation. This particular consumer knows how to operate its analytics application, but might not have any idea of how to build the required cluster infrastructure for the application.

To solve these challenges and ensure that tenants are able to allocate clusters in a self-service, infrastructure-knowledge-less manner, PCM-AE isolates the complexity of cluster internals from the tenants. So, as a tenant, you can only create clusters for which there is a published cluster definition inside your PCM-AE environment. If it has a published IBM InfoSphere BigInsights cluster definition, you can create a BigInsights cluster for your use and access BigInsights directly. If you need to run an ANSYS workload but your PCM-AE environment has no definition for this type of workload, the PCM-AE administrator must first create a cluster definition before PCM-AE can provision it for you.

A cluster definition is an information set that establishes these settings:

- ▶ Which types of nodes compose the cluster (master, compute, custom nodes)
- ▶ Which operating system is to be installed on each type of node
- ▶ Minimal and maximum amount of resources allowed (processor and memory, number of nodes)
- ▶ How to assign IP addresses
- ▶ Pre and post-install scripts
- ▶ Custom user variables to guide the scripts (for instance, where to get the applications from for installation, or in which directory to install them)
- ▶ Storage placement
- ▶ Other information that is pertinent to the provisioning of the cluster itself.

Figure 5-6 shows a definition for an IBM InfoSphere BigInsights 2.1 with a Platform Symphony 6.1 cluster.



Figure 5-6 Definition for an IBM BigInsights 2.1 and Platform Symphony 6.1 cluster

Figure 5-6 shows that this cluster definition is composed of two node types: Master and compute nodes. A BigInsights cluster has one master node and can have multiple compute nodes for processing. Therefore, the administrator creates a cluster definition with both of these node types. As you can also see in Figure 5-6, the operating system to be used on the master node (its selection is highlighted on the picture) is Red Hat Enterprise Linux 6.2.

In this particular cluster definition, the administrator used post-installation scripts to install IBM InfoSphere BigInsights (layer 2, master setup and compute setup steps). After its installation was completed on both nodes and synchronized (SignalComplete and WaitForSignal, which are custom scripts based on sleep-verify checks), the installation of

Platform Symphony took place (SymphonyMaster and SymphonyCompute scripts). Finally, the integration scripts were called.

Figure 5-7 shows the custom script that was used for installing BigInsights on the master node in the cluster definition example.

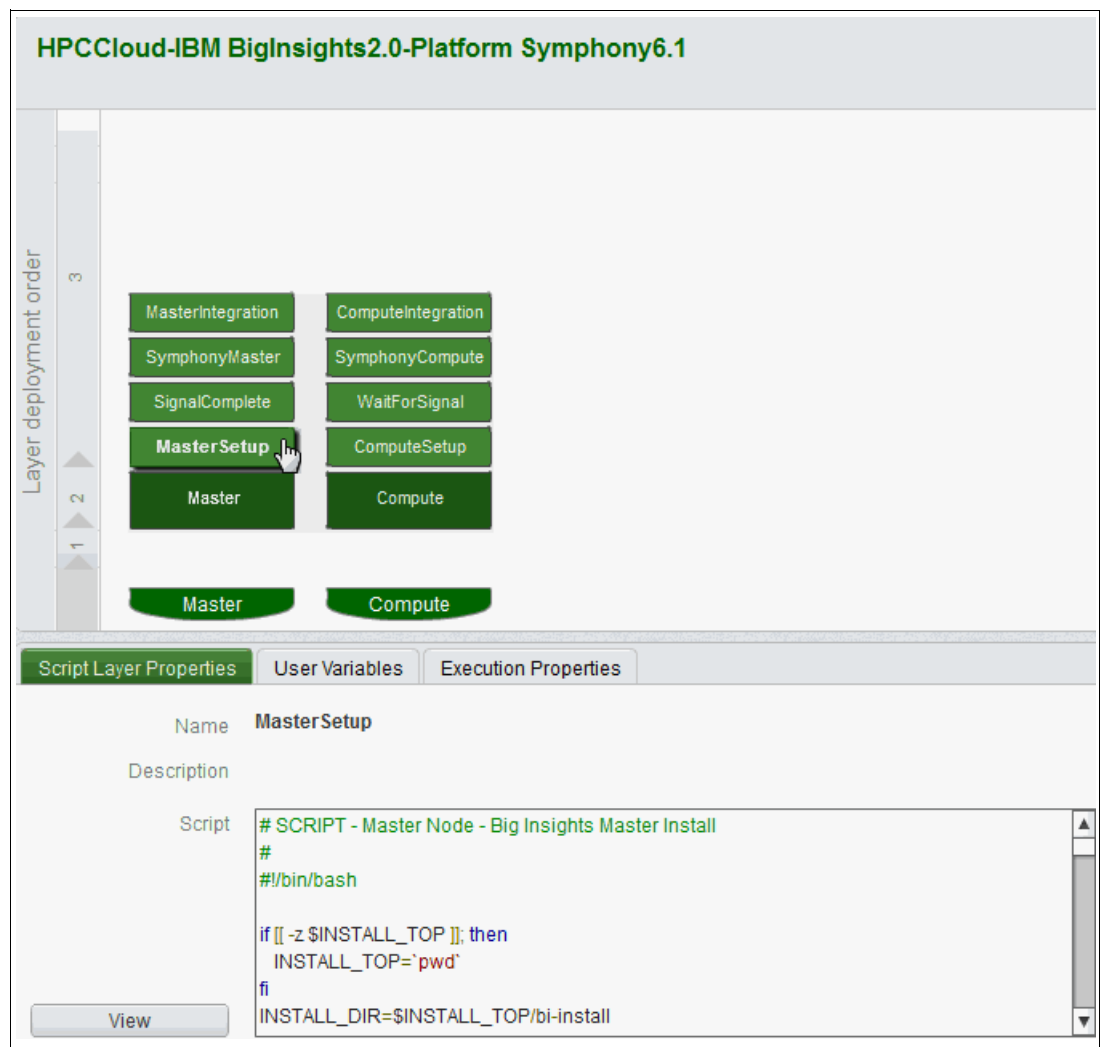


Figure 5-7 Example of a custom script in a cluster definition inside PCM-AE

PCM-AE cluster definitions give you a flexible way to define a thorough provisioning of your cluster. If your cluster software installation and software setup can be scripted, you can probably manage its deployment with PCM-AE.

5.5.2 Cluster deployment

After a cluster definition is available, users can self-service themselves by instantiating a cluster based on it. The process is as simple as going through a guided wizard, selecting the number of compute nodes (or other custom node types depending on the cluster definition), and how much memory and processor to assign to the cluster.

Figure 5-8 shows the initial cluster provisioning steps in PCM-AE.

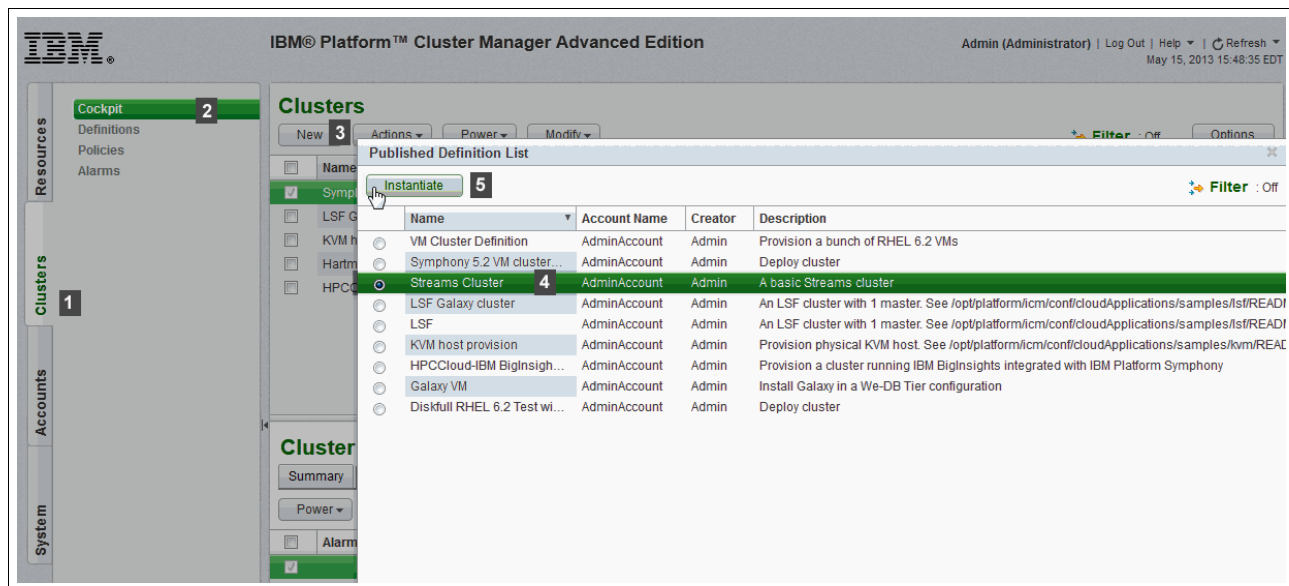


Figure 5-8 Creating a cluster inside PCM-AE

The numbers in Figure 5-8 denote the sequence of actions to provision a cluster with PCM-AE:

1. Go to the Clusters view in PCM-AE's navigation area.
2. Select the Cockpit area to visualize the active clusters.
3. Click **New** to start the cluster creation wizard.
4. Select a cluster definition for your cluster. In this example, create a streams cluster from the definition on the list of published cluster definitions.
5. Click **Instantiate** to go to the next and last step of the creation wizard.

After you complete these steps, select the amount of resources for your cluster as shown in Figure 5-9.

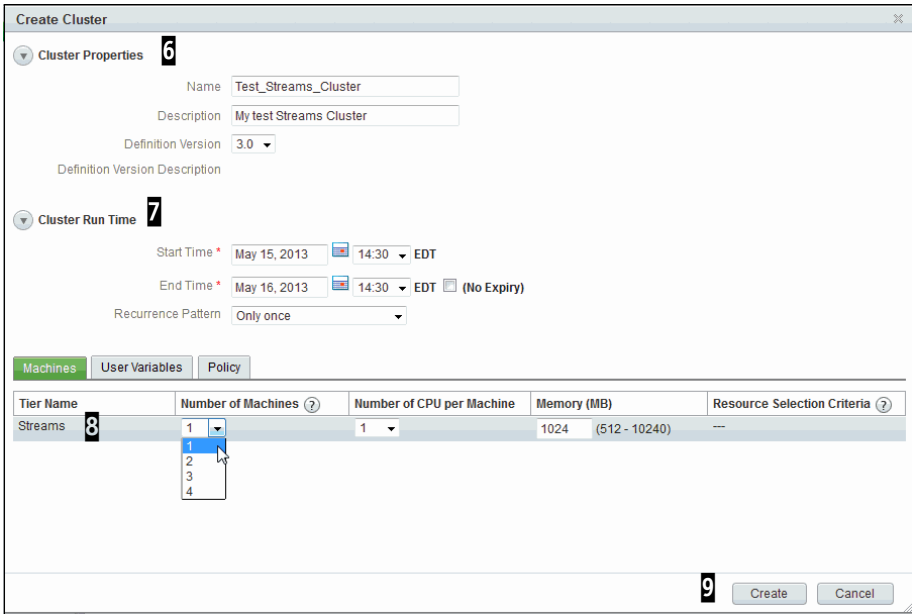


Figure 5-9 Resource assignment in cluster creation wizard

6. Edit the cluster properties as needed such as the cluster name and description, and select which definition version to use².
7. Define the cluster lifetime. You can provision the cluster for a limited amount of time (start - end), make it a recurrent provisioning (suits workloads that happen punctually but in a repetitive schedule base), or make it a lifetime cluster (no expiration).
8. Select the resources to assign to your new cluster. In this streams cluster, the cluster definition allows you to select the number of nodes, and the amount of processor and memory per node.
9. Click **Create**, and the wizard starts provisioning the new cluster.

This process is straightforward and does not require any specific knowledge of cluster infrastructure.

5.5.3 Cluster flexing

After a cluster is up and running, it might not always be at its peak utilization. Computations have their minimums, averages, and maximums. PCM-AE provides *cluster flexing* to make better use of the cloud infrastructure for technical computing.

Clusters can be flexed up (node addition) or down (node removal) as long as the current number of nodes obey the maximum and minimum node numbers in the cluster definition used with your particular cluster.

² Cluster definitions can use version tracking as the administrators make changes, enhance, or provide different architecture definitions for the same cluster type.

There are two ways to perform cluster flexing: Manually or automatically. A manual cluster flexing is done through PCM-AE's **Clusters** → **Cockpit** view in the main navigation area by selecting **Modify** → **Add or Remove Machines** as depicted in Figure 5-10.

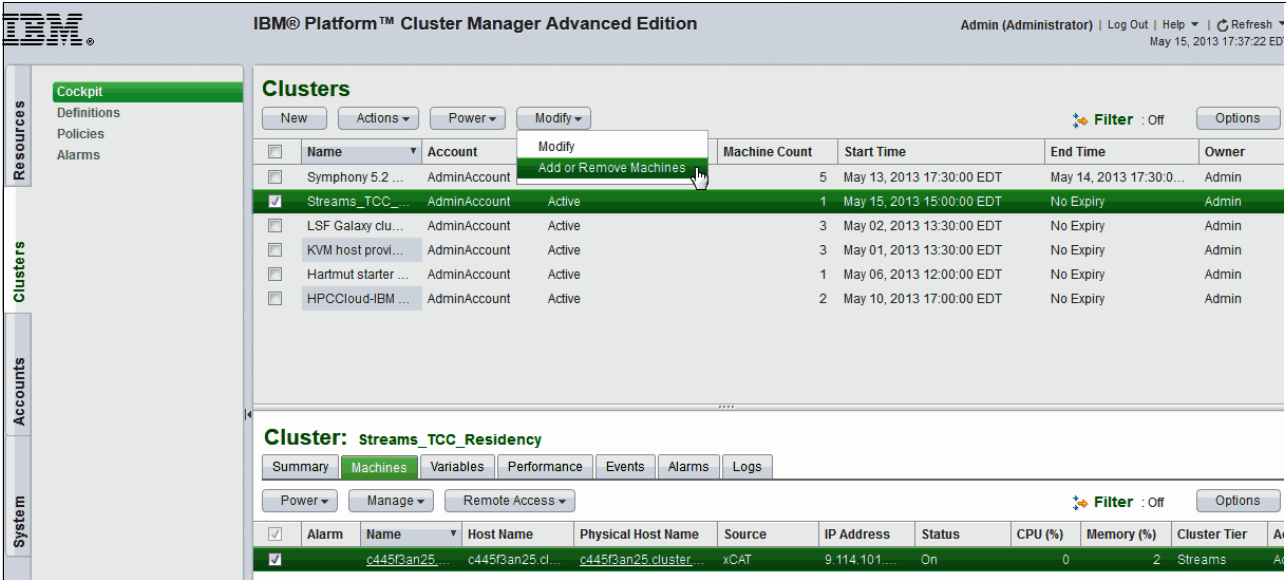


Figure 5-10 IBM PCM-AE main navigation window

Then, change the number of on-demand nodes as shown in Figure 5-11.

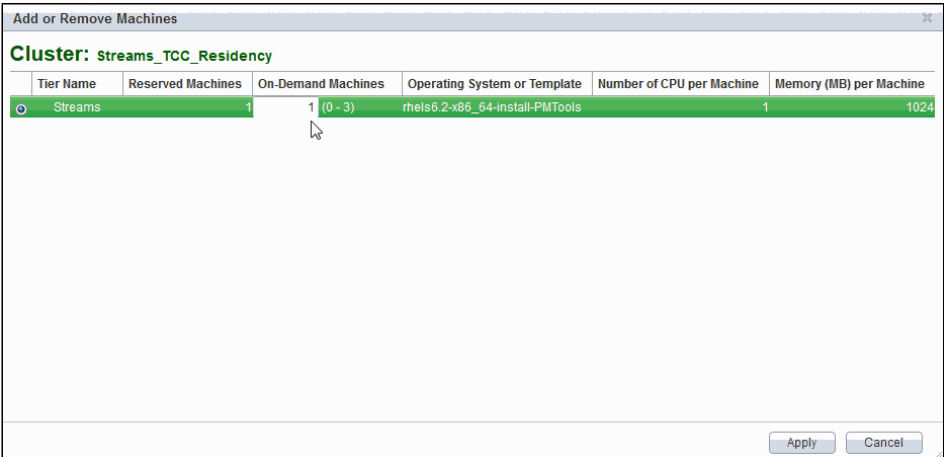


Figure 5-11 Flexing of an active cluster in PCM-AE

The cluster definition used in this example imposes a minimum number of one node and a maximum number of four nodes in the cluster. Therefore, you can change the “on-demand” nodes from 0 - 3.

It is also possible to configure PCM-AE to flex the cluster based on thresholds that you define. A cluster policy must be in place for that to happen. Policies can be predefined in cluster definitions, or can be created later.

Figure 5-12 shows a streams cluster definition with flexing threshold policies. In this case, the cluster is flexed up by one node (action parameters) if average processor utilization reaches 95%, or is flexed down by one node if average processor utilization gets below 15%. It is possible to send notification emails when cluster auto-flexing occurs.

Cluster Designer

Streams Cluster

Prebuild Elements

Layer deployment order

3

Post-nata script

Install Streams

Streams Prereqs

Users and Groups

Mount NFS

2

RHEL

1

Pre-nata script

Streams

Machine

Pre-nata script

Tier_2

Machine

Pre-nata script

Tier_3

Machine

Pre-nata script

Tier_4

Machine

Pre-nata script

Tier_5

Cluster Definition Properties

User Variables

Deployment Variables

Policy

Restrictions

Secure Network

Status

Enabled

Run Interval

10

minutes

Notification Email

rceron@br.ibm.com

Add

Delete

<input type="checkbox"/>	Target	Rule Name	Metric	Operator	Value	Action	Action Parameters	Stabilizat
<input type="checkbox"/>	Streams	FlexUp	AVG CPU U...	>=	95	None	1	1
<input checked="" type="checkbox"/>	Streams	FlexDown	AVG CPU U...	<=	15	None	1	1

Figure 5-12 Flexing threshold policy definition in cluster definition

Figure 5-13 shows the graphs of a simple one-node cluster that is deployed with the cluster definition from Figure 5-12 on page 102. After its creation, the processor workload is increased on the node. When it reaches the 95% threshold, it flexes up. Notice that the second node gets accounted for in the statistics at the beginning of the flex up process, but is only really available for processing (and thus appearing as ready) after its whole deployment process is complete. At the end, take the load out and the cluster flexes itself down back to one node only. The flex down happens much more quickly as you can verify in Figure 5-13.

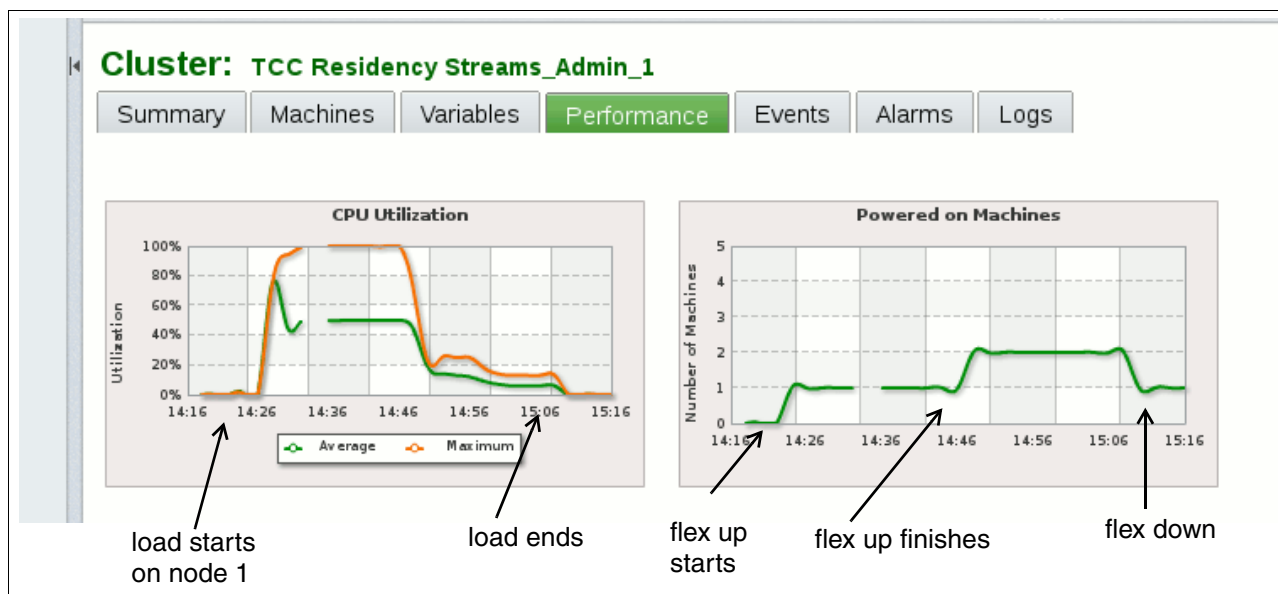


Figure 5-13 Example of automatic cluster flexing

5.5.4 Users and accounts

To manage a self-service cloud environment for Platform Computing, HPC, and analytics clusters, PCM-AE offers mechanisms to control user access and define accounts.

Users

A user in PCM-AE is the entity that represents a person who logs in to the environment. The user holds information for user authentication and other ordinary data such as email, phone number, department name, location, first and surnames, and business unit. The operations that a user can perform in a PCM-AE environment are based on its classification. Users can be classified as administrators, account owners, or account users. Each user type is granted permission to perform a set of operations. These permissions can be customized.

To understand more about or to customize the permissions that are granted to user types, check PCM-AE's access control description in its **System** → **Access Control** view as depicted in Figure 5-14.

IBM® Platform™ Cluster Manager Advanced Edition (Administrator) | Log Out | Help | Refresh
May 17, 2013 11:44:37 EDT

Access Control
The following are the permission settings for the Administrator, Account Owner, and Account User roles.
[Glossary of Permissions](#)

System Permissions	Roles		
	Administrator	Account Owner	Account User
Create Account	<input checked="" type="checkbox"/>	n/a	n/a
Account Permissions			
Control Sub Account	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	n/a
View Account	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Modify Account	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Assign or Unassign Users	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Control Account	<input checked="" type="checkbox"/>	n/a	n/a
Delete Account	<input checked="" type="checkbox"/>	n/a	n/a
Set Resource Limit	<input checked="" type="checkbox"/>	n/a	n/a
Create Cluster Definition	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	n/a
Create Cluster	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View All Cluster	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Assign IP Pool	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	n/a
Assign Storage	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	n/a
Cluster Definition Permissions			
View Cluster Definition	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Modify Cluster Definition	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	n/a
Publish or Unpublish Definition to Own Account	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	n/a
Delete Cluster Definition	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	n/a
Assign Accounts to Cluster Definition	<input checked="" type="checkbox"/>	n/a	n/a

Figure 5-14 PCM-AE user and account management permissions

The default permissions are the less restrictive set of permissions. This helps the self-service approach of cloud use described throughout this book.

PCM-AE can be integrated with Lightweight Directory Access Protocol (LDAP) user databases and with Windows Active Directory databases for authentication. For more information, see the *Platform Cluster Manager Advanced Edition Administering* guide, SC27-4760-01 at:

<http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC27-4760-01>

Accounts

Accounts are an administrative layer on top of the user layer. Users can be grouped in an account. Whichever resources and resource limits the account grants to its users, all of the users under that account are subject to these definitions. This is how you can control your PCM-AE cloud environment with a multi-tenant approach. A company's departments can be accounts. Also, a university department can be an account. This not only isolate users into groups, but also allows the cloud administrator to provide different service levels to different accounts.

Hint: All of the resources that a cluster uses are reported and tracked against the account of the user who created it.

In PCM-AE's **Accounts** view, you can view existing accounts, view cluster definitions available for use of an account, view clusters along with its systems or virtual machines deployed by users of a particular account, and trigger resource consumption by clusters within an account. Figure 5-15 shows which clusters are deployed under the PCM-AE's default account "SampleAccount".

Name	Account	Status	Machine Co...	Start Time	End Time
VM Cluster...	SampleAc...	Expired	4 (2 on Stan...	May 16, 2013 11:00:00 EDT	May 17, 2013 09:30:0...

Alarm	Name	Host Name	Physical Host Name	Source	IP Address	Status	CPU (%)	Memory (%)
<input checked="" type="checkbox"/>	c445vm12	c445vm12.clu...	c445f3an05.clu...	KVM	9.114.101....	Off	0	
<input type="checkbox"/>	c445vm11	c445vm11.clu...	c445f3an26.clu...	KVM	9.114.101....	Off	1	
<input type="checkbox"/>	c445vm10	c445vm10.clu...	c445f3an26.clu...	KVM	9.114.101....	Off	1	
<input type="checkbox"/>	c445vm09	c445vm09.clu...	c445f3an05.clu...	KVM	9.114.101....	Off	0	

Figure 5-15 Clusters that are deployed under a particular account in PCM-AE

5.5.5 Cluster metrics

As with any cloud environment, administrators must know how much of the resources are in use. This allows for capacity planning of the cloud. Also, it helps you decide whether to reach a peak resource consumption, or to go beyond and temporarily expand the cloud capacity by using a shared HPC cloud service as explained in 5.2, "Platform Cluster Manager - Advanced Edition capabilities and benefits" on page 90.

In addition, cloud administrators might want to identify how much of the resources of the cloud are in use or have been consumed by a particular cluster, or by a particular tenant who might own several clusters. Whether the technical computing cloud environment is private or public, this data is useful for charging tenants. Charging is a somewhat obvious concept in a public cloud because the consumers are all external. However, private clouds might also benefit if, for example, the IT department provides cloud infrastructure services for multiple departments within the company or institution. You might want to charge these internal consumers, allowing the IT department to provide the tenants difference service levels that are based on the charges.

Reports can be generated by using the **System** → **Reports** view inside of PCM-AE as illustrated in Figure 5-16.

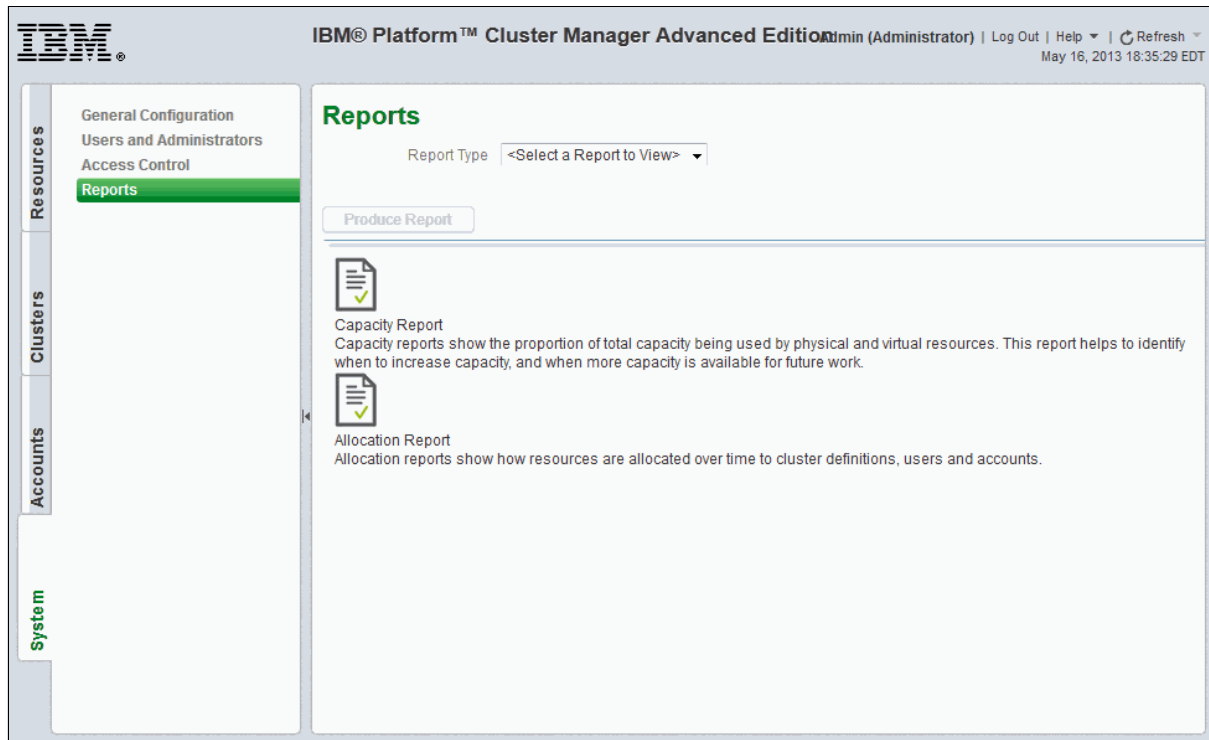


Figure 5-16 Accessing the reports view in PCM-AE

Figure 5-17 shows a capacity report for PCM-AE managed hardware. It shows the data by resource groups. This example shows a resource group of physical systems managed by xCAT, and another for virtual machines managed by KVM. The results within each resource group show the percentage of the total capacity that was used during the report period. It is possible to create custom reports for specific resource groups and for specific date ranges.

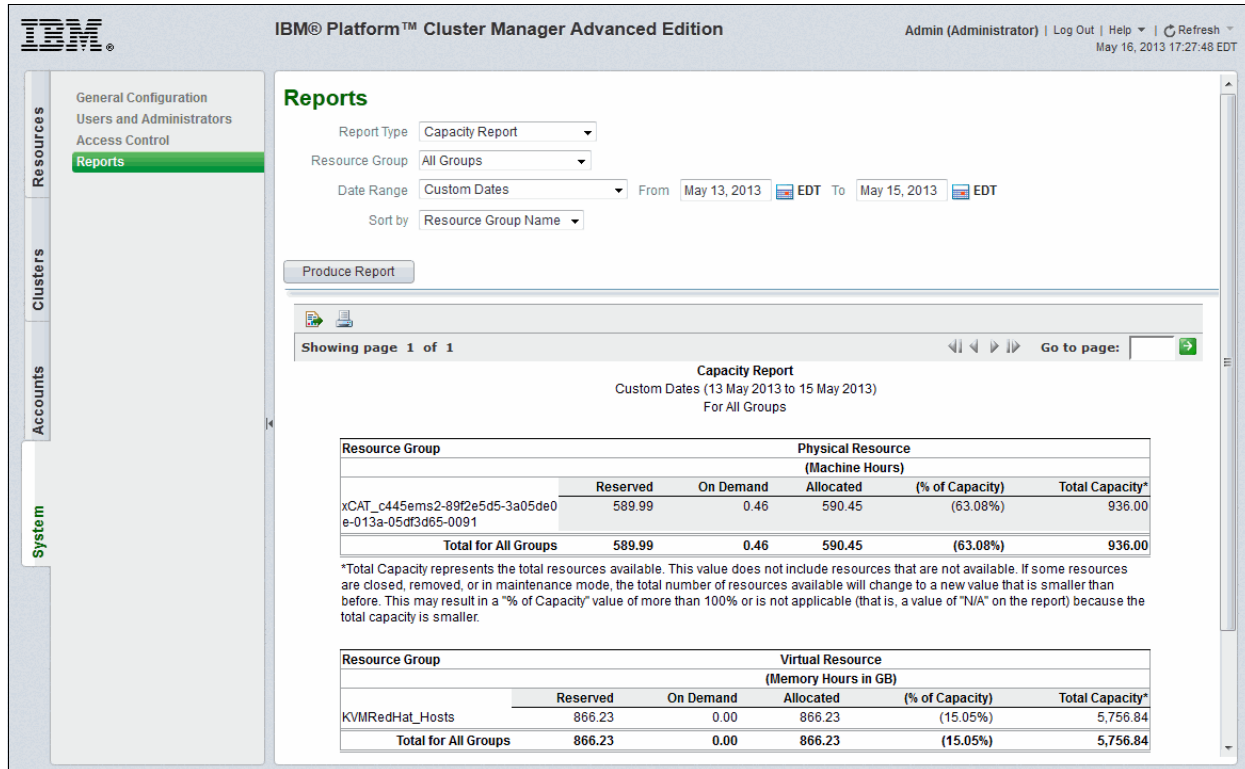


Figure 5-17 PCM-AE cloud capacity report

If you need a different reporting view, you can use a cloud resource allocation by cluster report example as shown in Figure 5-18. The view shows the number of hours each resource group was allocated by each cluster. On-demand hours also are displayed in the report.

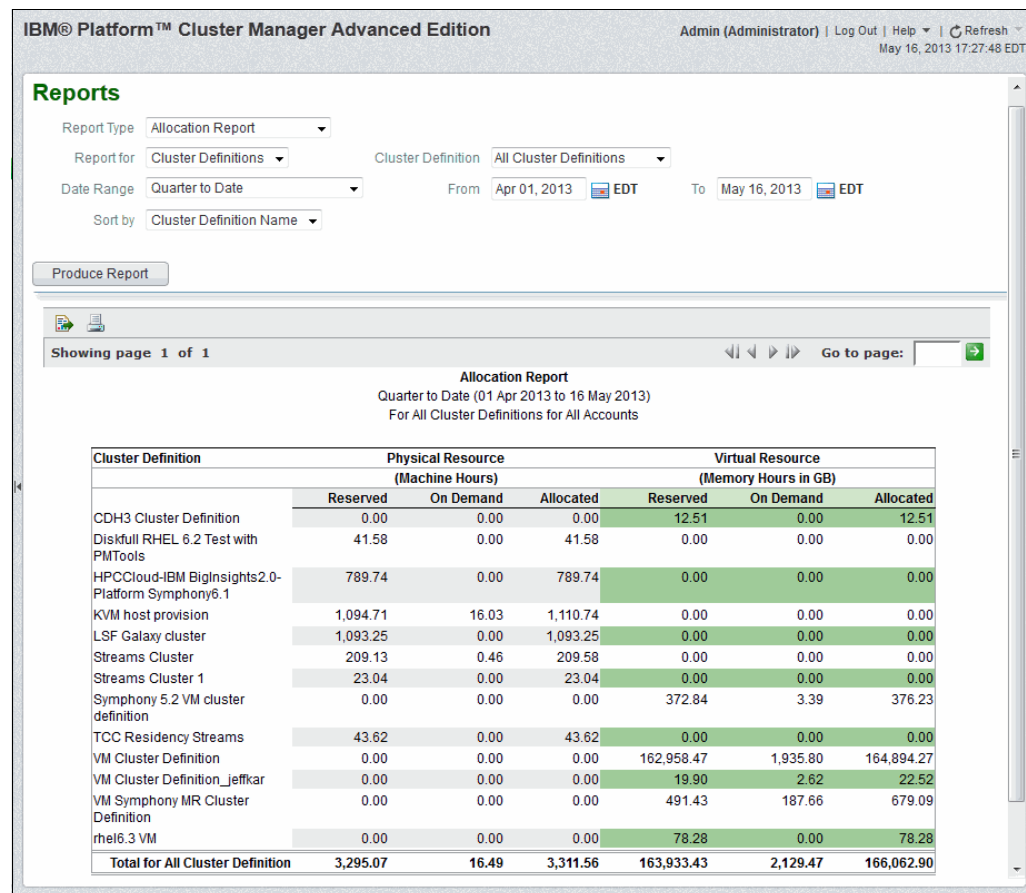


Figure 5-18 PCM-AE resource allocation report by cluster

In addition, reports based on single users or group accounts are available. Figure 5-19 depicts a user consumption report of the example PCM-AE cloud. This information can be used to account for tenants' use of the cloud in both public and private clouds.

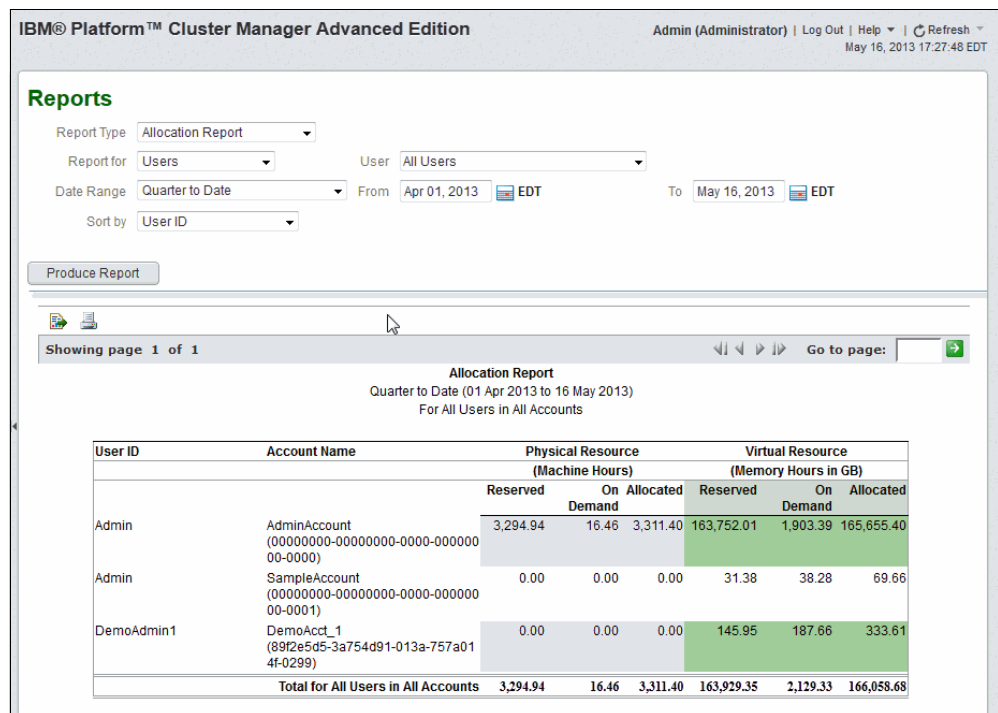


Figure 5-19 PCM-AE resource allocation report by user or account

So far only metrics that are useful from a cloud administration point of view have been covered. However, it is also possible to measure a cluster's instantaneous processor and memory usage, and check the number of active nodes over time (good for checking cluster flexing actions that are taken upon the cluster). This can be done by verifying the cluster Performance tab in the **Clusters** → **Cockpit** view as seen in Figure 5-20.

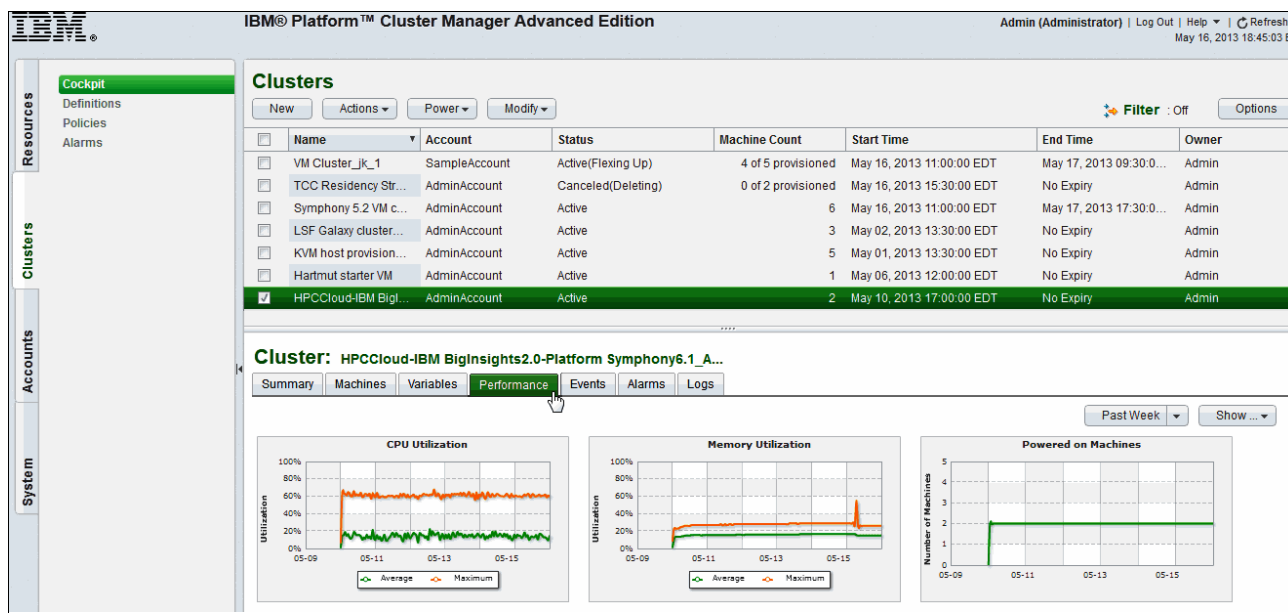



Figure 5-20 Cluster performance metrics: Processor, memory utilization, and number of active nodes



The IBM General Parallel File System for technical cloud computing

This chapter describes technical aspects of the IBM General Parallel File System (GPFS) that make it a good file system choice for technical computing clouds.

This chapter includes the following sections:

- ▶ Overview
- ▶ GPFS layouts for technical computing
- ▶ Integration with IBM Platform Computing products
- ▶ GPFS features for Technical Computing

This chapter also introduces the following new GPFS functions:

- ▶ Active File Management (AFM)
- ▶ File Placement Optimizer (FPO)

6.1 Overview

The IBM General Parallel File System (GPFS) has many characteristics that make it a good choice for the file system within technical computing environments and technical computing clouds. GPFS helps ease the management of the file system, ensure performance, and helps ensure concurrent access to data. The following is a list of the GPFS characteristics:

- ▶ High capacity
- ▶ High performance
- ▶ High availability
- ▶ Single-system image
- ▶ Multiple operating system and server architecture support
- ▶ Parallel data access
- ▶ Clustering of nodes
- ▶ Shared disks architecture

Each one of these characteristics is described in the following sections.

6.1.1 High capacity

GPFS is able to span a larger number of disks than other conventional file systems. This means that the final file system size that you can reach with it is larger. When you consider a technical computing environment for data intensive workloads, that extra space can make a difference. When you consider a cloud environment where you can deploy multiple technical computing environment workloads, the ability scale up and easily grow the file system size becomes important.

Currently, the maximum size a GPFS file system can reach is 512 XB (xonabytes), which is equal to 2^{99} bytes. The maximum number of files that can be stored in a single GPFS file system is 2^{63} files. That is 8 millions of trillions of files. Regarding disk size, the limit is imposed solely by the disk and operating system device drivers.

6.1.2 High performance

GPFS is built to provide performance to file access. I/O operations to a file in technical computing environments must happen efficiently. This is because files can be large, or many of them must be accessed at a time. To accomplish this, GPFS uses a few techniques to store the data:

- ▶ Large block size and full-stride I/O
- ▶ Wide striping: Spread a file over all disks
- ▶ Parallel multi-node I/O
- ▶ I/O multithreading
- ▶ Intelligent pre-fetch and file caching algorithms
- ▶ Access pattern optimization

6.1.3 High availability

Large technical computing environments and clouds must be highly available and eliminate single points of failure. GPFS embeds this characteristic by keeping metadata and logs, ensuring these are always available to the participating cluster nodes (for example, log replication), and by applying disk lease mechanisms (heartbeat).

If a node in a GPFS cluster fails, the log of the failed node can be replayed on the file system manager node.

Recoverability from failure situations is not the only way GPFS ensures high availability. High availability is also obtained with a correct disk access layout. Multiple GPFS server nodes can access the storage LUNs and serve the data. Also, the servers themselves access the LUNs through redundant paths. Moreover, each of the GPFS server nodes can access all of the LUNs. A clearer definition and understanding of high availability file system layout point of view can be found in 6.2, “GPFS layouts for technical computing” on page 115.

In addition, GPFS offers management capabilities that do not require downtime. These include the addition or removal of disks, and addition or removal of nodes in the GPFS cluster, which all can be performed without having to unmount the target file system.

6.1.4 Single system image

A GPFS file system is different from, say, an NFS one, as it can be seen as the same file system on each of the participating nodes. With NFS, the file system belongs to the node that serves that file system, and other nodes connect to it. In GPFS, there is no concept of a file system-owning node within the same GPFS cluster. All of the cluster nodes have global access to data on the file system.

It is also possible to create cross-cluster-mount scenarios where one GPFS cluster owns the file system and serves it to another remote cluster over the network. This is described in 6.2, “GPFS layouts for technical computing” on page 115.

This characteristic of GPFS simplifies operations and management, and allows nodes in different geographical locations to see the same file system and data, which is a benefit for a technical computing cloud environment.

6.1.5 Multiple operating system and server architecture support

The GPFS file system can be accessed by a multitude of operating systems. The supported operating systems for GPFS version 3.5 TL3 are shown in Table 6-1. Earlier releases of GPFS support older operating system versions. For an up-to-date list of supported operating systems, see:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=%2Fcom.ibm.cluster.gpfs.doc%2Fgpfs_faqs%2Fgpfsclustersfaq.html

Table 6-1 GPFS 3.5 TL3 operating systems support

Operating system	Versions
IBM AIX	6.1, 7.1
RedHat Enterprise Linux	5, 6
SUSE Linux Enterprise Server	10, 11
Microsoft Windows 2007 Server	Enterprise and Ultimate editions
Microsoft Windows 2008 Server	R1 (SP2), R2

Also, the following multiple server architectures are supported:

- ▶ IBM POWER
- ▶ x86 architecture
 - Intel EM64T
 - AMD Opteron
 - Intel Pentium 3 or newer processor
- ▶ Intel Itanium 2

This broad support allows the creation of technical computing environments that use different hardware server architectures (POWER, Intel-based servers, virtual machines) besides support for multiple operating systems. Therefore, GPFS is positioned to be the common file system of a multi-hardware, multi-purpose technical computing cloud environment.

6.1.6 Parallel data access

File systems are traditionally able to allow safe, concurrent access to a file, but it usually happens in a sequential form, that is, one task or one I/O request at a time.

GPFS uses a much more fine-grained approach that is based on byte range locking, a mechanism that is facilitated by the use of tokens. An I/O operation can access a byte range within a file if it holds a token for that byte range. By organizing data access in byte ranges and using tokens to grant access to it, different applications can access the same file at the same time, if they aim to access a different byte range of that file. This increases parallelism while still ensuring data integrity. The management of the data in byte ranges is transparent to the user or application.

Also, data placement on disks, multiple cluster node access to disks, and data striping and striding strategies allow GPFS to access different chunks of the data in parallel.

6.1.7 Clustering of nodes

GPFS supports clustering of nodes up to 8192 nodes. This benefits technical computing clouds that can be composed of thousands of nodes as well.

6.1.8 Shared disks architecture

GPFS uses a shared disk architecture that allows data and metadata to be accessible by any of the nodes within the GPFS cluster. This allows the system to achieve parallel access to data and increase performance.

The sharing of disks can happen in either a direct disk connection (a storage LUN is mapped to multiple cluster nodes) or by using the GPFS Network Shared Disk (NSD) protocol over the network (TCP/IP or InfiniBand). A direct disk connection provides the best performance, but might not suit all the technical scenarios. These different layouts are explained in 6.2, “GPFS layouts for technical computing” on page 115.

Because a technical computing cloud environment might be diverse in its hardware infrastructure and hardware components, GPFS provides flexibility while maintaining performance and parallelism to data access.

6.2 GPFS layouts for technical computing

For Technical Computing, GPFS can provide various cluster configurations independent of which file system features you use. There are four basic layouts for the GPFS cluster configurations within a Technical Computing cloud:

- ▶ Shared disk
- ▶ Network block I/O
- ▶ Mixed cluster
- ▶ Sharing data between clusters

6.2.1 Shared disk

In a shared disk cluster layout, nodes can all be directly attached to a common SAN storage as shown in Figure 6-1. This means that all nodes can concurrently access each shared block device in the GPFS cluster through a block level protocol.

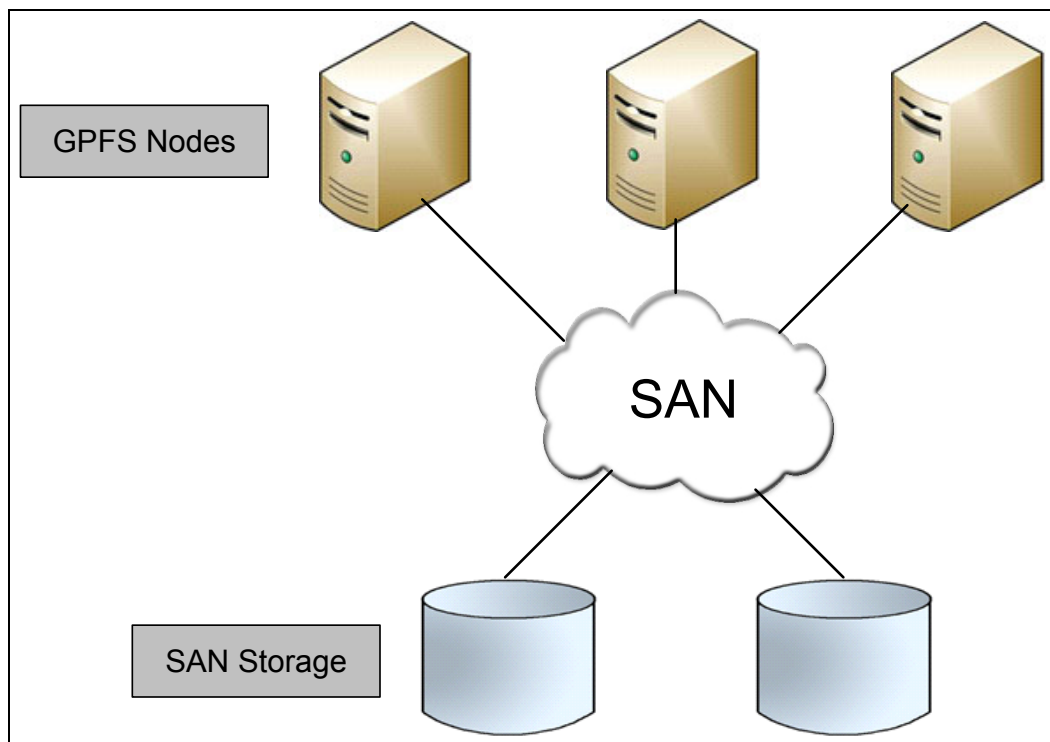


Figure 6-1 SAN-attached storage

Figure 6-1 shows that the GPFS cluster nodes are connected to the storage over the SAN and to each other over the LAN. It shows a Fibre Channel SAN through the storage attachment technology, which can be InfiniBand, SAS, FCoE or any other. Data that are used by applications that run on the GPFS nodes flow over the SAN and the GPFS control information flows among the GPFS instances in the cluster over the LAN. This is a good configuration for providing network file service to client systems using clustered NFS, high-speed data access for digital media applications, or a grid infrastructure for data analytics.

6.2.2 Network block I/O

In environments where you do not have to adopt a single SAN storage technology attached to every node in the cluster, a block level interface over Internet Protocol networks called the NSD protocol can be an option. GPFS clusters can use NSD server nodes to remotely serve disk data to other NSD client nodes. NSD is a disk level virtualization.

In this configuration, disks are attached only to the NSD servers. Within Technical Computing clouds, you can use NSD protocol in GPFS clusters to provide high speed data access for applications that run on NSD client nodes. To avoid a single point of server failure, define at least two NSD servers for each disk as shown in Figure 6-2.

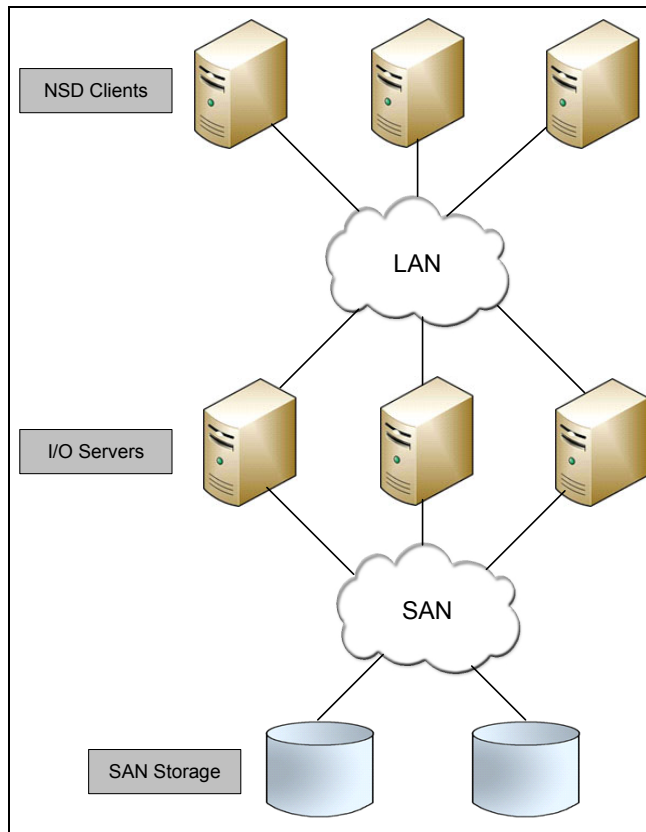


Figure 6-2 Network block I/O

Network block I/O is better for clusters with sufficient network bandwidth between NSD servers and clients. Figure 6-2 shows an example of a network block I/O configuration where NSD clients are connected to NSD servers through a high speed interconnect or an IP-based network such as Ethernet. Figure 6-2 also illustrates that data flows from the storage to the NSD servers over the SAN, and then flows from the NSD servers to the NSD clients across the LAN. NSD clients see the local block I/O devices, same as directly attached, and remote I/O is transparent to the applications. The parallel data access provides the best possible throughput to all clients.

The choice between SAN attachment and network block I/O are performance and costs. In general, using a SAN provides the highest performance; but the cost and management complexity of SANs for large clusters is often prohibitive. In these cases, network block I/O provides an option. For example, a grid is effective for statistical applications such as financial fraud detection, supply chain management or data mining.

6.2.3 Mixed clusters

Based on the previous two sections, you can also mix direct SAN and NSD attachment topologies in GPFS clusters. Figure 6-3 shows some application nodes (with high bandwidth) are directly attached to the SAN while other application nodes are attached to the NSD servers. This mix implementation makes a cluster configuration flexible and provides better I/O throughput to meet the application requirements. In addition, this mix cluster architecture enables high performance access to a common set of data to support a scale-out solution and to provide a high available platform.

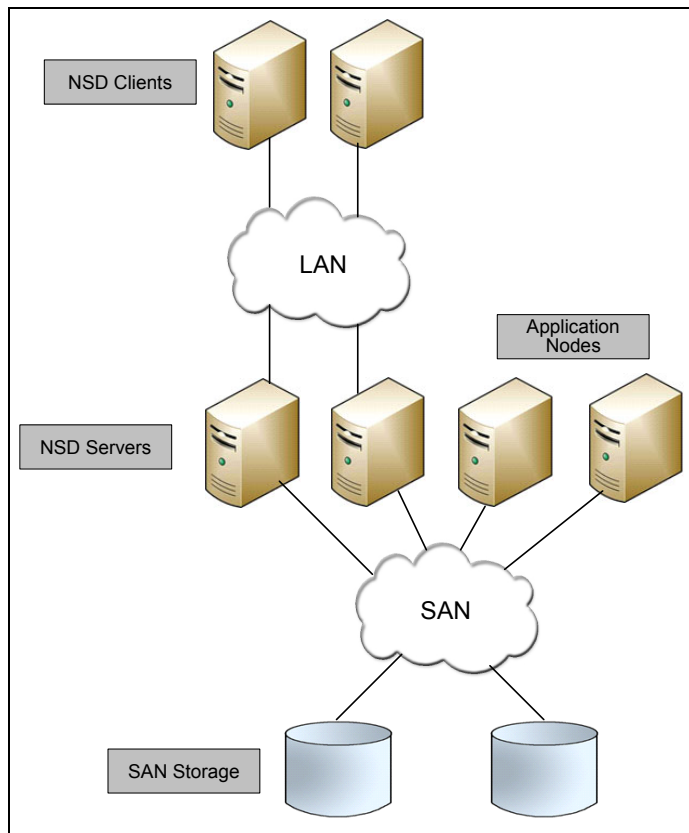


Figure 6-3 Mixed cluster architecture

A GPFS node always tries to find the most efficient path to the storage. If a node detects a block device path to the data, this path is used. If there is no block device path, the network is used. This capability is used to provide extra availability. If a node is SAN-attached to the storage and there is an HBA failure, for example, GPFS can fail over and use the network path to the disk. A mixed cluster topology can provide direct storage access to non-NSD server nodes for high performance operations, including backups.

6.2.4 Sharing data between clusters

There are two ways available to share data between GPFS clusters: GPFS multi-cluster and Active File Management (AFM). GPFS multi-clusters allow GPFS nodes to natively mount a GPFS file system from another GPFS cluster.

A multi-cluster file system features extreme scalability and throughput that is optimized for streaming workloads such as those common in web 2.0, digital media, scientific, and engineering applications. Users shared access to files in either the cluster where the file

system was created, or other GPFS clusters. Each site in the network is managed as a separate cluster, while still allowing shared file system access.

GPFS multi-clusters allow you to use the native GPFS protocol to share data across clusters. With this feature, you can allow other clusters to access one or more of your file systems. You can also mount and have access to file systems that belong to other clusters for which you have been authorized access. Thus, multi-clusters demonstrate the viability and usefulness of a global file system, but also reduce the need for multiple data copies. Figure 6-4 shows a multi-cluster configuration with both LAN and mixed LAN and SAN topologies.

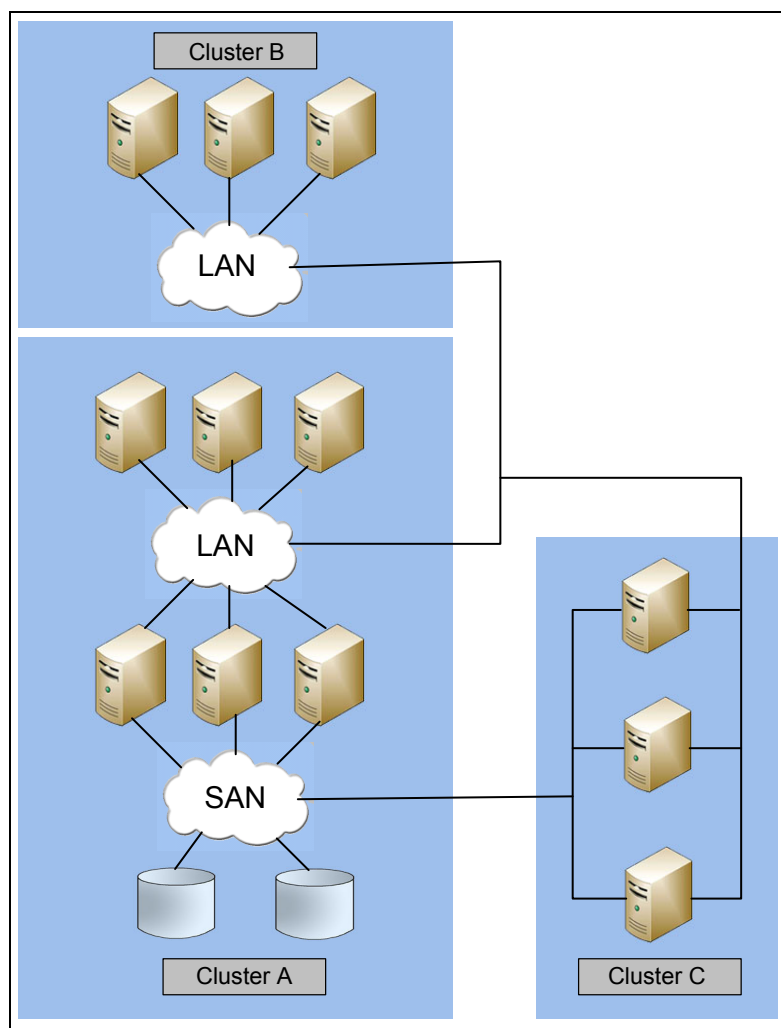


Figure 6-4 Multi-cluster

As shown in Figure 6-4, Cluster A owns the storage and manages the file system. Remote clusters such as Cluster B and Cluster C that do not have any storage are able to mount and access, if they are authorized, the file system from Cluster A. Cluster B accesses the data through the NSD protocol over the LAN, whereas Cluster C accesses the data over an extension of the SAN.

Multi-clusters can share data across clusters that belong to different organizations for collaborative computing, grouping sets of clients for administrative purposes, or implementing a global namespace across separate locations. A multi-cluster configuration connects GPFS multi-cluster clusters within a data center, across a campus, or across reliable WAN links. To share data between GPFS clusters across less reliable WAN links or in cases where you want a copy of

the data in multiple locations, you can use a new feature introduced in GPFS 3.5 called AFM. For more information, see 6.4.1, “Active File Management (AFM)” on page 124.

6.3 Integration with IBM Platform Computing products

To help solve challenges in Technical Computing clouds, IBM GPFS has implemented integrations with IBM Platform Computing products such as IBM Platform Cluster Manager - Advanced Edition (PCM-AE) and Platform Symphony.

6.3.1 IBM Platform Cluster Manager - Advanced Edition (PCM-AE)

PCM-AE integrates with GPFS, which provisions secure multi-tenant high-performance computing (HPC) clusters that are created with secure GPFS storage mounted on each server. Storage is secure because only authorized cloud users can access storage that is assigned to their accounts. You can use GPFS for PCM-AE as an extended storage that looks like one folder in the operating system for virtual machines or physical systems. Or you can use GPFS for PCM-AE as an image (raw, qcow2) storage repository for virtual machines.

Considerations

There are some considerations to have in mind before you implement GPFS for PCM-AE secure storage:

- ▶ The GPFS package URL. This is specified to create the GPFS storage adapter instance.
- ▶ GPFS connects to the provisioning (private) network
- ▶ Configure SSH passwordless logon
- ▶ Configure the **SSH** and **SCP** remote commands
- ▶ Install the **expect** command
- ▶ Enable the GPFS *adminMode* configuration parameter
- ▶ Allow access to the GPFS port (1191)
- ▶ Select the GPFS master
- ▶ Ensure that the GPFS master's host name resolves
- ▶ Install the LDAP client on the GPFS master and NSD server
- ▶ Disable the GPFS automatic mount flag

For GPFS performance tuning in Technical Computing clouds environments, you can disable the *GSSAPIAuthentication* and *UseDNS* settings in the ssh configuration file on the GPFS master within PCM-AE environments.

Usability and administration

This section describes usability and administration examples for GPFS in a PCM-AE cloud environment. In GPFS, you can add a GPFS storage adapter instance, and then create, delete, assign, or unassign storage for GPFS. You can even create a cluster definition with GPFS. The following list gives an administration summary for GPFS:

- ▶ Add a GPFS storage adapter instance
- ▶ Add storage
- ▶ Delete storage
- ▶ Assign storage to accounts
- ▶ Unassign storage from accounts
- ▶ Create a cluster definition with GPFS
- ▶ Instantiate a cluster definition with nodes mounting GPFS
- ▶ Troubleshoot GPFS storage

For more information about GPFS administration, see the “Secure storage using a GPFS cluster file system” chapter in *Platform Cluster Manager Advanced Edition Version 4 Release 1: Administering, SC27-4760-01*.

Integration summary

Cloud administrators must add a storage adapter instance, then go to the GUI page of a specific storage adapter instance to create storage through this adapter.

After storage is assigned to an account, all the users/groups of this account are given the permissions (read, write, execution, and rwx) to access the storage. To achieve this, PCM-AE communicates with the GPFS master node to add user/group to the access control list (ACL) for the related directory in the GPFS file system.

The machine definition author enables the GPFS storage, and can add only one GPFS storage to each machine layer. The storage that is selected in the machine definition is used for the GPFS postscript layer.

Figure 6-5 shows the resulting PCM-AE GUI after you select **Resources** → **Inventory** → **GPFS** → *storage_adapter_instance (c445gpfs2)*. Click the Storage tab and select the storage that you want to assign to an account (Figure 6-5).

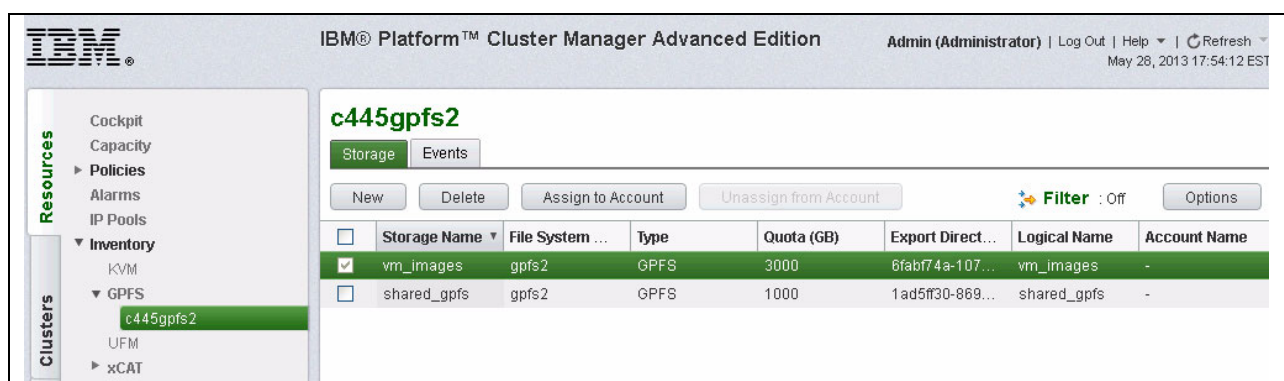


Figure 6-5 Storage tab in PCM-AE

Cloud administrators can define the cluster definition, and select **Storage1** for its machine layer, then publish it and instantiate this cluster definition. The machine definition author can specify a mount point. The mount point is a link path to the real storage under the GPFS file system. This is to be consistent with other shared storage.

The GPFS file system that contains the storage is mounted on the machines in tiers that contain the GPFS internal postscript layer. The postscript layer is an internal postscript layer that PCM-AE runs. Even if implemented as an internal postscript layer, you still must specify the location of the GPFS client packages by the GPFS package URL when you create the GPFS storage adapter instance.

Tip: When the cluster machines are provisioned with the GPFS file system that contains the storage, the user is not required to manually mount the file system.

The GPFS storage that is assigned to the account defaults to the mount point /gpfs. A subdirectory with the cluster ID as its directory name is created in the storage (by the local system command `mkdir`), and linked to the “mount point” that the user specifies. Therefore, the user can access this subdirectory with the path of “mount point”. All these steps run in the GPFS internal postscript layer.

Figure 6-6 shows how the cluster machines provisioned by PCM-AE work with the GPFS cluster.

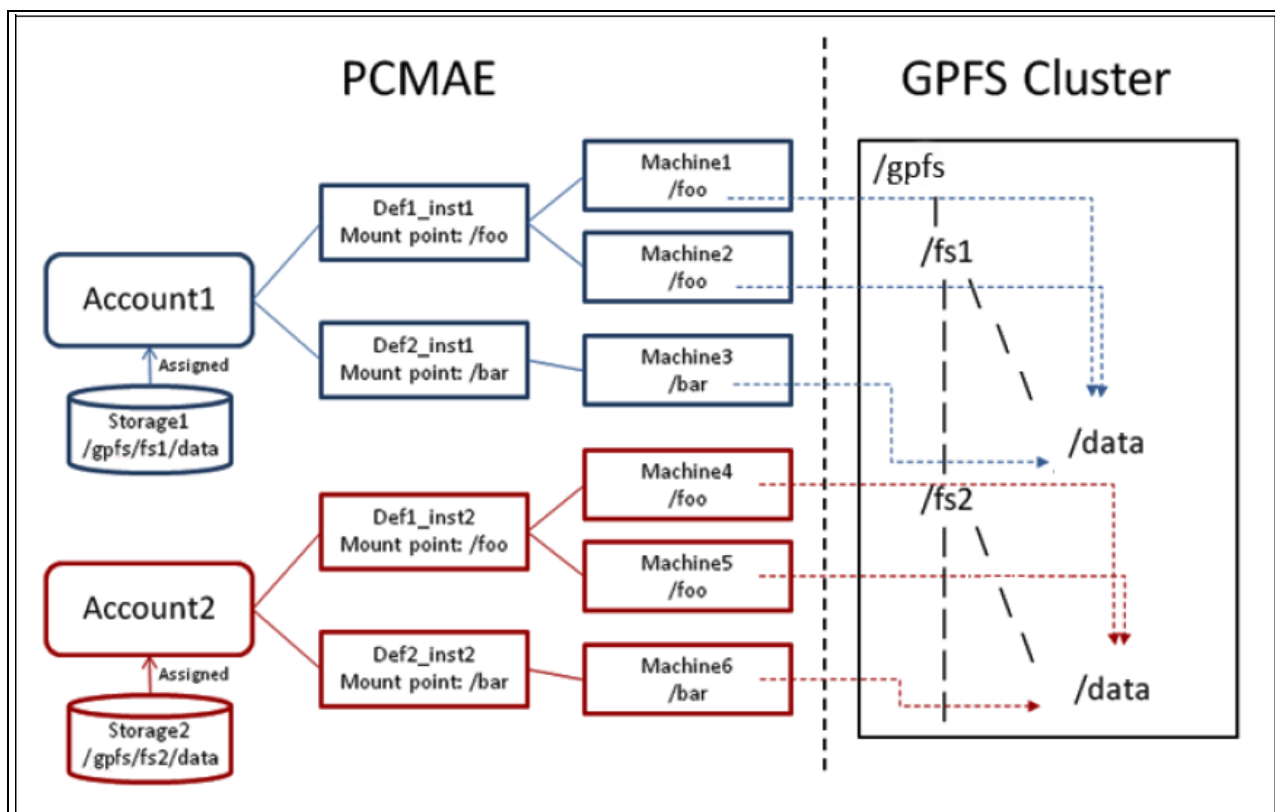


Figure 6-6 PCM-AE and GPFS

The cluster shares the *FileSysName/ExportDirectory* directory in the GPFS file system. The example shown in Figure 6-6 shows two accounts: *Account1* and *Account2*. *Storage1* and *Storage2* are two storages that were created by the administrator. *Storage1* is assigned to *Account1*, and *Storage2* is assigned to *Account2*. For *Storage1*, its file system name is *fs1*, and for *Storage2*, its file system name is *fs2*. For both of them, the export directory is *data*.

Account1 instantiates the cluster definition *Def1_inst1* and specifies */foo* as a mount point for *Machine1* and *Machine2*. *Account1* also instantiates the cluster definition *Def2_inst1* and specifies */bar* as a mount point for *Machine3*. Similarly, *Account2* instantiates the cluster definition *Def1_inst2* and specifies */foo* as a mount point for *Machine4* and *Machine5*. *Account2* also instantiates the cluster definition *Def2_inst2* and specifies */bar* as a mount point for *Machine6*. If you want to have its private folder in the cluster, you can create it in its own postscript layer.

Current considerations

In the integration between PCM-AE and GPFS, keep these considerations in mind:

- ▶ The GPFS cluster must use the same LDAP server with the PCM-AE master so that they can share the user database.
- ▶ One GPFS master cannot be shared by multiple PCM-AE environments.
- ▶ The GPFS master and the cluster must be set up and ready before you start client provisioning.

- ▶ There is a network route between the virtual machines, physical machines, the xCAT management node, the PCM-AE master, the KVM hypervisor, and the GPFS master/NSD server.
- ▶ xCAT is the DNS server for the GPFS master, virtual machines, and physical machines.

6.3.2 IBM Platform Symphony

With IBM InfoSphere BigInsights 2.1, IBM is bringing new capabilities to Hadoop and providing tight integrations to solve many challenges of data management, scheduling efficiency and cluster management.

IBM Platform Symphony and GPFS are included in the IBM InfoSphere BigInsights reference architecture, which can be also supported with IBM Power Linux. The integration between Platform Symphony and GPFS overcomes several specific limitations of Hadoop to meet the big data analytics demands under a Technical Computing clouds, because the high performance scheduler combined with the high performance file system can compliment to deliver the ultimate analytics environment. The integration of IBM Platform Symphony and IBM GPFS provides service-oriented high-performance grid manager with low-latency scheduling and advanced resource-sharing capabilities, and delivering an enterprise-class POSIX file system.

IBM InfoSphere BigInsights 2.1 and GPFS File Placement Optimizer (FPO) features provide users with an alternative to HDFS. The implementation brings POSIX compliance and no single point of failure file system, thus replacing standard HDFS in MapReduce environments.

GPFS FPO delivers a set of features that extend GPFS providing Hadoop-style data-aware scheduling on a shared-nothing environment (meaning cluster nodes with local disk). A GPFS “connector” component included in IBM InfoSphere BigInsights (and also in GPFS) provides BigInsights customers with several advantages. For details, see 6.4.2, “File Placement Optimizer (FPO)” on page 129.

Note: GPFS FPO version 3.5.0.9 has been certified for use with Apache Hadoop 1.1.1 (the same version of Hadoop on which IBM InfoSphere BigInsights 2.1 is based). Platform Symphony 6.1.0.1 has also been tested with open source Hadoop 1.1.1. The necessary connector to emulate HDFS operations on GPFS FPO is included as part of the GPFS software distribution.

Layout

Figure 6-7 illustrates the implementation of the application adapter within the MapReduce framework in IBM Platform Symphony, and the integration between GPFS and Platform Symphony.

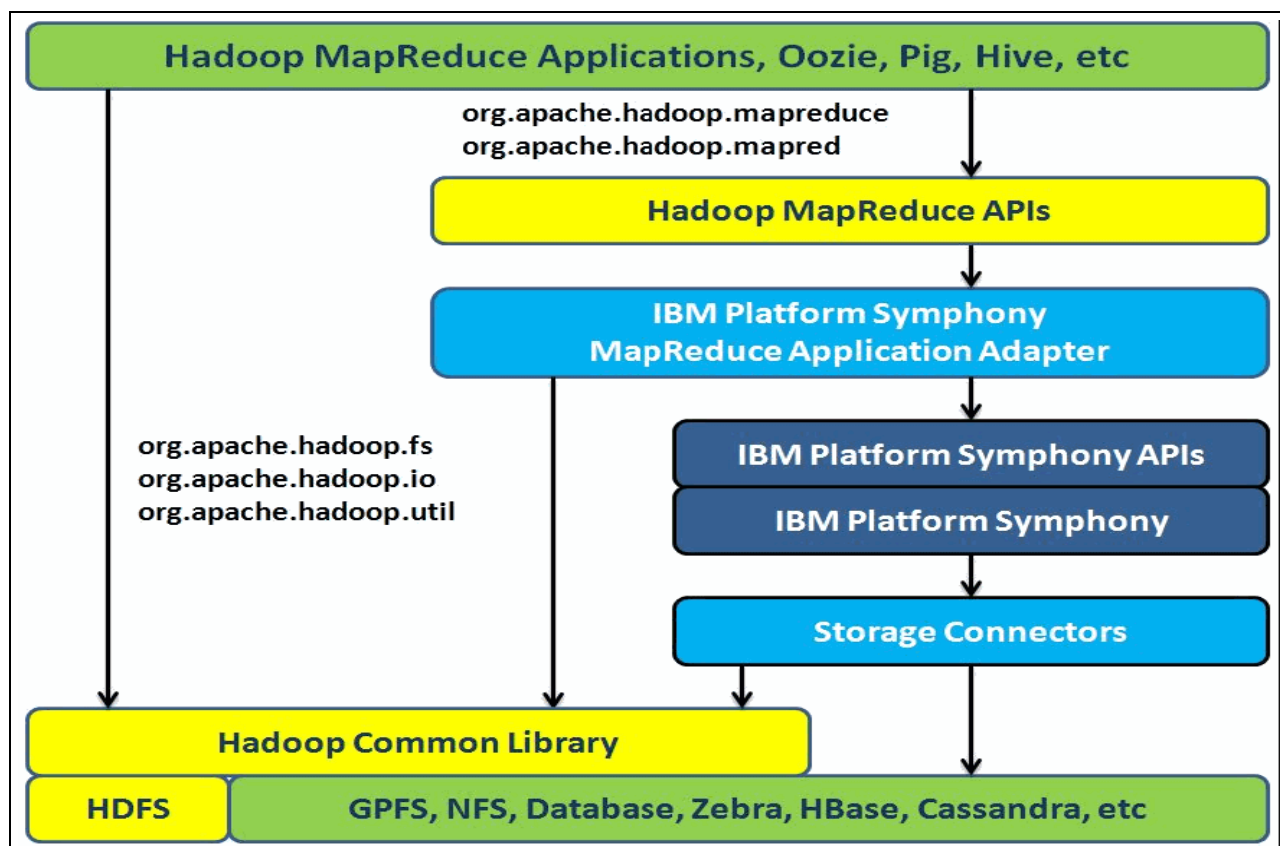


Figure 6-7 Integration between GPFS and Platform Symphony

Existing Hadoop MapReduce applications call the same Hadoop MapReduce and common library APIs. Hadoop-compatible MapReduce applications, however, can talk to and run on Platform Symphony, instead of on the Hadoop JobTracker and TaskTracker, through the MapReduce application adapter as shown in Figure 6-7. Both Hadoop MapReduce applications and the application adapter still call the Hadoop common library for distributed file systems such as HDFS and GPFS configurations.

The MapReduce framework in Platform Symphony also provides storage connectors to integrate with different file and database systems such as GPFS and Network File System (NFS). In the middle, the MapReduce application adapter and storage connectors are aware of both Hadoop and Platform Symphony systems.

Procedure

GPFS is a high-performance shared-disk file system that can provide fast data access from all nodes in a homogeneous or heterogeneous cluster of servers. The MapReduce framework in Platform Symphony provides limited support for running MapReduce jobs using GPFS 3.4 for data input, output, or both. For this purpose, the MapReduce framework in Platform Symphony includes a GPFS adapter (gpfs-pmr.jar) for the generic DFS interface, which is in the distribution *lib* directory.

The current MapReduce adapter for GPFS does not apply any data locality preferences to the MapReduce job's task scheduling. All data are considered equally available locally from any compute host. Large input files are still processed as multiple data blocks by different Map tasks (by default, the block size is set to 32 MB). Complete these steps to configure the GPFS adapter and run MapReduce jobs with it:

- ▶ Configure the GPFS cluster and start it (for more information, see the *IBM General Parallel File System Version 3 Release 5.0.7: Advanced Administration Guide*, SC23-5182-07). Mount the GPFS file system to the same local path for the MapReduce client and all compute hosts (for example, /gpfs).
- ▶ Merge the sample \$PMR_HOME/conf/pmr-site.xml.gpfs file to the general configuration file \$PMR_HOME/conf/pmr-site.xml on all the hosts. Restart the related MapReduce applications by disabling or enabling the applications.
- ▶ Use the GPFS adapter by specifying the gpfs:/// schema in the data input path, output path, or both. For example, if input data files in GPFS are mounted under local path /gpfs/input/ and you want to store the MapReduce job's result in GPFS mounted under the local path /gpfs/output, submit a MapReduce job by using the following syntax:

```
mrsh jar jarfile gpfs:///gpfs/input gpfs:///gpfs/output
```

6.4 GPFS features for Technical Computing

This section contains key features to support technical computing.

6.4.1 Active File Management (AFM)

Active File Management (AFM) is a distributed file caching feature in GPFS that allows the expansion of the GPFS global namespace across geographical distances. AFM helps provide a solution through enabling data sharing when wide area network connections are slow or not reliable for Technical Computing clouds.

AFM Architecture

AFM uses a home-and-cache model in which a single home provides the primary data storage, and exported data is cached in a local GPFS file system. Also, AFM can share data among multiple clusters running GPFS to provide a uniform name space and automate data movement (Figure 6-8 on page 125).

Home

Home can be thought of as any NFS mount point that is designated as the owner of the data in a cache relationship. Home can also be a GPFS cluster that has an NFS exported file system or independent file set. As the remote cluster or a collection of one or more NFS servers that cache connects to, only one Home exists in a cache relationship to provide the main storage for the data.

Cache

Cache is a kind of GPFS file set, and can be thought of as the container used to cache the home data. Each AFM-enabled file set cache is associated with a file set at home. File data is copied into a cache when requested. Whenever a file is read, if the file is not in cache or is not up to date, its data is copied into the cache from home. There are one or more gateway nodes as the cache is handling the cache traffic and communicating with home. Data that are written to cache are queued at the gateway node and then automatically pushed to home as quickly as possible.

In the AFM architecture (Figure 6-8), there are two *Cache* clusters sites and one *Home* cluster site. AFM minimizes the amount of traffic that is sent over the WAN. When Cache is disconnected from Home, cached files can be read, and new files and cached files can be written locally. This function provides high availability to Technical Computing clouds users. Also, cloud user access to the global file system goes through the Cache and they can see local performance if a file or directory is in the Cache.

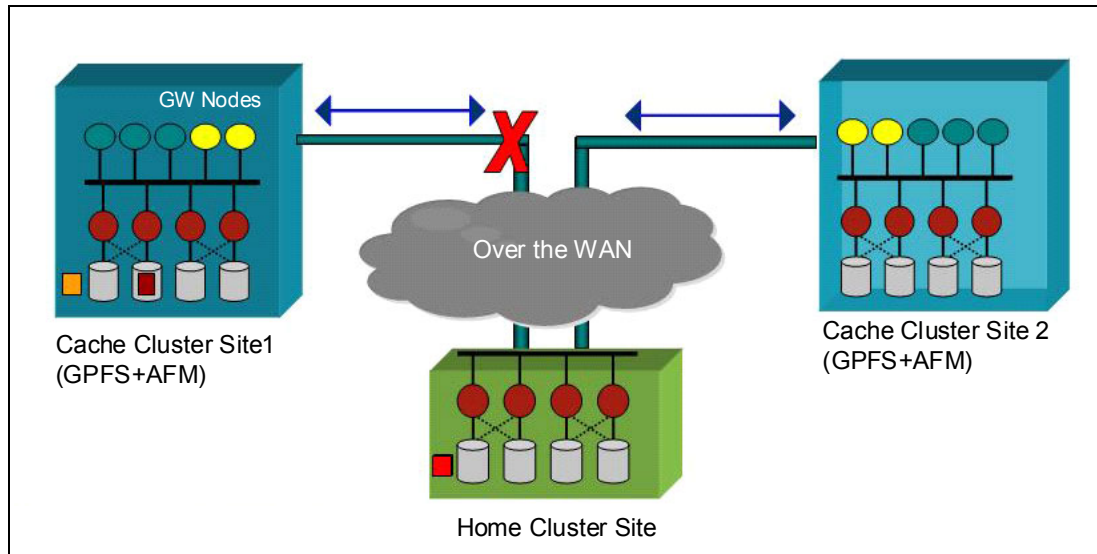


Figure 6-8 AFM architecture

Supported features

AFM supports many features, which include the following:

- ▶ File set granularity. Different file sets at cache sites can link to different homes. Each exported home file set is cached by multiple cache sites.
- ▶ Persistent, scalable, read and write caching. Whole file, metadata, and namespace caching on demand. Configurable to be read-only or cache write only.
- ▶ Standard transport protocol between sites. NFSv3 is supported.
- ▶ Continued operations when network is disconnected. Automatic detection of network outage.
- ▶ Timeout-based cache revalidation. Configurable per object type per file set.
- ▶ Tunable delayed write-back to home. Configurable delayed writeback with command-based flush of all pending updates. Write back for data and metadata.
- ▶ Namespace caching. Directory structure that is created on demand.
- ▶ Support for locking. Locking within cache site only (not forwarded to home).
- ▶ Streaming support.
- ▶ Parallel reads. Multi-node parallel read of file chunks.

For more information about caching modes, see the “Active File Management” chapter in the *General Parallel File System Version 3 Release 5.0.7: Advanced Administration Guide*, SC23-5182-07.

File system operations

There are a few different AFM operations for Technical Computing clouds:

- ▶ Synchronous operations require an application request to block until the operation completes at the home cluster, such as read and lookup.
- ▶ For asynchronous operations, an application can proceed as soon as the request is queued on the gateway node, such as create and write.
- ▶ For synchronization updating, all modifications to the cached file system are sent back to the home cluster in the following situations:
 - The end of the synchronization lag.
 - If a synchronous command depends on the results of one or more updates, it synchronizes all depending commands to the server before its execution.
 - An explicit flush of all pending updates by using the `mmafmc1` command.

For more information about operations, see the “Active file management” chapter in the *General Parallel File System Version 3 Release 5.0.7: Advanced Administration Guide*, SC23-5182-07.

Cache modes

GPFS 3.5.0.7 has three AFM caching modes that are available to control the flow of data to support various cloud environments:

- ▶ Local-update: Files that are created or modified in the cache are never pushed back to home. Cache gets dirty on modifications, and thereafter the relationship with home is cut off from the parent level.
- ▶ Read-only: Data in the cache is read-only.
- ▶ Single-writer: Only one cache file set does all the writing to avoid write conflicts. In other words, a file set exported from home-server is assumed to be accessed from only one caching site.

For more information about caching modes, see the “Active file management” chapter in the *General Parallel File System Version 3 Release 5.0.7: Advanced Administration Guide*, SC23-5182-07.

Limitations/restrictions of AFM

Some restrictions apply for a configuration using AFM, which was introduced in GPFS 3.5.0.7:

- ▶ Snapshots on either side are also not played on the other side.
- ▶ The hard links at the home site are not copied as hard links to cache.
- ▶ A file clone can be displayed either only at the home system or only in the cache system, not both. Clones at home are displayed as normal files in cache.
- ▶ The clone, snapshot, hard link, and rename restrictions also apply to data that is prefetched from the home site to the cache site.
- ▶ AFM does not support NFS delegation.
- ▶ A file renaming at the home site results in different inode numbers assigned at the cache site.
- ▶ There is no option to encrypt or compress AFM data.

For more information, see the “Active file management” chapter in the *General Parallel File System Version 3 Release 5.0.7: Advanced Administration Guide*, SC23-5182-07.

Solution for file sharing

AFM can provide good solutions to address different data management issues. When file sharing is mentioned, you might think more of how to enable clusters to share data at higher performance levels than normal file sharing technologies like NFS or Samba. For example, you might want to improve the performance when geographically dispersed people need access to the same set of file based information in the clouds. Or you might want to manage the workflow of highly fragmented and geographically dispersed file data in the clouds. AFM can provide a good solution for sharing file in a Technical Computing cloud environment.

AFM can use GPFS global namespace to provide these core capabilities for file sharing:

- ▶ Global namespace enables a common view of files and file location no matter where the file requester or file are.
- ▶ Active File Management handles file version control to ensure integrity.
- ▶ Parallel data access allows for large number of files and people to collaborate without performance impact.

As a key important point in clouds, AFM can take global namespace by automatically managing asynchronous replication of data. But there are some restrictions of files and file attributes replication. For more information, see “Limitations/restrictions of AFM” on page 126. Generally, AFM can see all data from any cluster, and cache as much data as required or fetch data on demand. The file system in each cluster has the same name and mount point.

Figure 6-9 shows an example of a global namespace. There are three clusters at different locations in an AFM cache relationship. The GPFS client node from any site sees all of the data from all of the sites. Each site has a single file system, but they all have the same mount point. Each site is the home for two of the subdirectories and cache file sets pointing to the data that originates at the other sites. Every node in all three clusters has direct access to the global namespace.

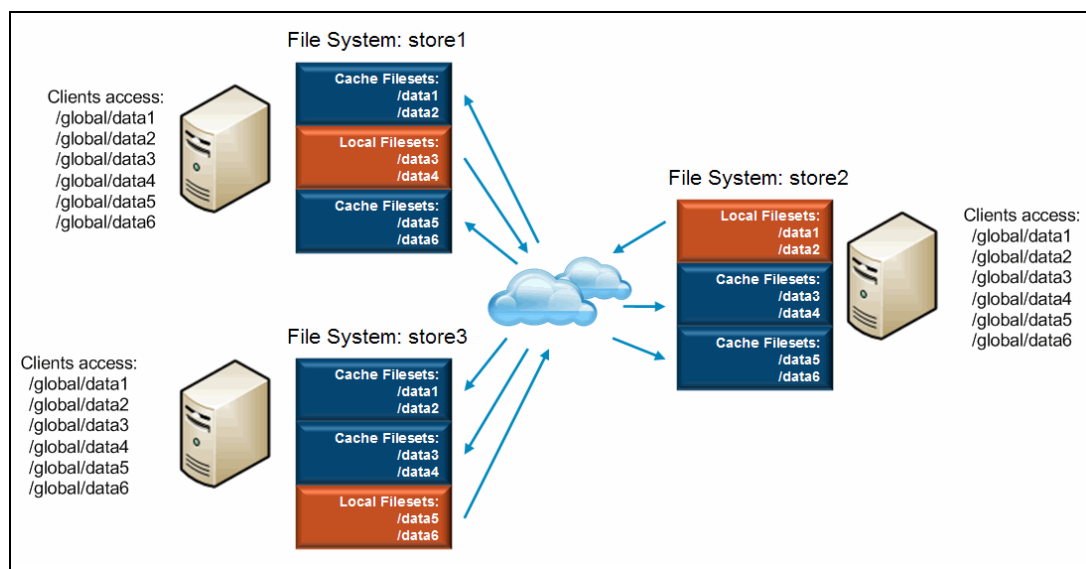


Figure 6-9 Global namespace

Solution for centralized data management for branch offices

In this scenario, assume that you have a geographically dispersed organization with offices of all different sizes. In the large offices, you have IT staff to perform data management tasks such as backups. In smaller offices, you need local systems to provide performance for the

local user community, but the office is not large enough to warrant a complete backup infrastructure. You can use AFM in this case to provide a way to automatically move data back to the central site.

The AFM configuration contains a central site that has an NFS export for each remote site. Each remote site is configured as a single writer for that home. During a complete failure at the remote office, you can preinstall a new server at the central site by running a prefetch to fill the file system, then ship it to the remote office. When the new system is installed at the remote office, the data are there and are ready to continue serving files. In the meantime, the clients, if you like, can connect directly to the home for their data.

Figure 6-10 shows a central site where data are created, updated, and maintained. Branch sites can periodically prefetch by using a policy or pull on demand. Data is revalidated when accessed. Data are maintained at central location, and other sites in various locations pull data into cache and serve the data locally at that location.

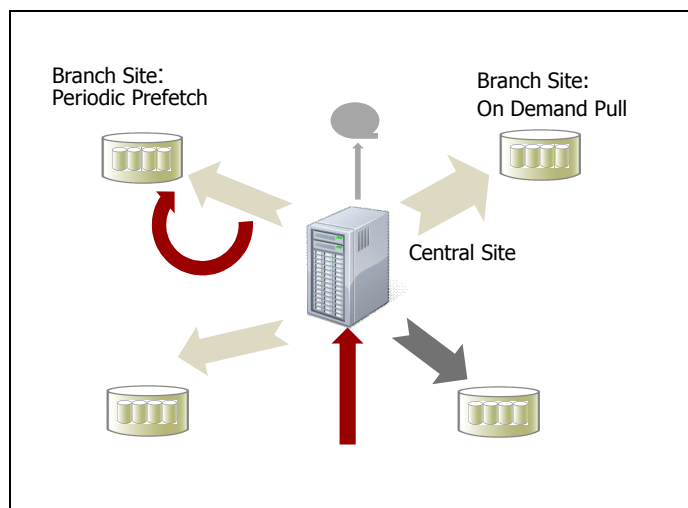


Figure 6-10 Central and branch offices

Solution for data library with distributed compute

Sometimes as compute farms grow, it is impractical to scale up a full bandwidth network to support direct access to the library of data from each compute node. To address this challenge, AFM can be used to automatically move the data closer to the compute nodes that are doing the work.

In the example shown in Figure 6-11, the working set of data is relatively small, but the number of compute nodes is large. There are three GPFS compute clusters and one data library from the remote. When the job starts, the data that are needed for the job are prefetched from the library into each of the three cache clusters (Compute Cluster1, Compute Cluster2, Compute Cluster3). These cache clusters are within isolated InfiniBand networks that provide fast access to the copy of the data in the cache. This way, you can distribute the data close to the compute nodes automatically, even if the compute cluster are in different cities or countries.

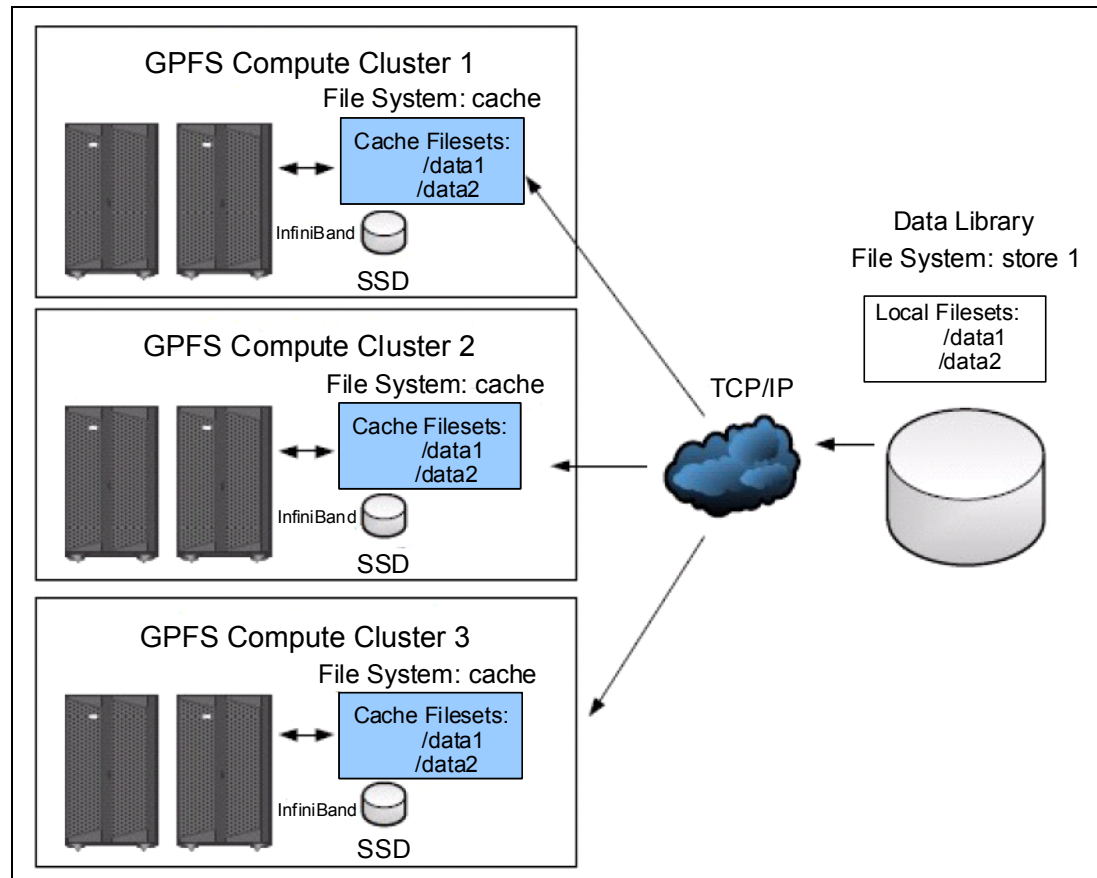


Figure 6-11 Data library with distributed compute clusters

6.4.2 File Placement Optimizer (FPO)

The new FPO feature is available starting with GPFS 3.5.0.7, can be used only for Linux environments either on IBM System x or on IBM Power servers. FPO is designed for MapReduce workloads, present in the emerging class of big data applications. It aims to be an alternative for the Hadoop Distributed File System (HDFS) currently used in Hadoop MapReduce framework deployments.

IBM InfoSphere BigInsights 2.1, as a first example, integrates GPFS-FPO in its extended Hadoop MapReduce software stack:

<http://www-01.ibm.com/software/data/infosphere/biginsights/>

How FPO works

All the cluster nodes are configured as NSD servers and clients to each other. Each node gets access to the local disk space of the other nodes. Such a setup is called *shared nothing architecture* or *shared nothing cluster*. Figure 6-12 presents a topology layout of the GPFS-FPO architecture.

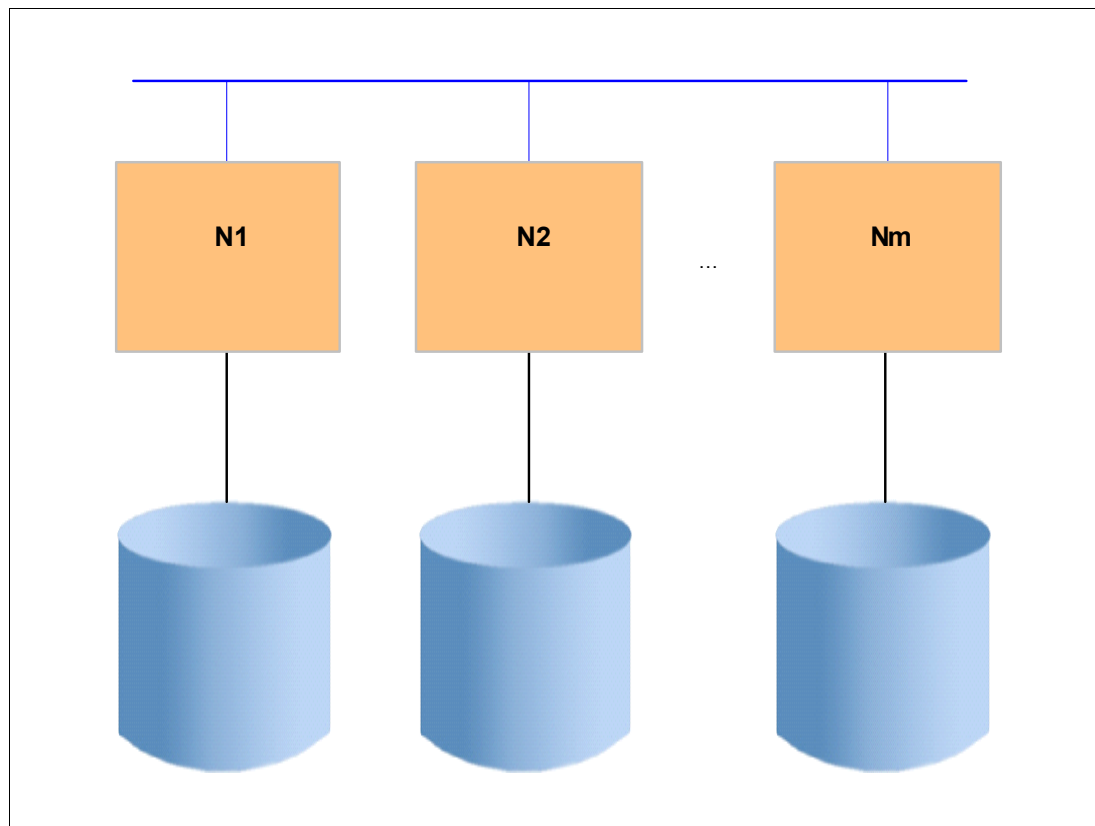


Figure 6-12 GPFS-FPO topology layout

The configured local disk space of each node is aggregated by GPFS into a redundant shared file system. Traditional GPFS was able to provide this function, but only as a standard GPFS cluster file system. The GPFS-FPO extension additionally provides HDFS equivalent behavior, and even some enhancements through the following feature optimizations:

- Locality awareness** This is implemented by an extension of the traditional GPFS failure group notation from a simple integer number to a three-dimensional topology vector of three numbers (r, p, n) such as encoding rack, placement inside the rack, and node location information. Location awareness allows MapReduce tasks to be optimally scheduled on nodes where the data chunks are.
- Chunks** A chunk is a logical grouping of file system blocks allocated sequentially on a disk to be used as a single large block. Chunks are good for applications that demand high sequential bandwidth such as MapReduce tasks. By supporting both large and small block sizes in the same file system, GPFS is able to meet the needs of different types of applications.
- Write affinity** Allows applications to determine the placement of chunk replicas on different nodes to maximize performance for some special disk access patterns of the applications. A write affinity depth parameter indicates

the approach for the striping and placement of chunk replicas in different failure groups and relative to the writer node. For example, when using three replicas, GPFS might place two of them in different racks to minimize chances of losing both. However, the third replica might be placed in the same rack with one of the first two, but in a different half. This way the network traffic between racks that is required to synchronize the replicas is reduced.

Pipeline replication Higher performance is achieved by using pipeline replication than is possible with single-source replication, allowing better network bandwidth utilization.

Distributed recovery Restripe and replication load is spread out this way over multiple nodes. That minimizes the effect of failures during ongoing computation.

New GPFS terms and commands options

When configuring a GPFS-FPO environment, specify related attributes for entities in the GPFS cluster at various levels: Cluster, file system, storage pool, file, failure group, disk, chunk, or block.

To help introduce the new features, the definitions for the storage pool and failure group notions are reviewed, and the format of the stanza file presented as already introduced by GPFS 3.5. Then, focus on the new FPO extensions that were introduced by GPFS 3.5.0.7.

For more information about these topics, see the product manuals at:

<http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs.doc/gpfsbooks.html>

For more information about the standard GPFS 3.4 with implementation examples for various scenarios, see *Implementing the IBM General Parallel File System (GPFS) in a Cross-Platform Environment*, SG24-7844, which is available at:

<http://www.redbooks.ibm.com/abstracts/sg247844.html>

Storage pools and failure groups

A storage pool is a collection of disks that share a common set of administrative characteristics. Performance, locality, or reliability can be such characteristics. You can create tiers of storage by grouping, for example, SSD disks in a first storage pool, Fibre Channel disks in a second pool, and SATA disks in a third pool.

A failure group is a collection of disks that share a common single point of failure and can all become unavailable if a failure occurs. GPFS uses failure group information during data and metadata placement to ensure that two replicas of the same block are written on disks in different failure groups. For example, all local disks of a server node should be placed in the same failure group.

Specify both the storage pool and the failure groups when you prepare the physical disks to be used by GPFS.

The containing storage pool of a disk is specified as an attribute for the NSD. An NSD is just the cluster-wide name at the GPFS level for the physical disk. First define the disk as an NSD with the `mmcrnsd` command, then create a file system on a group of NSDs by using the `mmcrfs` command.

Stanza files

GPFS 3.5 extends the syntax for some of its commands with a stanza file option. Instead of a disk descriptor line for each considered physical disk, as specified in previous versions, now the file contains an NSD stanza. Example 6-1 shows an NSD stanza in a stanza file. The NSD stanza starts with the %nsd: stanza identifier, and continues with stanza clauses explained here by the preceding comments lines. The comments lines (#) are optional.

Example 6-1 NSD stanza example

```
%nsd:
nsd=gpfs001 # NSD name for the first disk device
# first disk device name on node n01
device=/dev/sdc
# NSD server node for this disk device
servers=n01
# storage pool for this disk
pool=dataPoolA
# failure group of this disk
failureGroup=2,1,0
```

The storage pool of the considered disk is specified as dataPoolA string value in the pool clause.

Extended failure group

The failure group clause of a disk can be specified not only as a single number such as in standard GPFS, but also as a vector of up to three comma-separated integer numbers. The line at the end of previous Example 6-1 specifies the failure group for the disk /dev/sdc, having a vector value of 2,1,0.

The vector specifies topology information that is used later by GPFS when making data placement decisions. This topology vector format allows the user to specify the location of each local disk in a node within a rack with three coordinate numbers that represent, in order:

- ▶ The rack that contains the node with the involved disk installed
- ▶ A position within the rack for the node that contains the involved disk
- ▶ The node in which the disk is located

Example 6-1 shows the topology vector 2,1,0 that identifies rack 2, bottom half, and first node.

When considering two disks for striping or replica placement purposes, it is important to consider the following factors:

- ▶ Disks that differ in the first of the three numbers are farthest apart (in different racks).
- ▶ Disks that have the same first number but differ in the second number are closer (in the same rack, but in different halves).
- ▶ Disks that differ only in the third number are in different nodes in the same half of the same rack.
- ▶ Only disks that have all three numbers in common are in the same node.

The default value for the failure group clause is -1, which indicates that the disk has no point of failure in common with any other disk.

The pool and failure group clauses are ignored by the **mmcrnsd** command, and are passed unchanged to the generated output file to be used later by the **mmcrfs** command.

Pool stanza

GPFS 3.5.0.7 extends the functionality for the storage pool object and adds a so-called *pool stanza* to pass FPO-related attributes to the `mmcrfs` or `mmaddisk` commands. A pool stanza must match the following format:

```
%pool:
pool=StoragePoolName
blockSize=BlockSize
usage={dataOnly | metadataOnly | dataAndMetadata}
layoutMap={scatter | cluster}
allowWriteAffinity={yes | no}
writeAffinityDepth={0 | 1}
blockGroupFactor=BlockGroupFactor
```

Block group factor

The new block group factor attribute allows GPFS to specify the chunk size. Chunk size is how many successive file system blocks to be allocated on disk for a chunk. It can be specified on a storage pool basis by a *blockGroupFactor* clause in a storage pool stanza as shown in the syntax above. This can also be specified at file level, by using by the `--block-group-factor` argument of the `mmchattr` command. The range of the block group factor is from 1 to 1024. The default value is 1. For example, with a 1 MB file system block size and block group factor of 128, you get an effective large block size of 128 MB.

Write affinity

The *allowWriteAffinity* clause indicates whether the storage pool has the FPO feature enabled or not. Disks in an FPO-enabled storage pool can be configured in extended failure groups that are specified with topology vectors.

The *writeAffinityDepth* parameter specifies the allocation policy used by the node writing the data. A write affinity depth of 0 indicates that each replica is to be striped across the disks in a cyclical fashion. There is the restriction that no two disks are in the same failure group. A write affinity depth of 1 indicates that the first copy is written to the writer node, unless a write affinity failure group is specified by using the `write-affinity-failure-group` option of the `mmchattr` command. For the second replica, it is written to a different rack. The third copy is written to the same rack as the second copy, but on a different half (which can be composed of several nodes). A write affinity depth of 2 indicates that the first copy is written to the writer node. The second copy is written to the same rack as the first copy, but on a different bottom half (which can be composed of several nodes). The target node is determined by a hash value on the file set ID of the file, or it is chosen randomly if the file does not belong to any file set. The third copy is striped across the disks in a cyclical fashion with the restriction that no two disks are in the same failure group.

Note: Write affinity depth of 2 is supported starting from 3.5.0.11(3.5 TL3)

Write affinity failure group

Write affinity failure group is a policy that can be used by an application to control the layout of a file to optimize its specific data access patterns. The write affinity failure group can be used to decide the range of nodes where the chunk replicas of a particular file are allocated. You specify the write affinity failure group for a file through the `write-affinity-failure-group` option of the `mmchattr` command. The value of this option is a semicolon-separated string of one, two, or three failure groups in the following format:

```
FailureGroup1[;FailureGroup2[;FailureGroup3]]
```

Each failure group is represented as a comma-separated string that identifies the rack (or range of racks), location (or range of locations), and node (or range of nodes) of the failure group in the following format:

```
Rack1{:Rack2{...{:Rackx}}};Location1{:Location2{...{:Locationx}}};ExtLg1{:ExtLg2{...{:ExtLgx}}}
```

For example, the attribute 1,1,1:2;2,1,1:2;2,0,3:4 indicates the following locations:

- ▶ The first replica is on rack 1, rack location 1, nodes 1 or 2
- ▶ The second replica is on rack 2, rack location 1, nodes 1 or 2
- ▶ The third replica is on rack 2, rack location 0, nodes 3 or 4

If any part of the field is missing, it is interpreted as 0. For example, the following settings are interpreted the same way:

```
2
2,0
2,0,0
```

Wildcard characters (*) are supported in these fields. The default policy is a null specification, which indicates that each replica is to be wide striped over all the disks in a cyclical fashion such that no two replicas are in the same failure group. Example 6-2 shows how to specify the write affinity failure group for a file.

Example 6-2 Specifying a write affinity failure group for a file

```
mmchattr --write-affinity-failure-group="4,0,0;8,0,1;8,0,2" /gpfs1/file1
mmfslattr -L /gpfs1/file1
file name: /gpfs1/file1
metadata replication: 3 max 3
data replication: 3 max 3
immutable: no
appendOnly: no
flags:
storage pool name: system
File set name: root
snapshot name:
Write Affinity Depth Failure Group(FG) Map for copy:1 4,0,0
Write Affinity Depth Failure Group(FG) Map for copy:2 8,0,1
Write Affinity Depth Failure Group(FG) Map for copy:3 8,0,2
creation time: Wed May 15 11:40:31 2013
Windows attributes: ARCHIVE
```

Recovery from disk failure

Automatic recovery from disk failure is activated by the cluster configuration attribute:

```
mmchconfig restripeOnDiskFailure=yes -i
```

If a disk becomes unavailable, the recovery procedure first tries to restart the disk. If this fails, the disk is suspended and its blocks are re-created on other disks from peer replicas.

When a node joins the cluster, all its local NSDs are checked. If they are in a down state, an attempt is made to restart them.

Two parameters can be used for fine-tuning the recovery process:

```
mmchconfig metadataDiskWaitTimeForRecovery=seconds
mmchconfig dataDiskWaitTimeForRecovery=seconds
```

dataDiskWaitTimeForRecovery specifies a time, in seconds, starting from the disk failure, during which the recovery of dataOnly disks waits for the disk subsystem to try to make the disk available again. If the disk remains unavailable, it is suspended and its blocks are re-created on other disks from peer replicas. If more than one failure group is affected, the recovery actions start immediately. Similar actions are run if disks in the system storage pool become unavailable. However, the timeout attribute in this case is **metadataDiskWaitTimeForRecovery**.

The default value for **dataDiskWaitTimeForRecovery** is 600 seconds, whereas **metadataDiskWaitTimeForRecovery** defaults to 300 seconds.

The recovery actions are asynchronous, and GPFS continues its processing while the recovery attempts occur. The results from the recovery actions and any encountered errors are recorded in the GPFS logs.

GPFS-FPO cluster creation considerations

This is not intended to be a step by step procedure on how to install and configure a GPFS-FPO cluster from scratch. The procedure is similar to setting up and configuring a traditional GPFS cluster, so follow the steps in “3.2.4 Setting up and configuring a three-node cluster” in *Implementing the IBM General Parallel File System (GPFS) in a Cross-Platform Environment*, SG24-7844. This section only describes the FPO-related steps specific to a shared nothing cluster architecture.

Installing GPFS on the cluster nodes

Complete the three initial steps to install GPFS binaries on the Linux nodes: Preparing the environment, installing the GPFS software, and building the GPFS portability layer. For more information, see “Chapter 5. Installing GPFS on Linux nodes” of the *Concepts, Planning, and Installation Guide for GPFS release 3.5.0.7*, GA76-0413-07.

In a Technical Computing cloud environment, these installation steps are integrated into the software provisioning component. For example, in a PCM-AE-based cloud environment, the GPFS installation steps can be integrated within a bare-metal Linux and GPFS cluster definition. Or inside a more comprehensive big data ready software stack composed of a supported Linux distribution, GPFS-FPO as alternative to HDFS file system, and an IBM InfoSphere BigInsights release that is supported with the GPFS-FPO version.

Activating the FPO features

Assume that a GPFS cluster has already been created at the node level with the **mmcrcluster** command, and validated with the **mmfsccluster** command. The GPFS-FPO license must now be activated:

```
mmchlicense fpo [--accept] -N {Node[,Node...]} [NodeFile | NodeClass]
```

Now the cluster can be started up by using the following commands:

```
mmfsllicense -L  
mmstartup -a  
mmgetstate -a
```

Some configuration attributes must be set at cluster level to use FPO features:

```
mmchconfig readReplicaPolicy=local  
mmfscconfig
```

Disk recovery features can also be activated at this moment:

```
mmchconfig restripeOnDiskFailure=yes -i  
mmfscconfig
```

Configuring NSDs, failure groups, and storage pools

For each physical disk to be used by the cluster, you must create an NSD stanza in the stanza file. You can find details about the stanza file preparation in “Pool stanza” on page 133. Then, use this file as input for the `mmcrnsd` command to configure the cluster NSDs:

```
mmcrnsd -F StanzaFile  
mmnsd
```

Each storage pool to be created must have a pool stanza specified in the stanza file. For FPO, you must create a storage pool with FPO property enabled by specifying `layoutMap=cluster` and `allowWriteAffinity=yes`. The pool stanza information is ignored by the `mmcrnsd` command, but is used when you further pass the file as input to the `mmcrfs` command:

```
mmcrfs Device -F StanzaFile OtherOptions  
mmfs Device
```

The maximum supported number of data and metadata replicas is three for GPFS 3.5.0.7 and later, and two for older versions.

Licensing changes

Starting with GPFS 3.5.0.7, the GPFS licensing for Linux hosts changes to a Client/Server/FPO model. The GPFS FPO license is now available for GPFS on Linux along with the other two from previous versions: GPFS server license and GPFS client license. This new license allows the node to run NSD servers and to share GPFS data with partner nodes in the GPFS cluster. But the partner nodes must be properly configured with either a GPFS FPO or a GPFS server license. The GPFS FPO license does not allow sharing data with nodes that have a GPFS client license or with non-GPFS nodes.

The announcement letter for the extension of GPFS 3.5 for Linux with the FPO feature provides licensing, ordering, and pricing information. It covers both traditional and new FPO-based GPFS configurations, and is available at:

<http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=ca&infotype=an&appname=iSource&supplier=897&letternum=ENUS212-473>

A comprehensive list of frequently asked questions and their answers, addressing FPO among other topics, is available at:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs.doc/gpfs_faqs/gpfsclustersfaq.html

Current limitations

These restrictions apply to the FPO feature at the GPFS 3.5.0.7 level:

- ▶ Storage pool attributes are set at creation time and cannot be changed later. The system storage pool cannot be FPO-enabled and have the `metadataOnly` usage attribute.
- ▶ When adding a disk in an FPO pool, you must specify an explicit failure group ID for this disk. All disks in an FPO pool that share an NSD server must belong to the same failure group. Only one NSD server can share the disks in an FPO pool. If one storage pool of a file system is FPO-enabled, all the other storage pools in that file system must be FPO-enabled as well.
- ▶ Nodes running the FPO feature cannot coexist with nodes that run GPFS 3.4 or earlier.
- ▶ The architectural limits allow FPO clusters to be scaled at thousands of nodes. GPFS 3.5.0.7 FPO feature tested limit is 64 nodes. Contact gpfs@us.ibm.com if you plan to deploy a larger FPO cluster.

- ▶ All FPO pools must have the same `blockSize`, `blockGroupFactor`, and `writeAffinityDepth` properties.
- ▶ Disks that are shared among multiple nodes are not supported in an FPO file system.
- ▶ An FPO-enabled file system does not support the AFM function.
- ▶ FPO is not supported on the Debian distribution.

Comparison with HDFS

GPFS-FPO has all the ingredients to provide enterprise-grade distributed file system space for workloads in Hadoop MapReduce big data environments.

Compared with Hadoop, HDFS GPFS-FPO comes with the enterprise class characteristics of the regular GPFS: Security, high availability, snapshots, back up and restore, policy-driven tiered storage management and archiving, asynchronous caching, and replication.

Derived from the intrinsic design of its decentralized file and metadata servers, GPFS operates as a highly available cluster file system with rapid recovery from failures. Also, metadata processing is distributed over multiple nodes, avoiding the bottlenecks of a centralized approach. For Hadoop HDFS in the stable release series, 1.0.x and 0.20.x, the name-node acts as a dedicated metadata server, and is a single point of failure. This implies extra sizing and reliability precautions when choosing the hosting physical machine for the namenode. The 2.0.x release series of Hadoop adds support for high availability, but this cannot be considered yet for production environments as they are still in alpha or beta stages:

<http://hadoop.apache.org/releases.html>

Also, GPFS follows POSIX semantics, which makes it easier to use and manage the files. Any application can read and write them directly from and to the GPFS file system. There is no need to copy the files between the shared and local file systems when applications that are not aware of HDFS must access the data. Also, disk space savings occurs by avoiding this kind of data duplication.

Because both system block size and larger chunk size are supported, small and large files can be efficiently stored and accessed simultaneously in the same file system. There is no penalty in using the appropriate block size, either larger or smaller, by various applications with different data access patterns, which can now share disk space.



Solution for engineering workloads

This chapter provides a preview of the solution and architecture for running engineering workloads in cloud-computing environments. To understand how to get the engineering workloads deployed for running in the cloud, you must understand all the components that are part of the solution architecture. This chapter also provides technical computing use case solutions for engineering workloads.

This chapter includes the following sections:

- ▶ Solution overview
- ▶ Architecture
- ▶ Components
- ▶ Use cases

7.1 Solution overview

Under intense market pressure to produce better product designs quickly and cost-effectively, engineering teams are becoming more diverse than ever before. Workgroups are distributed across multiple locations worldwide, each one situated in a different type of regulatory and IT environment. In addition, each workgroup can be using different standards, tools, and processes.

To run resource-intensive simulations, globalized workforces are moving toward a shared high-performance computing (HPC) model in which centralized HPC systems replace local computing infrastructure. This arrangement can work well, but it raises two challenges. First, decentralized data creates versioning issues, especially when different teams need access to the same simulation results. Second, moving large simulation files is time consuming, so much so that the time delay can negate productivity gains made by sharing HPC resources.

In today's fast-paced environments, aerospace, defense, and automotive companies that develop or manufacture products need speed, agility, control, and visibility across the design environment and lifecycle to meet time-to-market requirements and maximize profitability. IBM technical computing clouds solution for engineering can help these companies transform their design chain to develop products better, faster, and cheaper.

7.1.1 Traditional engineering deployments

Manufacturers face enormous pressures to make products that are stronger and last longer, while reducing cost, increasing innovation, and shortening development cycles. To address these demands, manufacturers need engineering simulation solutions that allow users to design and verify products in a virtual, risk-free environment. This minimizes the need for physical prototypes and tests.

In a traditional engineering environment, computing resources are often deployed in support of a single workload, project, or organization. As a result, computing silos are formed that must be managed and maintained. Thus, user and application portability is limited, and often allocated resources fail to meet demand. The outcome is uneven and constrained processing across your organization, higher costs, and the potential for delayed results.

Compute cluster

When engineers work remotely, they access engineering applications and centralized product development centers from a notebook, desktop, web browser, or another rich client. These applications run on hosted servers suitable for *computer-aided design* (CAD), *computer-aided manufacturing* (CAM), *computer-aided engineering* (CAE), process management, and other workloads. What makes these servers work well is the ability to allocate compute, memory, and other resources to workloads dynamically, as well as migrate running workloads from one system to another. Separate technical computing clusters, HPC resources and dynamic job schedulers enable shared, highly utilized analysis environments where engineers can run multiple simulation runs, pre-production runs, and other HPC capabilities.

Deskside visualization

It has been typical to provide costly physical systems to be used as engineering workstations to deliver 3D graphics applications that demand hardware acceleration of a high-end graphics card to efficiently render very large models (millions of vertices) such as airplanes and automobiles. As a result, many engineers today have workstations for 3D design, and others to run enterprise applications and collaboration tools such as email and instant messaging. This multiple workstation model is inefficient and costly. In addition, this workstation model

approach does not lend itself well for the type of collaboration necessary to enable real-time review of component designs.

Figure 7-1 illustrates a common workflow found in most traditional engineering environments.

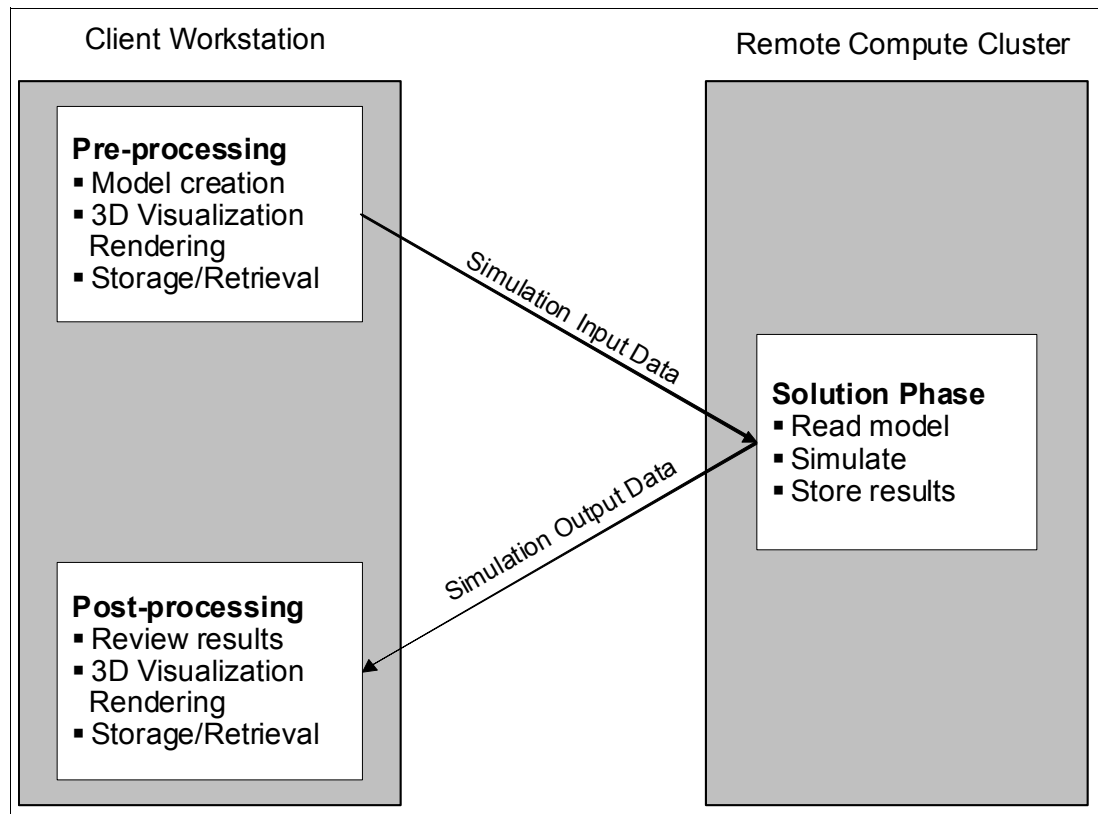


Figure 7-1 Traditional engineering workflow based on desktop visualization

7.1.2 Engineering cloud solution

The engineering cloud solution provides a high performance visual computing environment, enabling remote and scalable graphics without the need for high-end workstations. This open standards-based solution provides a cloud infrastructure environment for organizations that have large 3D intensive graphics requirements and want to reduce costs and improve collaboration between their own designers and remote designers, testers, and component manufacturers. This solution allows engineering designers to use a standard desktop environment to gain access to 3D applications within the 3D Cloud infrastructure without the need for extra graphics cards within their desktop. In addition, the technology enables effective collaboration with local or remote designers, testers, and manufacturers without requiring them to have a powerful desktop system. This collaboration aspect has increased in importance as the workforce has expanded to new locations. With the ability to move 3D applications to a cloud infrastructure, clients can gain economies of scale, enhanced management, improved collaboration, and improved ROI for user workstations.

Figure 7-2 shows a generic view of an engineering cloud deployment showing three large cloud infrastructure groups: Desktop, storage, and compute clouds.

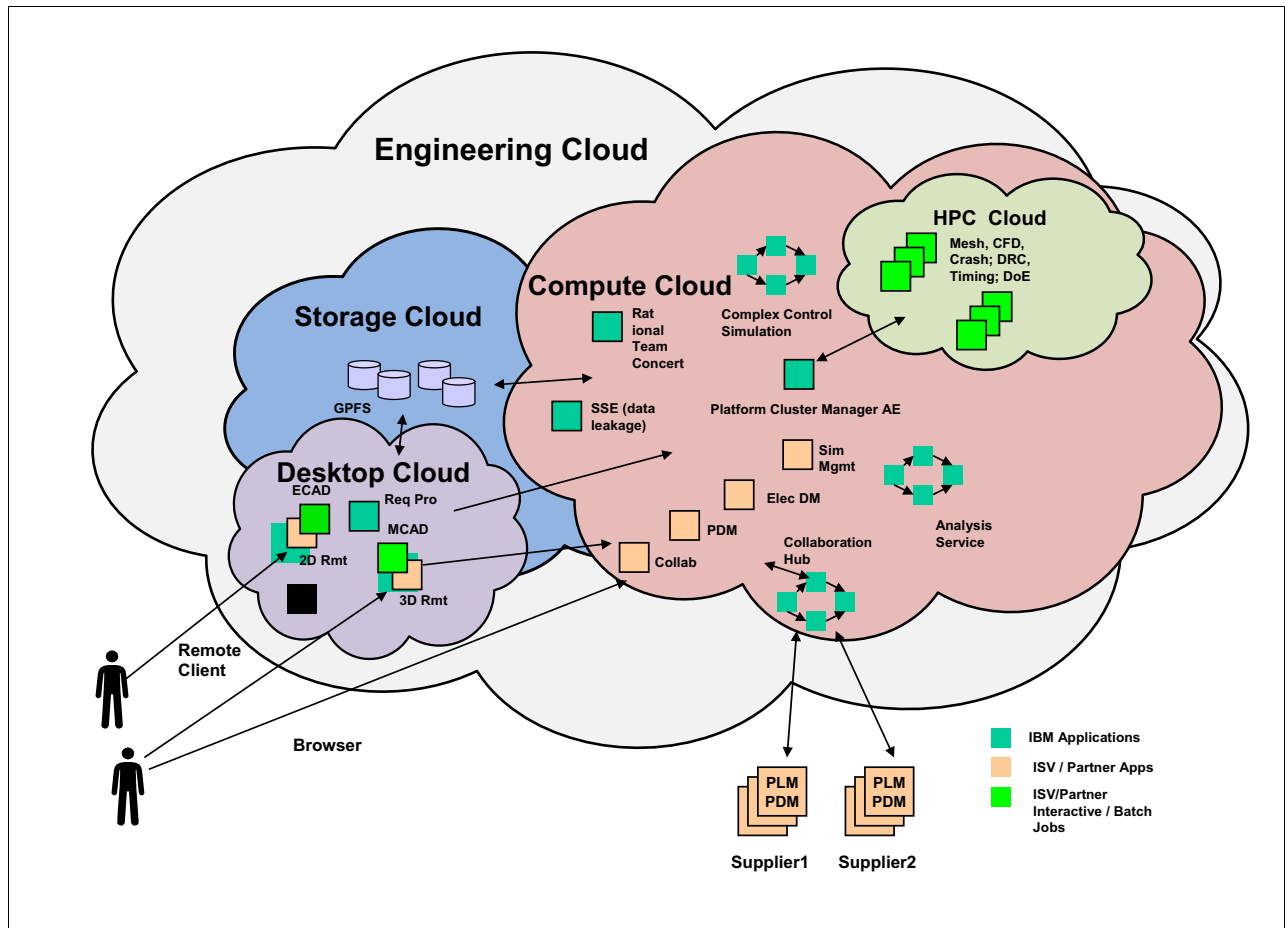


Figure 7-2 Engineering cloud solutions comprehensive deployment view

Desktop cloud

The virtualized 2D or 3D desktop cloud solution allows pre-processing and post-processing to be performed by using commodity hardware such as low-cost notebooks or desktops, instead of expensive high-end workstations. The graphics processing units (GPUs) are present visualization nodes in the remote cluster, alongside the compute nodes. These visualization nodes take care of the rendering for the desktop sessions started by the thin client workstation as shown in Figure 7-3 on page 143. Only the keyboard and mouse events are sent to the 3D session remote server, and only the rendered pixels are exchanged with the client graphics display.

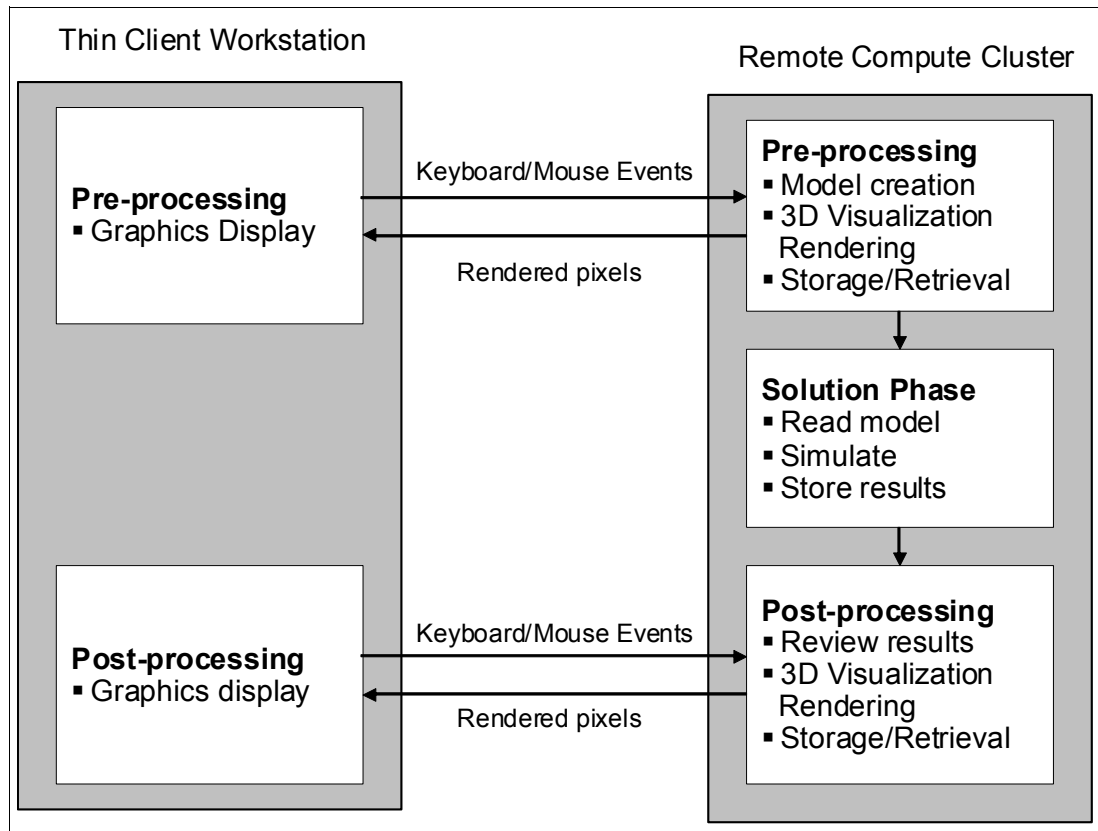


Figure 7-3 Engineering cloud workflow for 3D remote visualization

The solution has a robust but open architecture that allows the client to grow and change their engineering environment as new technologies become available. Because the compute and graphics processing is run in the cloud and commodity hardware is used to display the results, the client can easily and quickly move to newer graphics and cloud technologies as they become available. Figure 7-4 illustrates the advantages of a desktop 3D cloud infrastructure.

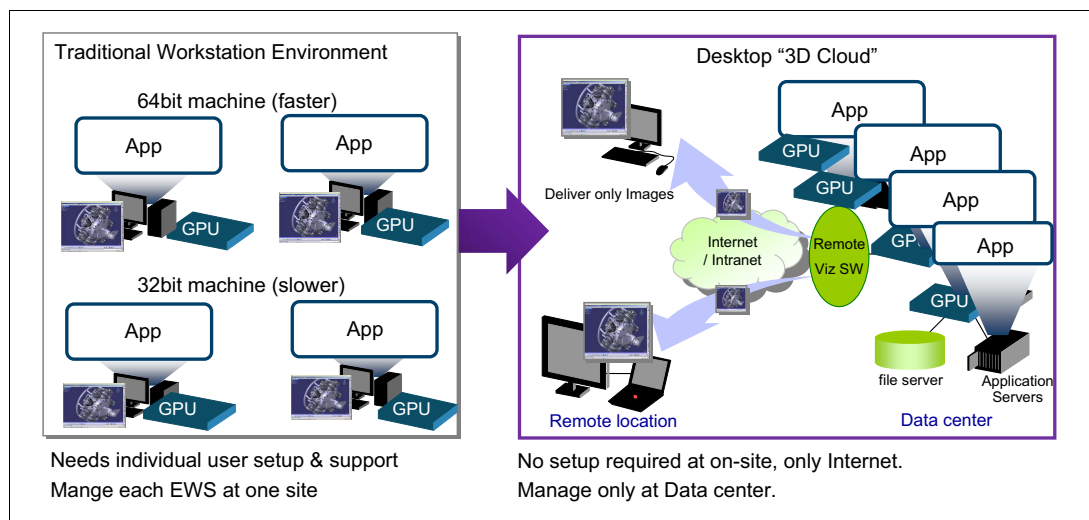


Figure 7-4 Advantages of a 3D desktop cloud infrastructure

Another characteristic of the desktop cloud is higher utilization of GPUs on visualization nodes. By running specialized virtualization software on the visualization nodes, the clients can have multiple desktop sessions on a single GPU.

Compute cloud

On the execution end of the engineering cloud solution, you have HPC clusters with robust workload and resource management for shared, highly utilized engineering environments.

A comprehensive set of management tools is needed to ensure departmental, enterprise, or community resources are optimally deployed, and easy to access and manage. The following are among the services provided:

- ▶ Self-service job submission and management.
- ▶ Easy to use web-based interface for remote access to shared resources and simplified application integration.
- ▶ Dynamic provisioning, job migration, and checkpoint-restart for automatic adaptation to changing workload requirements.
- ▶ A centralized environment where remote users run host-based HPC applications.
- ▶ Intelligent policy-based scheduling to ensure that application servers and 3D graphics are fully utilized
- ▶ Application templates to enable reuse of engineering assets.

Storage cloud

Allocating the correct amount of data storage to the correct users at the correct time is an ongoing challenge for engineering companies of all sizes. The storage cloud can enable you to cost effectively handle electronic documents such as contracts, email and attachments, presentations, CAD and CAM designs, source code and web content, bank check images and videos, historical documents, medical images, and photographs. This multilayer, managed storage virtualization solution incorporates hardware, software, and service components to facilitate simplified data access. Its seamless scalability, faster deployment, and flexible management options can help reduce complexity and costs while enabling your company's continual growth and innovation.

The storage cloud ties together the engineering cloud infrastructure. Centralized, shared storage allows engineers to access information globally. This is much different from typical environments, in which engineers manage files locally and move them back and forth to HPC resources located elsewhere. For certain types of software simulations, uploading files can take one or two hours, and file downloads can take significantly longer. This can compromise productivity and impede the use of HPC.

7.1.3 Key benefits

The IBM solutions for engineering clouds enable web portal access to centralized engineering desktops and workload optimized private or private hosted HPC clouds. The engineering cloud focuses on mechanical, electronics, and software development domains, and the seamless integration between these domains, including original equipment manufacturer (OEM) and supplier collaboration. With accelerated 2D and 3D remote graphics and modeling, agile systems and workload management, and independent software vendor (ISV) integration, the engineering cloud can help you address multiple customer challenges. These include reducing IT costs, increasing IT flexibility, improving engineer collaboration, and saving engineering time. The following section lists the key benefits of this model.

Distributed and mobile workforce

The following are the benefits of the distributed and mobile workforce model:

- ▶ Reduces the time that is needed to complete a design through improved collaboration and skill sharing.
- ▶ Allows outsourced and off-shored design staff while storing data and applications centrally.
- ▶ Remote collaboration between locations and with external third-party partners.
- ▶ Ubiquitous access to the user infrastructure.
- ▶ Unlocks designer skills from any location with remote access capabilities.

IT infrastructure management complexity

The following are the benefits of the IT infrastructure management complexity model:

- ▶ Transforms siloed environments into shared engineering clouds, private and private-hosted initially, changing to public over time.
- ▶ Increases infrastructure flexibility.
- ▶ Decreases dependence on costly high-end engineering workstations.
- ▶ Supports an infrastructure that is independent of hardware platforms and operating systems.
- ▶ Reduces branch office IT support requirements.
- ▶ Uses ideal tools, and standardizes processes and methodologies.
- ▶ Realizes improved operational efficiency and competitive cost savings.

Security control

The following section describes the benefits of the security control model:

- ▶ Patch compliance enhanced because the operating system and applications are centrally managed.
- ▶ Manages security risks in data and infrastructure.
- ▶ Centralized compliance with regulations.
- ▶ Provides greater and more secure access to compute and storage resources.

Cost of workstation management

The following section describes the benefits involving the cost of workstation management:

- ▶ Eases deployment/support of desktop systems.
- ▶ Makes IT costs predictable.
- ▶ Increases operational flexibility.
- ▶ Increases resource utilization (processor, GPU).
- ▶ Lowers TCO and rapid ROI.
- ▶ Improves procurement usage in purchasing of software.
- ▶ Achieves significant energy savings.

Note: The benefit of the engineering cloud can only be realized if the organization uses CAD/CAE applications that adhere to OpenGL standards for graphics delivery.

7.2 Architecture

The engineering cloud solution addresses the increasing demand for faster results in the Technical Computing space.

By placing the high-end graphics applications in an IT cloud infrastructure and providing access to those applications, IT can more effectively maintain a consolidated graphics hardware environment and provide graphics capabilities to new and existing users.

The self-service portal that is used in this solution provides the user with a means to find available engineering applications and engineering assets. Users are able to perform daily work in much the same way as they did in the past. The self-service portal provides administrators with the ability to define and administer engineering users. They can grant those users access to specific engineering applications and determine who can share a design.

The engineering cloud can be an extra service offered by almost any cloud environment already in place. This solution can use existing security services such as LDAP, Directory Services, and IP tunneling. The existing enterprise business administration and management systems can also be used to monitor the systems that are used to provide the engineering cloud services.

Note: The engineering cloud solution concepts described here use the Cloud Computing Reference Architecture (CCRA). It is predominately an infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS) solution. Note that although the engineering cloud solution is an IaaS component, there are multiple software products within it. The integration and maintenance of these products provides a potential for services.

7.2.1 Engineering cloud solution architecture

The architecture that supports the engineering cloud solution is robust yet relatively simple. This architecture supports open standards and can integrate with currently available security and management mechanisms. Although the implementation described here focuses on high-end engineering 3D graphics, it can be used as a platform for other types of 3D and 2D graphics applications.

Figure 7-5 represents the architecture overview diagram of the engineering cloud solution. It shows how the environment can be incorporated into an existing cloud environment and uses existing security, business management, and monitoring. Notice how this architecture relates back to the CCRA.

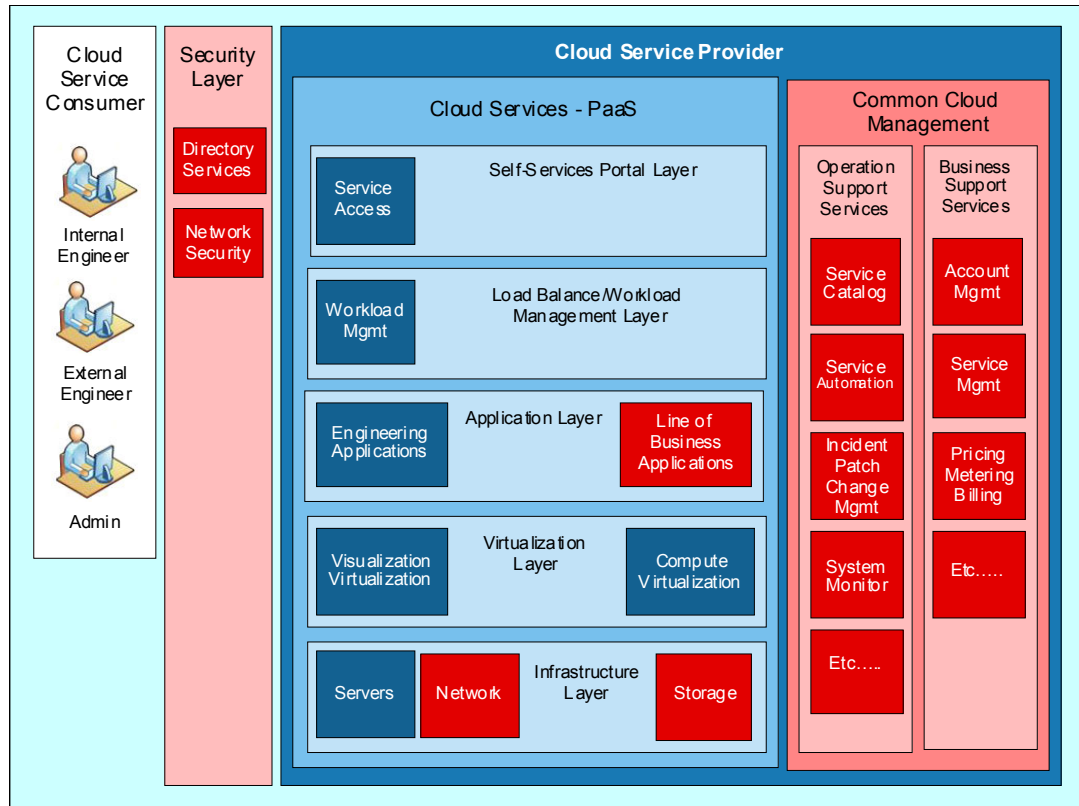


Figure 7-5 Engineering cloud architecture overview

The items shaded in blue in Figure 7-5 are directly associated with the engineering cloud solution. The items shaded in red are layers and items that most likely already exist in your environment. Any of these items that do not exist in your environment must be added for the engineering cloud solution to be deployed and managed properly.

Cloud service consumer

In the context of an engineering cloud, the consumer is any engineer who needs access to the engineering models. These engineers can be part of a client's organization or part of a third-party organization. They can be local to a single client office or dispersed globally among multiple client offices or third-party partners. Third-party users or collaborators are just another form of cloud consumer user.

Cloud services

The PaaS block in the CCRA is the most prominent piece of the solution. This block represents the specialized hardware and software that are required for this solution, which is provided as a platform to the cloud service consumers.

Depending on how the client sets up their cloud infrastructure, and how the engineering cloud has been inserted, the solution can also be in the IaaS block of cloud services.

Table 7-1 describes generalized definitions to determine whether the engineering cloud is an IaaS or PaaS service.

Table 7-1 Cloud service consumer characteristics

Cloud service consumer characteristics	Type of service employed
Has access to and can manipulate the operating system of the Cloud services provided.	IaaS
Only has access to the engineering applications and cannot manipulate the operating system of the cloud services provided.	PaaS

Note: The engineering cloud solution can be set up as both IaaS and PaaS in the same cloud environment, depending on the level of access the cloud service consumer requires and the applications being used.

Infrastructure

Because the engineering cloud solution uses specific hardware, there is a strong connection with the infrastructure block as shown in Figure 7-5 on page 147. For more information about the hardware that is required for this solution, see 7.3.6, “Hardware configuration” on page 154.

Common cloud management platform

The engineering cloud solution maps to every block within the operational support services (OSS) and business support services (BSS) of the CCRA. It is important to note here that if you have an existing cloud environment, the engineering cloud solution becomes just another service. Therefore, this solution can take advantage of the processes and utilities already in place for OSS and BSS services in your environment.

7.3 Components

Figure 7-6 takes the architecture overview from 7.2.1, “Engineering cloud solution architecture” on page 146 and adds the components of the solution. As before, the items shaded in blue are directly associated with the engineering cloud solution. The items shaded in red are layers that, most likely, exist in your environment.

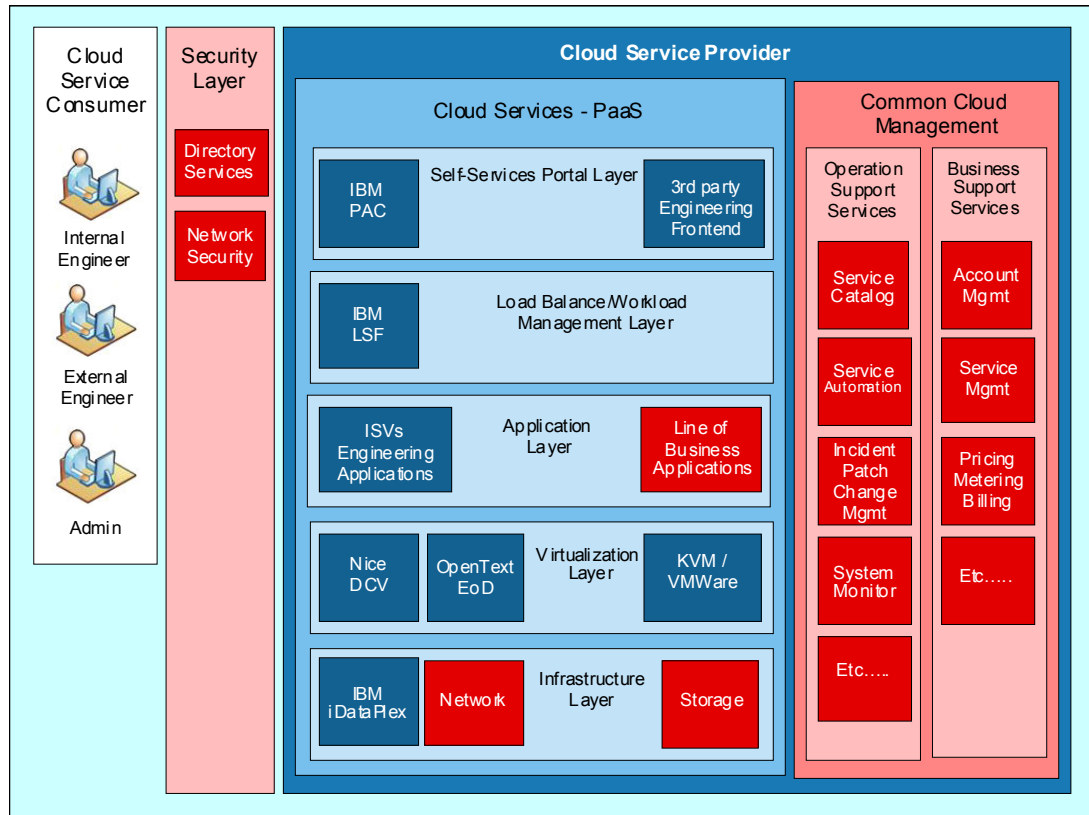


Figure 7-6 Engineering cloud PaaS component model

There is room for considerable flexibility in the environment to accommodate specific customer requirements in this architecture. The hardware and software components in this architecture can be implementing using the components described next.

7.3.1 Cloud service consumer

Various internal and external engineers as well as the internal administrators need access to the engineering cloud solution. These engineers and administrators can use various clients to access, monitor, and administer the environment. The access clients can be web browsers, third-party engineering applications (front ends), and others. Other thick and thin clients can be used in this channel as required by line of business applications. Remember that an engineer can be local, remote, or belong to a third-party partner. There is no particular IBM hardware or software that is used here. However, note that this layer runs on commodity hardware instead of high-end graphics hardware.

7.3.2 Security layer

The applications already in use by the customer to control, monitor, and administer their security environment. This includes, but is not limited to, Directory Services, Network Monitoring, employment of encryption, and so on.

This layer can use the breadth of the IBM security portfolio to help ensure trusted identities, manage user access, secure applications and services, and protect data and infrastructure. For more information about security and IBM Platform Computing, see chapter 5 of *Security of the IBM Platform Computing Integration Solutions*, SG24-8081.

7.3.3 Cloud services provider

There are multiple items and layers represented in the CCRA that can be used as part of the engineering cloud solution. The bulk of these items comprise the cloud service provider area of that architecture. The cloud service provider area contains the cloud services and common cloud management areas of the architecture overview diagram.

Remember that the solution can be deployed as a PaaS cloud service solution or as an IaaS cloud service solution. The deployment model that is used depends on your requirements and engineering applications. The PaaS deployment model was chosen in Figure 7-6 on page 149 because that is the most common deployment of the engineering cloud solution. Five layers are present in the cloud deployment model as described in the following sections.

Self-service portal layer

A self-service portal is used for user and administrative access to the engineering applications. The *IBM Platform Application Center (PAC)* is an example of such portal. PAC is the suggested portal environment for controlling and accessing engineering applications that are delivered by using the engineering cloud solution. However, it is possible to use other front ends provided by third-party engineering software suites.

Load balancing/workload management layer

The workload management application is required to use the available graphics cloud based resources and spread the work evenly across the available compute resources. The load balancer must be tightly integrated with the 3D desktop virtualization applications so that the GPUs are properly load balanced.

IBM Platform Computing Load Sharing Facility (LSF) is the load balancing engine that can be used to spread engineering workloads across the appropriate compute nodes in the cloud. This application can be found at the Load Balance/Workload Management layer of the architecture overview diagram.

Application layer

Various engineering and non-engineering applications that are managed by the portal and used by the clients. The solution presented here focus on high-end 3D engineering applications, but other 2D engineering and even non-graphics applications can be implemented in this environment. Although there can be IBM applications in this layer, there are no specific IBM products for the engineering cloud in this layer. However, IBM has partnerships with both Siemens and Dassault that provide appropriate and compliant 3D graphics applications.

Virtualization layer

Various components are used for the virtualization layer. There are two different types of virtualization in this solution.

First is the application that encapsulates and compresses the OpenGL graphics for transport to the user clients. Both NICE DCV and OpenText Exceed onDemand (EoD) are third-party applications that are supported to provide this virtualization layer in the engineering cloud solution. For more information about these applications, see 7.3.5, “Third-party products” on page 153.

The other type is the application that manages the compute nodes (virtual machines) in the environment. The example environment uses KVM.

Using KVM allows you to run more than one CAD virtual desktop on a physical server. When used with NICE DCV (Figure 7-7) you can share the GPU among multiple virtual desktops.

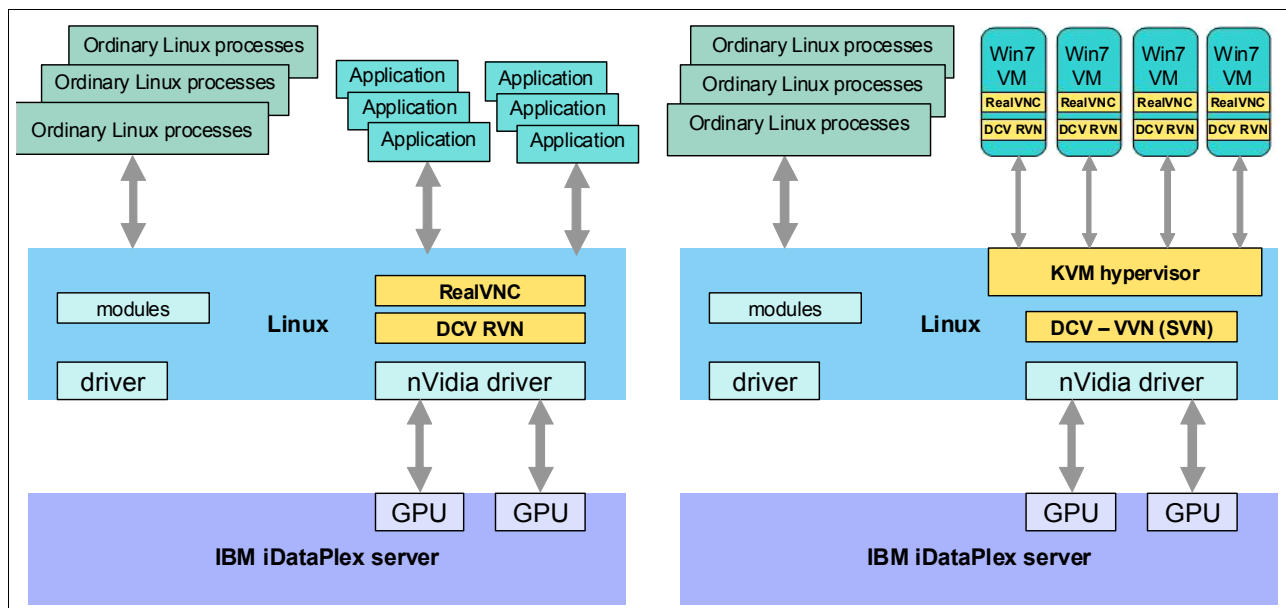


Figure 7-7 DCV on Linux and on KVM

NICE DCV supports GPU sharing or virtualization when used on application running on Linux and KVM virtual machines (along with GPU native access when running on Microsoft Windows).

DCV and EoD support different types of deployment. Table 7-2 summarizes the features of each product.

Table 7-2 3D Virtual desktop solution comparison

Feature	NICE DCV	OpenText Exceed onDemand
Guest operating system support	Supports both Linux and Windows (using KVM)	Supports OpenGL applications running on Windows, Linux, and AIX application servers
Multi-user sharing of GPUs	Supported	Supported

Feature	NICE DCV	OpenText Exceed onDemand
Direct3D support	Not supported	Not supported
Portable devices support	Limited support (required for better WAN performance)	Supports portable devices (iOS, Thin clients)

Infrastructure layer

The server hardware, network hardware, and storage devices needed to connect the engineers to the required virtual hardware and the necessary intellectual property (models) needed to perform their jobs. For more information about the hardware configurations that are required to implement the engineering cloud solution, see 7.3.6, “Hardware configuration” on page 154.

Note: IBM iDataPlex dx360 was the base hardware solution used for the example environment. For more information about the IBM iDataPlex family of servers, see:

<http://www-03.ibm.com/systems/x/hardware/rack/dx360m4/>

Several IBM storage solutions can be added to this layer such as Scale Out Network Attached Storage and General Parallel File System (GPFS). Note that the network component in the infrastructure layer of this model must be robust enough to handle the clients' bandwidth needs.

Within the IBM portfolio, the Storwize® V7000 Unified and Scale Out Network Attached Storage are two network-attached storage systems that offer storage capabilities with a data-centric view of resources and shared data repositories. These systems can be co-located with the compute resources to optimize engineering workflows and enable increased collaboration through remote access.

Both systems are built on *IBM Active Cloud Engine™*, which is a powerful policy-driven engine that is tightly coupled with the file system and designed for managing massive amounts of data. Specifically, it manages files in an automated, scalable manner, creating the appearance of a single, fast system regardless of differences in geography, storage media, and other physical factors. IBM Active Cloud Engine enables users to search huge amounts of data, and rapidly store, delete, distribute, and share these data. This is important for engineering software users because it gives users the ability to manage large numbers of files efficiently, locate relevant data quickly and move the data to where it is needed seamlessly.

For more information about file system management, in particular Active File Management (AFM), caching, replication, consistency, sharing, and other topics, see Chapter 6, “The IBM General Parallel File System for technical cloud computing” on page 111.

Note: For more information about the solution and its components, see the solution brief “IBM Engineering Solutions for Cloud: Aerospace, and Defense, and Automotive,” DCS03009-USEN-01, at:

<ftp://ftp.software.ibm.com/common/ssi/ecm/en/dcs03009usen/DCS03009USEN.PDF>

7.3.4 Systems management

The PAC self-service portal is the primary system management facility for the engineering cloud solution. This portal is the service that is used to define available applications, and the

users that can access them. But PAC can also be part of a broader, larger infrastructure layer that is managed by *IBM Platform Cluster Manager - Advanced Edition (PCM-AE)*.

The environment administrator can take advantage of the application templates feature in PAC to define job submission forms for each type of application that is used by the engineering teams. The customizable interface builder enables forms to be easily tailored and selectively shared among users or groups.

Another system management aspect to the engineering cloud solution is the duties/tasks that are performed by the system administrators. The engineering cloud solution requires a specific set of hardware and software resources. Due to these specific resource requirements, the administrators might need to modify existing practices to ensure that the engineering cloud and the engineering sessions created in that environment are provisioned and deployed properly. This is necessary when the engineering cloud solution is deployed as a *cluster definition* in an existing cloud infrastructure managed by PCM-AE. For more information about PCMAE cluster definitions, see 5.5.1, “Cluster definition” on page 96.

For instance, If the engineering cloud solution is defined as a cluster definition that can be provisioned into an existing PCM-AE cluster infrastructure, that definition must provision a set of engineering specific resources from the available pool of resources. This must be do to ensure that engineering sessions have access to the appropriate hardware and software.

However, if the engineering cloud solution is employed as a stand-alone cloud service, the IBM Platform LSF product runs these provisioning functions. There is only one set of hardware and software resource pools. If the dynamic cluster plug-in is installed, advanced provisioning features are employed to meet software requirements for the engineering jobs. The advantages of dynamic cluster are explained in 2.2.5, “IBM Platform Dynamic Cluster” on page 18. Again, multiple resource pools might be required based on your requirements.

Although the self-service portal adds more management requirements to the existing infrastructure, it eliminates the need to manage individual high-end engineering workstations throughout the enterprise.

7.3.5 Third-party products

This section provides third-party products that are useful when running engineering workloads.

Nice desktop Cloud Visualization (DCV)

This application encapsulates, compresses, and transmits all of the OpenGL graphics information that is used in this solution. For more information, see:

<http://www.nice-software.com/products/dcv>

OpenText Exceed onDemand

Exceed onDemand is a managed application access solution that is designed for enterprises. The solution offers pixel drawing, low-cost scalability, and trusted security access over any network connection. For more information, see:

<http://connectivity.opentext.com/products/exceed-ondemand.aspx>

Real VNC Enterprise Visualization

This application allows you to connect to remote sessions (3D Engineering or otherwise). RealVNC is used to connect the engineer with the engineering cloud resources needed. RealVNC establishes connections between computers irrespective of operating system. The

RealVNC Viewer is installed on the engineer's lightweight workstation, and the RealVNC Server is installed on the visualization nodes in the cloud. For more information, see:

<http://www.realvnc.com/products/>

3D Engineering CAD/CAE applications

The CAD/CAE applications are not necessary for the solution itself, but are necessary to realize the ROI gained by virtualizing physical high-end engineering workstations. All of these applications are in the application layer of the architectural overview diagram. For more information about ANSYS, a 3D Engineering CAD/CAE application that was used in the example environment, see:

<http://www.ansys.com/>

7.3.6 Hardware configuration

The engineering cloud solution uses a specific hardware stack to provide appropriate compute and high-end graphics capabilities in the cloud.

Shared memory systems

Some engineering workloads require large shared memory system to achieve good performance. Therefore, it is important to select the correct server family, processor, and memory so that applications can operate efficiently. For computation, Intel Xeon E5-2600 series (or E5-4600 series), 8-core and 2.6 GHz or faster processors are preferable. Configure sufficient memory using the latest direct inline memory module (DIMM) technology, which offers speeds up to 1600 MHz, so that problems are solved in-core. This eliminates the risk of bottlenecks due to slow I/O.

Clusters and scalability

When one system (node) is not sufficient to solve an engineering problem, multiple nodes are connected with a communication network so that a single problem can be run in parallel. In this situation, the communication delay (latency) and rate of communication among systems (bandwidth) affect performance significantly. IBM server products support InfiniBand switch modules, offering an easy way to manage high performance InfiniBand networking capabilities for IBM server systems.

Storage systems

This section describes the storage systems as part of the hardware solution for storage.

IBM Storwize V7000 Unified

The IBM Storwize V7000 Unified storage system can combine block and file storage into a single system for simplified management and lower cost. File modules are packaged in a 2U rack-mountable enclosures, and provide attachment to 1 Gbps and 10 Gbps NAS environments. For block storage, I/O operations between hosts and Storwize V7000 nodes are performed by using Fibre Channel connectivity. The following are the relevant features:

Number of disk enclosures	Up to 10
Size of each enclosure	24 2.5" 1 TB nearline SAS 7.2 K RPM = 24 TB
Total disk capacity of the system	240 TB
Host attachment – File storage	1 Gbps and 10 Gbps Ethernet
Host attachment – Block storage	SAN-attached 8 Gbps Fibre Channel (FC)

Entry-level file server with internal storage system

For small or economical environments, an IBM System x3650 M4 can be used as an NFS file server. The x3650 M4 system contains up to 16 internal 1 TB 7.2 K RPM nearline SAS drives with RAID6 configuration. The file system is mounted over the fastest network (Gigabit Ethernet or InfiniBand) provided in the cluster.

Cluster interconnect

An important constraint to consider is network bandwidth and capacity. Although the engineering solution provides great value, it also increases network traffic. This increase in network traffic is because graphics used in the engineering design process are now transported through TCP/IP. When considering this solution, you need to understand how design engineers currently use their graphics applications (length of design effort, frequency of user access, model size, number of simultaneous users, and so on). Conduct a network traffic assessment with your IT staff to ensure that your network infrastructure will be able to handle the projected workloads deployed onto the cloud infrastructure. It might be advantageous if you have a globally dispersed engineering staff to consider the configuration of multiple engineering cloud compute clusters.

Network latency also plays a significant role in the performance of the Engineering 3D Cloud solution. If there is network latency greater than 40 ms between the engineer and the engineering cloud, performance of the graphics applications suffers. The applications still work properly in these instances, but the graphics images will not move smoothly, which might cause user complaints.

Best practices

The specific systems configuration for your engineering solution depends on the workload type and application characteristics and requirements. Here are some configurations tips for servers dedicated to scale-out workloads.

Systems

Configure the systems as follows:

- ▶ 2-socket-based systems with GPU support
 - IBM iDataPlex dx360 M4
 - IBM PureFlex™ x240
- ▶ 2-socket-based systems without GPU support
 - IBM System x3550 M4

Processor

The processor has the following characteristics:

- ▶ 2-socket systems
 - Intel Xeon E5-2670 2.6 GHz 8 Core

Memory

Allocating sufficient memory to solve in-core improves performance significantly. Consider this first before you add other resources such as more cores or GPUs. The following are the configuration characteristics (Table 7-3 on page 156):

- ▶ Use dual-rank memory modules with 1600 MHz speed
- ▶ Use the same size DIMMs
- ▶ Populate all memory channels with equal amounts of memory

- ▶ A 2-socket system has eight channels
- ▶ Populate the memory slots in each channel in this order:
 - First slots in all memory channels
 - Second slots in all memory channels

Table 7-3 Recommended memory configurations

Total memory per node	2-socket systems
64 GB	8 x 8 GB DIMMs
128 GB	16 x 8 GB DIMMs
256 GB	16 x 16 GB DIMMs

GPU accelerators

The following IBM systems are enabled for GPU usage:

- ▶ IBM System dx360 M4 with up to two NVIDIA
- ▶ IBM PureFlex System x240 with up to one NVIDIA

Supported GPUs for acceleration:

- ▶ NVIDIA M2090
- ▶ NVIDIA K10
- ▶ NVIDIA K20 and K20X

IBM has published solution guides that are focused on specific engineering ISVs, for example ANSYS. These documents address in more detail each application requirement, providing hardware configuration best practices. See the documents listed in Table 7-4.

Table 7-4 IBM solution and best practice guides for engineering

Publication ID	Publication name	URL
XSO03160-USEN-00	<i>IBM Information Technology Guide For ANSYS Fluent Customers</i>	https://storage.ansys.com/corp/2012/April/it/it_guide.pdf
XSO03161-USEN-00	<i>Best Practices for Implementing ANSYS Fluent Software on Cluster Technologies from IBM</i>	https://storage.ansys.com/corp/2012/April/it/best_practice.pdf
TSS03116-USEN-0	<i>ANSYS and IBM: optimized structural mechanics simulations</i>	http://www.ansys.com/staticassets/ANSYS/staticassets/partner/IBM/IBM-ANSYS%20Structural%20Mechanics%20Solution%20Brief.pdf
TSS03117-USEN-00	ANSYS and IBM: agile, collaborative engineering solution	http://public.dhe.ibm.com/common/ssi/ecm/en/tss03117usen/TSS03117USEN.PDF

Example configuration

The example environment involved a development cluster configured to provide 3D engineering cloud access to run ANSYS applications. The cluster solution was set up according to the reference architecture described in this book.

Compute node 2-socket system without GPU

The following section describes the hardware and software for the compute node 2-socket system without GPU:

- ▶ Hardware
 - iDataPlex dx360 M3
 - 16 GB RAM
- ▶ Software
 - RedHat Enterprise 6.2
 - DCV 2012.0.4557
 - VNC Enterprise Visualization

Compute node 2-socket system with GPU

The following section describes the hardware and software for the compute node 2-socket system with GPU:

- ▶ Hardware
 - iDataPlex dx360 M3
 - 2 NVIDIA Quad 5000 GPU
 - 192 GB RAM
- ▶ Software
 - RedHat Enterprise 6.2
 - DCV Version 2012.2-7878
 - EoD Version 8

The resources dashboard in Figure 7-8 shows a rack view of the nodes dedicated for the engineering environment. This view in PAC provides the status of each node. The user is logged in as **wsadmin**, which is an administrator for this cluster instance.

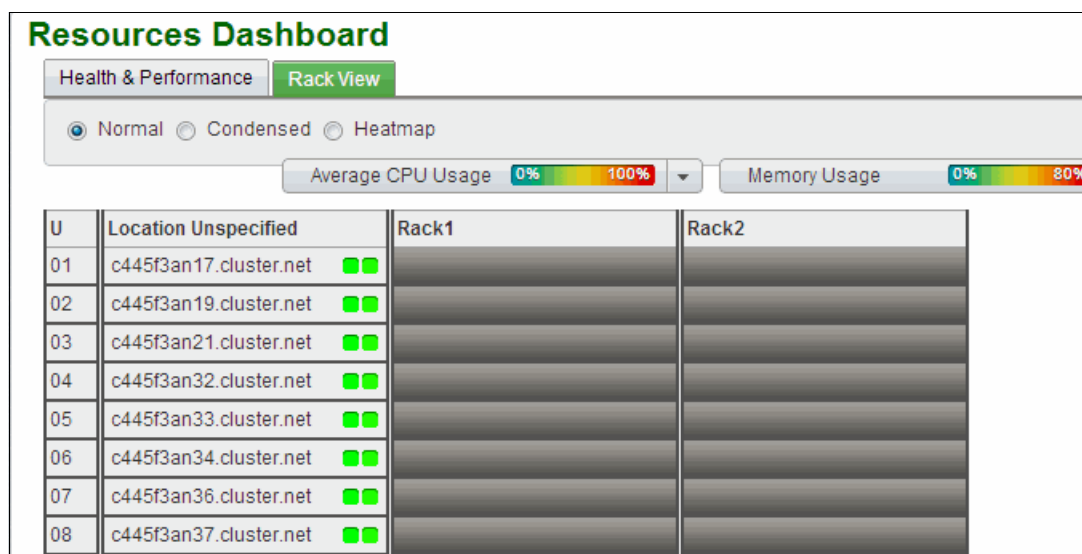


Figure 7-8 The resources dashboard in PAC showing the nodes available in the test cluster

In the hosts view in PAC, as shown in Figure 7-9, you can see the details for each node in the cluster.

IBM Platform Application Center 9.1

Isfadmin
(Administrator)
| Log Out | Help

May 23

Hosts

Open Host
Close Host

Filter : Off

<input type="checkbox"/>	Host Name	LSF Status	Host Resou...	CPUs	Cores	Job Slots in...	CPU Usage
<input checked="" type="checkbox"/>	c445f3an37.cluster.net	OK	cs	2	16	0	0 %
<input type="checkbox"/>	c445f3an36.cluster.net	OK	cs	2	16	0	0 %
<input type="checkbox"/>	c445f3an34.cluster.net	OK	cs	2	16	0	0 %
<input type="checkbox"/>	c445f3an33.cluster.net	OK	cs	2	16	0	0 %
<input type="checkbox"/>	c445f3an32.cluster.net	OK	cs	2	16	0	0 %
<input type="checkbox"/>	c445f3an21.cluster.net	OK	eod dcw_linux	2	16	1	0 %
<input type="checkbox"/>	c445f3an19.cluster.net	Closed_Full	eod dcw_linux	2	16	2	0 %
<input type="checkbox"/>	c445f3an17.cluster.net	OK	eod dcw_linux	2	16	1	1 %

Host: c445f3an37.cluster.net

Summary
Jobs

Host Name c445f3an37.cluster.net
LSF Status OK
CPUs 2
CPU Usage 0 %
Memory 62112 / 65512 MB
Space in /tmp 260096 / 280554 MB
Disk IO Rate 0.3 KB/s
15s/1m/15m Load 0 / 0 / 0
Job Slots in Use 0

Host Status OK
Cores 16
Login Sessions 0
Swap Space 1024 / 1024 MB
Disks 1
Idle Time 13416 min
Host Resources cs
Total Job Slots 16

Figure 7-9 The example environment shown in the PAC web interface

7.4 Use cases

This section details example use cases that were evaluated in the example environment while accessing the IBM engineering cloud deployment. These use cases use software products developed by ANSYS for computational fluid dynamics and structural mechanics.

The typical user of ANSYS applications performs the following tasks:

- ▶ Preprocessing where the engineering model is created using graphics-based applications.
- ▶ Solution phase where a simulation of the model is carried out.
- ▶ Post processing where the results are evaluated using graphics-based applications.

These tasks are shown in Figure 7-10.

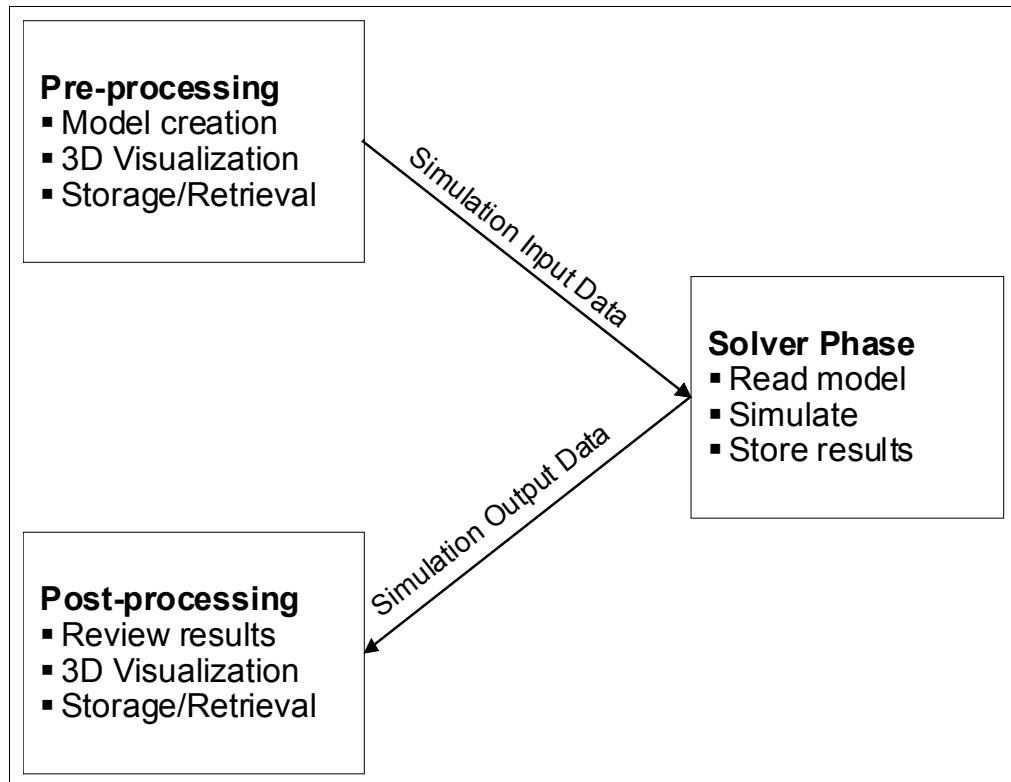


Figure 7-10 ANSYS main use case

The resource allocation to address this use case scenario can be fully distributed, partially distributed, or fully centralized. The use case scenario where the resources are fully distributed such that each workstation is self contained to address all the three steps (Figure 7-10) is not considered in the context of an engineering cloud solution. In this architecture, it is assumed one or more of the three steps are performed on a centralized resource. In practice, the main use case described in Figure 7-10 is to use centralized computing resources in the following two ways:

1. Local workstations and remote clusters

The typical use case is illustrated in Figure 7-1 on page 141. In this case, an ANSYS user prepares data on the workstation using an application such as ANSYS Workbench. The user then submits a simulation job to use the performance and throughput of the cluster. After the simulation is complete, the results are downloaded and viewed on the client workstation. ANSYS simulation software such as ANSYS Fluent and ANSYS Mechanical, which are computationally intensive, run on the cluster.

2. Thin clients and remote clusters:

As shown in Figure 7-3 on page 143, both compute intensive simulation using ANSYS Fluent and ANSYS Mechanical, and graphics intensive visualization run in the cluster environment. The client device can be a thin client with display capabilities instead of a powerful workstation. The only data that is transmitted between the client and the cluster are the keystrokes from the client and the rendered pixels from the cluster.

Note: The two use case scenarios are similar, for both ANSYS Fluent and ANSYS Mechanical. However, each of these application areas has slightly different computing, graphics, network, and storage requirements. For example, in the case of ANSYS Mechanical, the memory requirements and data that are generated for post processing during simulation can be significantly larger than for ANSYS Fluent. These differences might have some implications on the selection of cluster resources such as network bandwidth, memory size, and storage subsystem.

Desktop Cloud Visualization (DCV) and EoD are used to implement remote rendering requirements of ANSYS application suite. For more information, see 7.3.5, “Third-party products” on page 153. Their use in implementing remote 3D visualization for ANSYS application suite is described in this section. However, the internal architecture and implementation of these components is not covered in this document.

7.4.1 Local workstation and remote cluster

The use case, as shown in Figure 7-11, requires a facility to submit a batch job to the cluster by providing input data sets that are either local to the workstation or in the cluster.

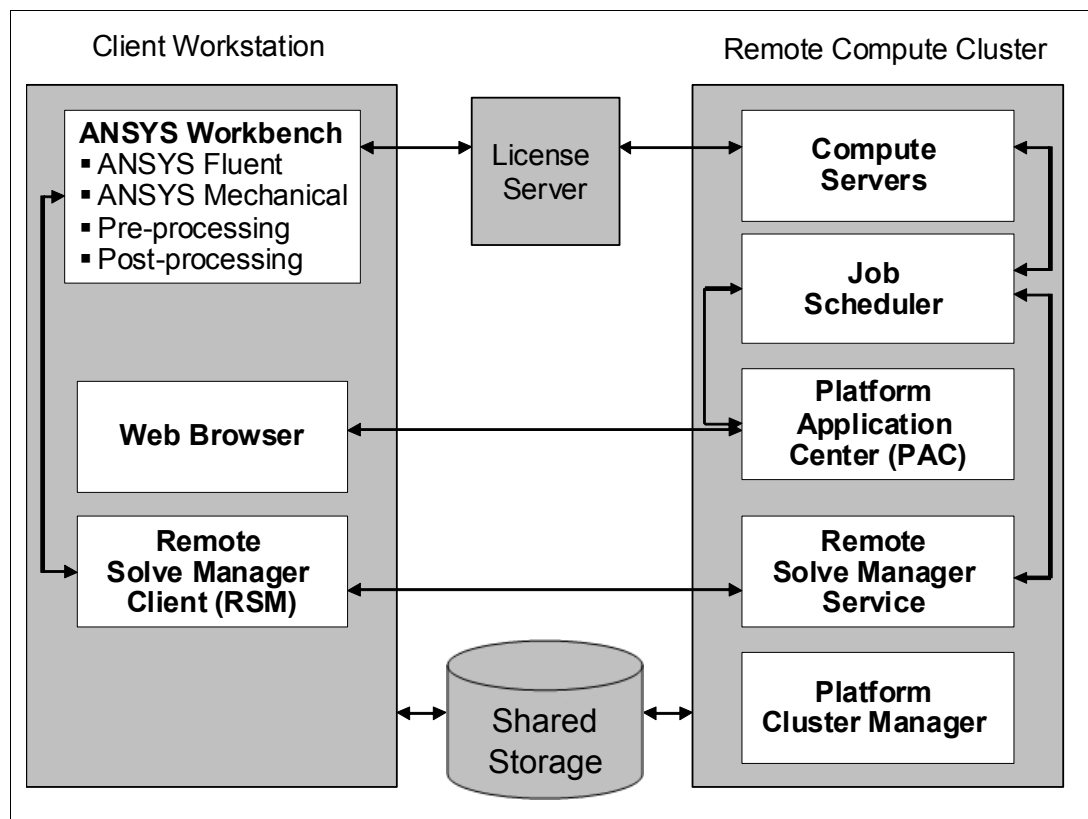


Figure 7-11 Architectural overview of the local workstation and remote cluster use case

The primary interface to the ANSYS Fluent and ANSYS Mechanical is IBM PAC. In PAC, this interface is provided through the application templates.

Each of these templates has the following information embedded in them by the system administrator so that it is transparent to the user (the resources are in the cluster):

- ▶ PATH to the application executable file
- ▶ License server information

When a new version of ANSYS application is installed, the reference to ANSYS application in PAC must be updated either automatically or by the system administrator.

ANSYS Fluent

When the users click the ANSYS-Fluent template, a web-based form is presented that requests information as shown in Figure 7-12.

The screenshot shows the IBM Platform Application Center 9.1 web interface. The top header displays the IBM logo, the title 'IBM Platform Application Center 9.1', and the user 'Isfadmin (Administrator)'. The left sidebar contains navigation tabs: 'Jobs', 'Resources', 'Settings', and 'Reports'. Under the 'Jobs' tab, a list of submission forms is shown, with 'FLUENT-TEST' highlighted. The main content area is titled 'Submit a job: FLUENT-TEST-TEST' and includes buttons for 'Submit', 'Save As', and 'Delete'. The form is divided into three sections: 'Application Parameters' (Job Name: MYFLuent, Version: 3d, Console Support: Yes, Additional FLUENT Options: -g -pinfiniband -ssh), 'Cluster Parameters' (Queue: normal, Memory Architecture: DMP, CPUs: 4, MPI Type: pcmpi, Additional Parameters:), and 'Application Data Files' (FLUENT Journal File: Add Local File / Add Server File, CAS Input File (.cas .dat): Add Local File / Add Server File). At the bottom are 'Submit' and 'Revert' buttons.

Figure 7-12 Using a modified ANSYS Fluent application template to submit a job

Note: The input data sets can either be local to the workstation or on the server.

After the users provide the information and click **Submit**, PAC constructs an LSF command that includes all the information that is needed and submits the ANSYS Fluent job to LSF. Optionally, for each job, a temporary working directory is created. If the data sets are on the workstation, they are transferred to a working directory on the cluster. All of the output, if any, that is generated is stored in the same directory. The users can retrieve the information and manage the information in this directory. The progress of these jobs can be monitored by selecting the jobs tab on the left side of the screen.

ANSYS Mechanical

The process of starting ANSYS Mechanical is similar to ANSYS Fluent. Special consideration is given to the amount of data that is generated for post processing while offloading ANSYS Mechanical jobs to the cluster from a remote workstation. These data sets can be very large,

making it difficult to transfer them over slow networks. In this case, the options are either remote visualization or high-speed connectivity to the user workstation.

7.4.2 Thin client and remote cluster

This scenario demonstrates the business value of remote visualization in the engineering cloud solution. It involves an interactive session from commodity notebooks (no graphics accelerator) from where applications that use OpenGL based graphics are started. As mentioned before, the two visualization engines that are supported in this architecture are DCV and EoD. PAC templates for DCV and EoD are created to allow the users to submit a request for a visualization session

Figure 7-13 illustrates the architecture that is explored in this use case.

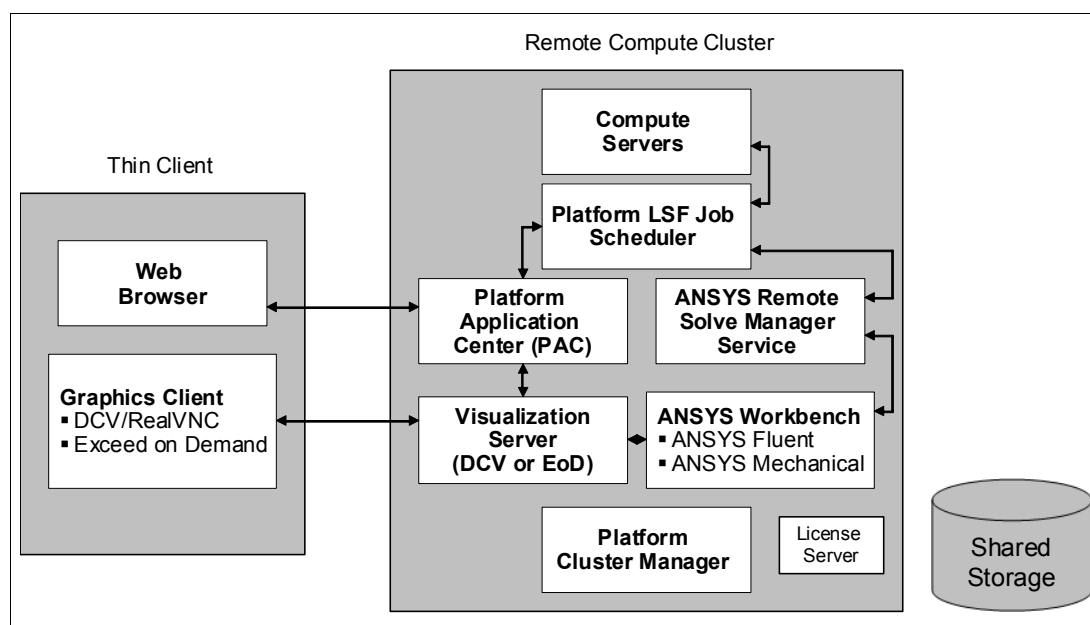


Figure 7-13 Architectural overview to support the virtualized 3D client use case

After a user has been allocated a virtual desktop, the user interacts with an application-specific graphics-intensive modeling tool such as ANSYS Workbench running on the cluster. Remote visualization is enabled by intercepting GL calls, rendering them on the server, and sending the rendered output to the client by using the X protocol. This is supported by either a combination of EoD or RealVNC, and Nice DCV, where a VNC viewer for local display is provided by all options. The configuration script provided supports any of these options. The DCV support for 3D rendering is more tightly integrated into the viewer. Both the DCV and EoD support network security, and have similar functionality. In both cases, the appropriate optimized graphics device driver is installed on the cluster.

Method #1: DCV

Users can access the VM running 3D OpenGL applications from their personal notebook or desktop by using RealVNC VE edition. Figure 7-14 shows the scenario where users access KVM guests by using RealVNC. NICE DCV server component grants GPU driver access to the KVM hypervisor, providing virtualized 3D capabilities to each VM.

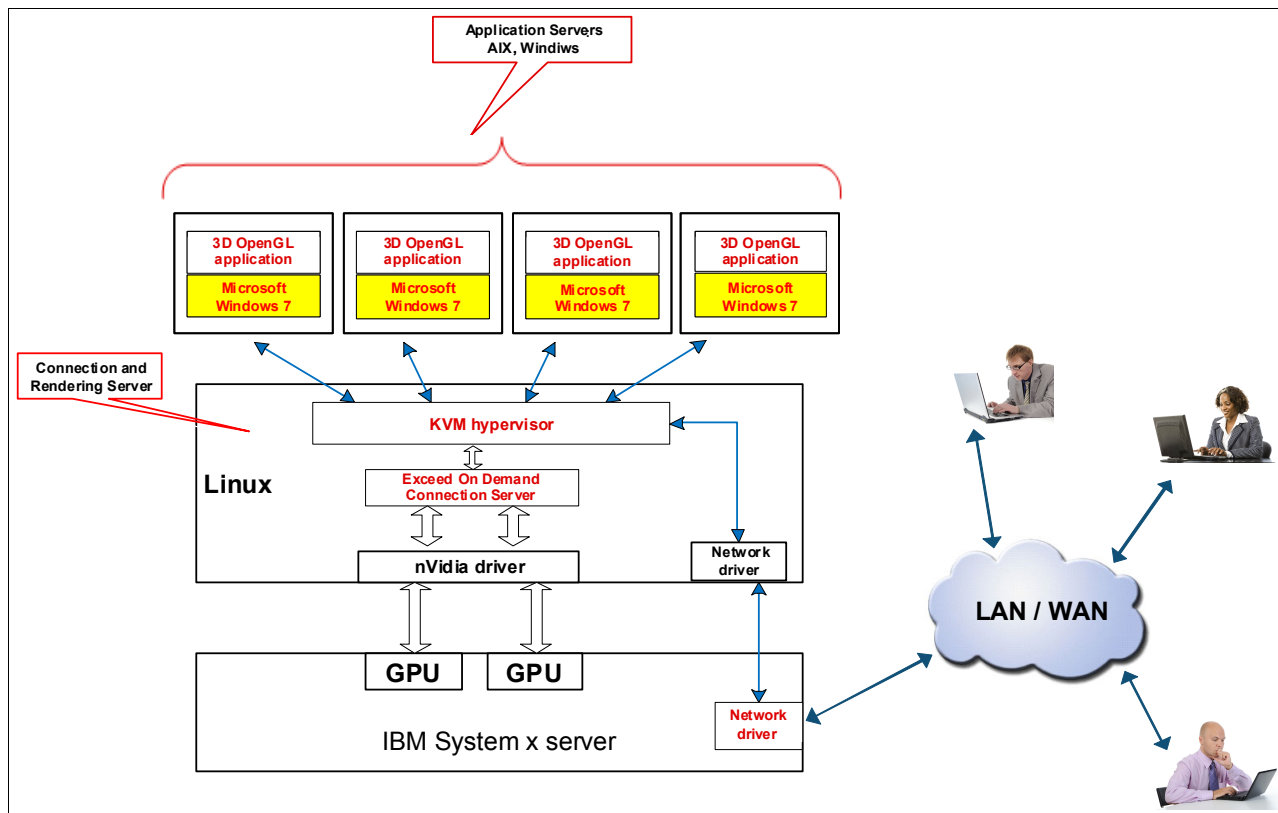


Figure 7-14 Running 3D applications on KVM using DCV and RealVNC

Workflow

When the user clicks the DCV template from the list under job submission forms, the form shown in Figure 7-15 is displayed. The command to start ANSYS Workbench is embedded into this form so that users do not have to remember the platform or application installation-specific details. For users who are familiar with the specific operating system platform and require command line facilities, a generic DCV template is provided to open an xterm window to run those operations. Providing a desktop facility allows the users to manage their data sets locally on the cluster.

IBM Platform Application Center 9.1

Submit a job: AppDCVonLinux

Submit Save As Delete

Session Parameters

Note: DCV and RealVNC VE server software must be installed on all LSF server hosts. RealVNC client must be installed on your browser host (the host you are using to access PAC). The purpose of this form is to submit a job on linux that can be managed through PAC.

Job Name MYJOB

Application Command * /home/hari/run.cmd1

Display Resolution 1024x768

Queue dcv

Submit Revert

Figure 7-15 Submitting a new job to run a 3D application using DCV on Linux

After the user submits the form, PAC requests LSF to allocate an interactive session to use DCV. Initially, the jobs are assigned the status Pending if an interactive session is not available. The job status currently shows that the job is waiting for the interactive session.

After the hardware resource is properly allocated for the job, the status changes to Running as shown in Figure 7-16.

IBM Platform Application Center 9.1

Visualize Rerun Kill Resubmit

Job ID 1613

Job Name MYJOB

Job Status Running

Pending Reasons -

More Details

Figure 7-16 Clicking Visualize activates the interactive session

The progress of these requests can be monitored by clicking the jobs tab on the left side of the window. When the interactive session is allocated, a new status icon Visualize is displayed as shown in Figure 7-16. When the user clicks this button, session information is downloaded to the workstation and the user is prompted to start a real-VNC session and provide credentials for authentication.

Figure 7-17 shows the process to get the remote session started.

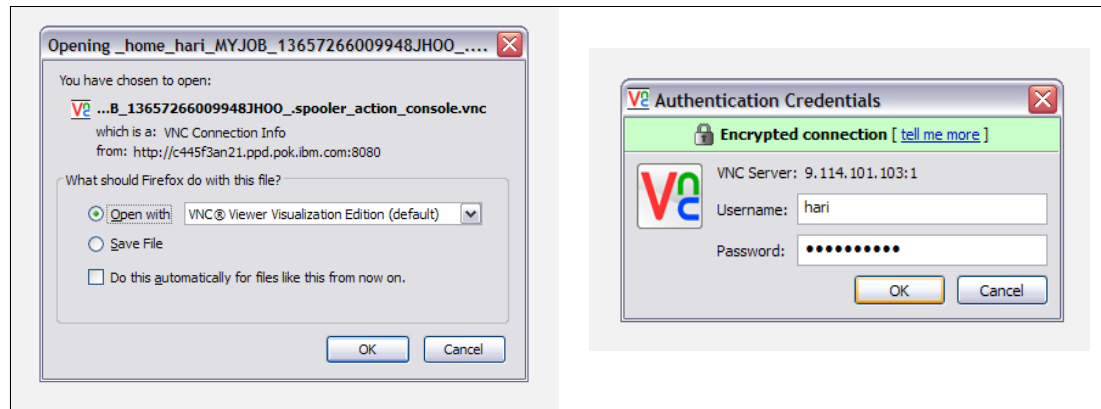


Figure 7-17 Session downloads session information and performs user authentication

After the user session is authenticated, PAC starts an ANSYS Workbench session like the one shown in Figure 7-18. From this point ANSYS users can work with ANSYS Fluent and ANSYS Mechanical. Because both ANSYS Fluent and ANSYS Mechanical use OpenGL calls for graphics, DCV intercepts these calls and runs rendering operations on the server, then compresses the bitmaps and transfers them to the real-VNC client on the user workstation. If the session disconnects, PAC does not end the session. The session can be reconnected starting from the step in Figure 7-16 on page 164 by selecting the job in progress and clicking **Visualize**.

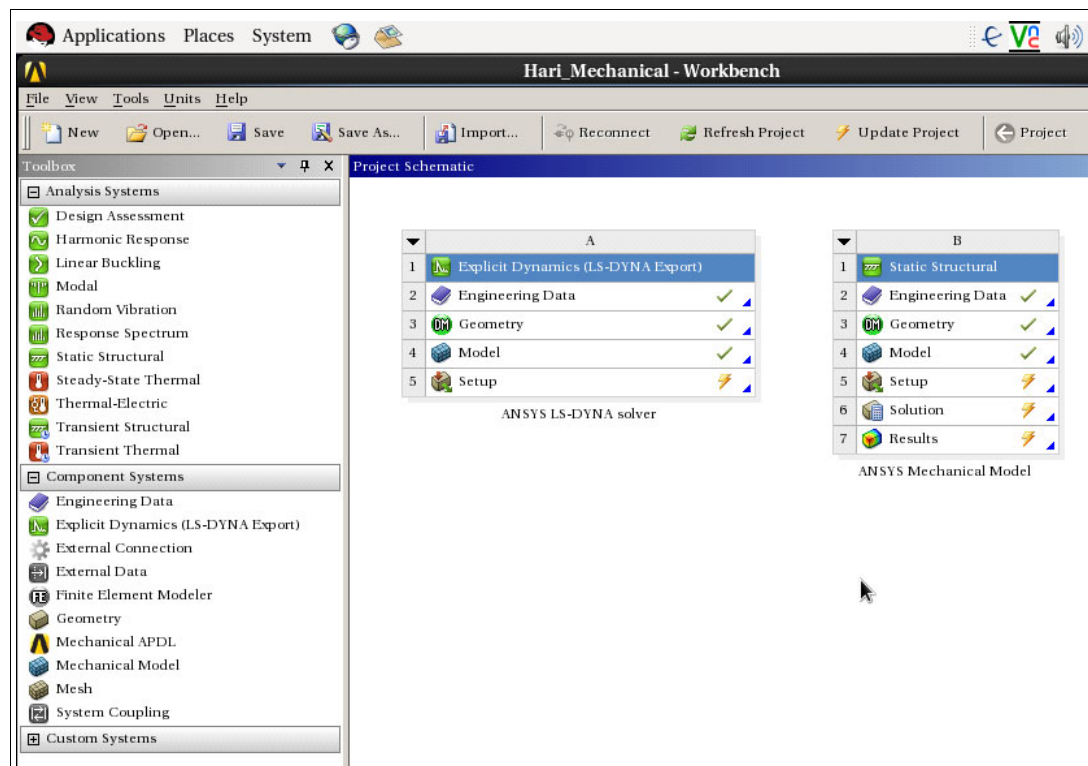


Figure 7-18 PAC automatically starts the ANSYS Workbench running on the backend

Figure 7-19 demonstrates a user working on ANSYS Mechanical through the remote desktop connection that is provided by the engineering cloud solution.

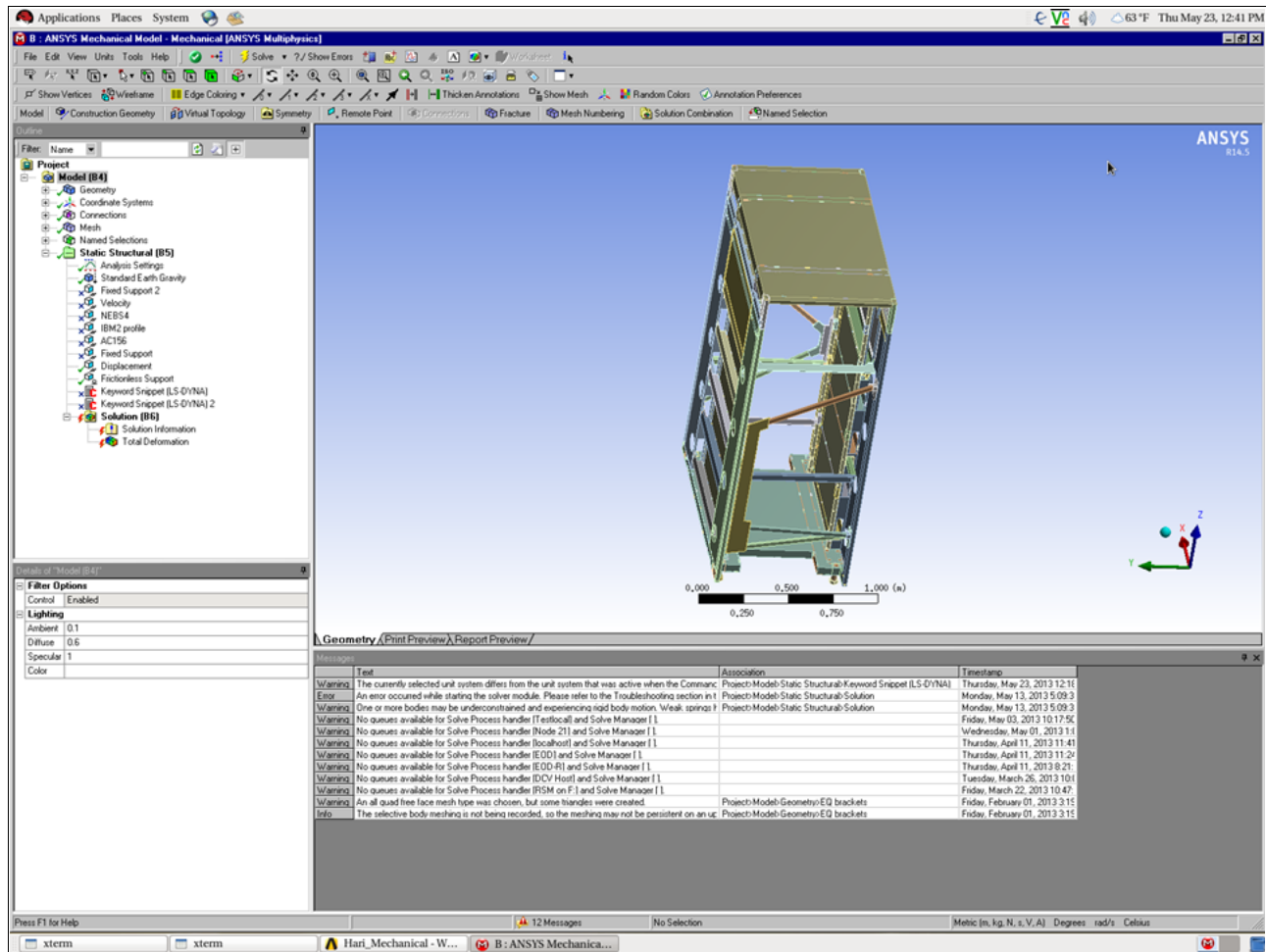


Figure 7-19 User interacts with ANSYS Fluent and ANSYS Mechanical through Workbench

After the user completes the graphics operations on the 3D models, the next step is to submit the solver jobs in batch mode to LSF to be scheduled on the back-end cluster. There are two ways that this can be accomplished:

- ▶ ANSYS Workbench uses a tool called Remote Simulation Manager (RSM) through which ANSYS Fluent and ANSYS Mechanical jobs can be submitted to an LSF cluster.
- ▶ After the input data sets are prepared, the two application templates, ANSYS Fluent and ANSYS Mechanical, can be used to submit the batch jobs outside ANSYS Workbench.

Method #2: EoD

The process of submitting a request for interactive session that uses Exceed onDemand for remote visualization is similar to DCV. However, EoD is able to provide two different deployment models.

Direct server-side rendering

Figure 7-20 shows the Exceed onDemand 3D direct server-side rendering.

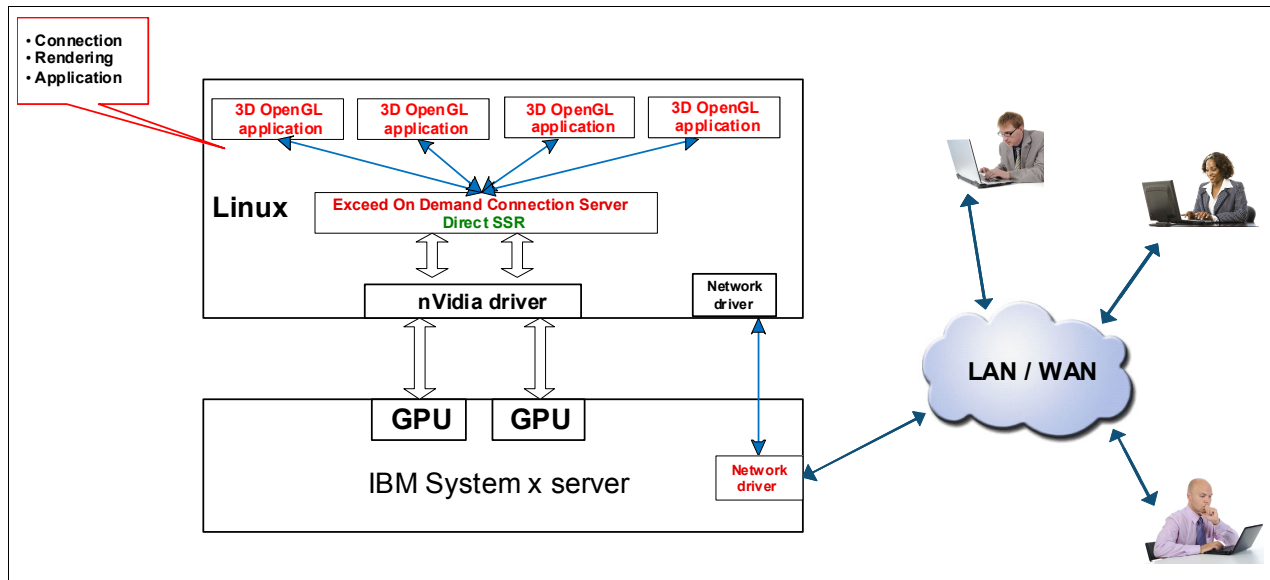


Figure 7-20 Exceed onDemand 3D: Direct server-side rendering

Indirect server-side rendering

Using Exceed onDemand allows you to run more than one CAD session per server supporting applications running on IBM AIX and Windows. Figure 7-21 shows Exceed onDemand 3D indirect Server Side Rendering with application server on Windows.

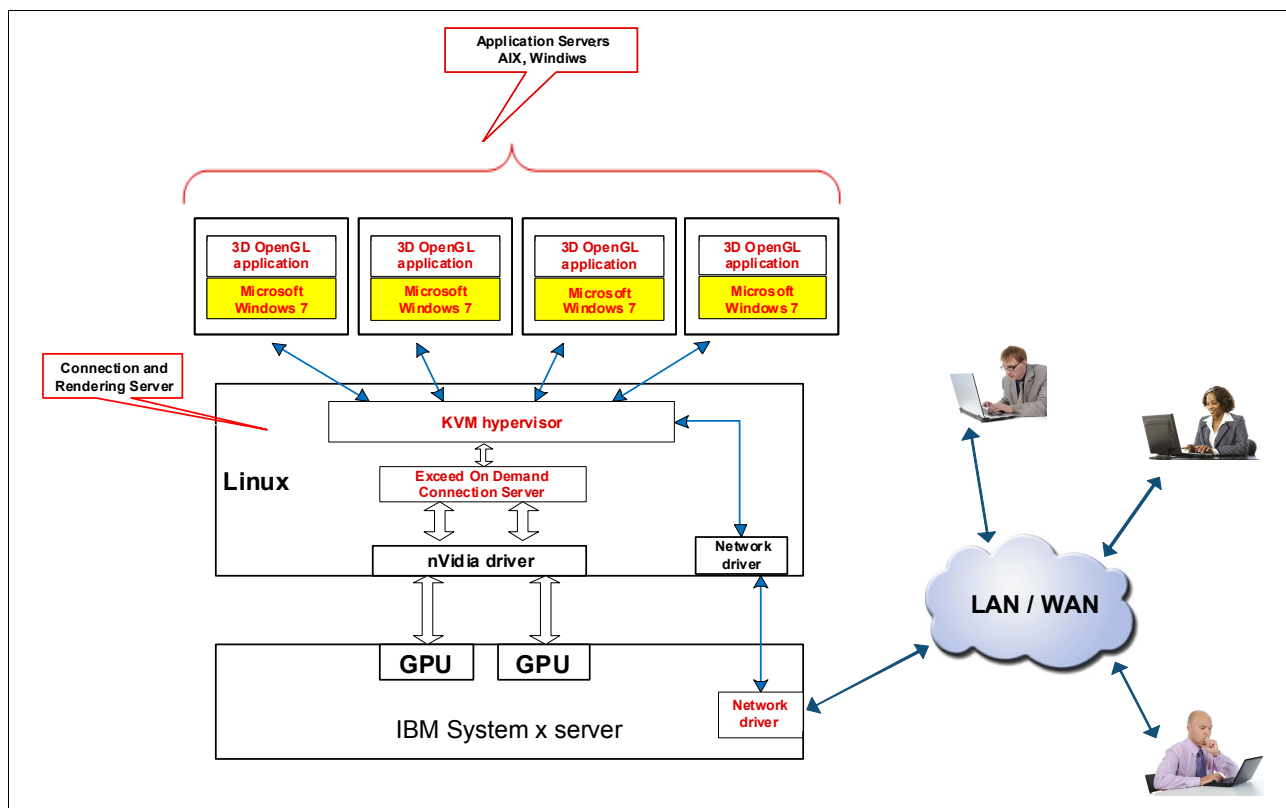


Figure 7-21 Desktop cloud architecture using KVM and EoD

Workflow

After the user submits the EoD form, PAC requests LSF to allocate an interactive session to use EoD (Figure 7-22). Initially, the jobs are assigned the status Pending if an interactive session is not available.

Submit a Job: EXCEED_Xterm_Enterprise

Form Name:

Session Parameters

Monitor Job Name:

Reusable:

Submit to this Queue:

CPU:

Memory(m):

Idle time-out(min):

Figure 7-22 Submitting an EoD job on PAC

After the hardware resource is properly allocated for the job, the status moves to the Running status as shown in Figure 7-23. The progress of these requests can be monitored by clicking the jobs tab on the left side of the window.

Jobs

plc service is not running. Job information is out of date. Start up the service with 'perfadmin start plc'.

Filter : ON

ID	Type	Name	State	Application	Submitted	Ended	User
2628	Job	test-rogarcia	Running	EXCEED_X...	2013-05-29...	-	Isfadmin
2622	Job	MYDCV	Running	DCVApp:Ap...	2013-05-23...	-	hari
2610	Job	test	Running	Eod_xterm...	2013-05-20...	-	luoming
2609	Job	MYJOB	Running	AppDCVon...	2013-05-20...	-	Isfadmin

Job: test-rogarcia (2628)

Job ID 2628 Job execution hosts c445f3an17.cluster.n... Submitted 2013-05-29 09:26:14

Job Name test-rogarcia

Figure 7-23 Job running

When the interactive session is allocated, a new status icon **Visualize** is displayed. When the user clicks this button, session information is made available in the form of a EoD file that can be started in the local notebook, provide that the user has installed the EoD client component.

The user is prompted to start a EoD session and provide credentials for authentication. Figure 7-24 shows the process to get the remote session started.

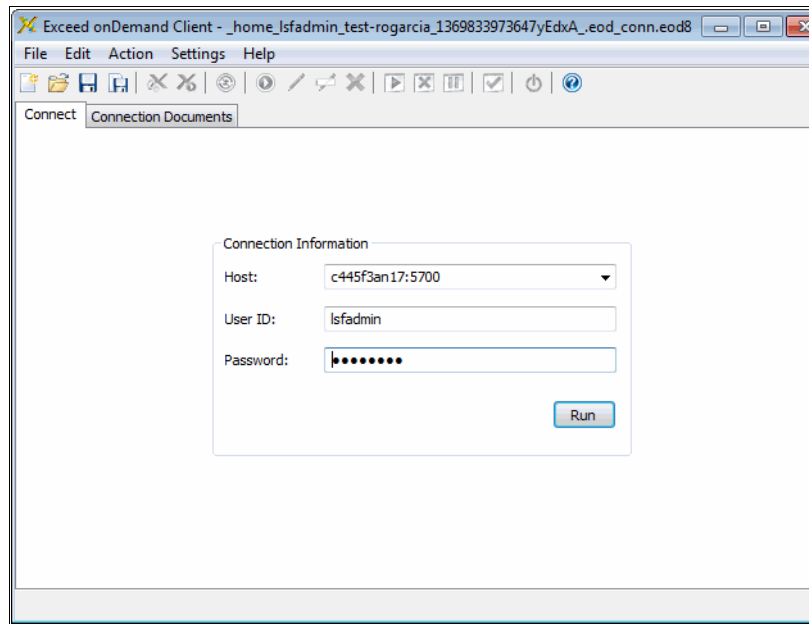


Figure 7-24 Starting the EoD connection

After the session is connected, the EoD client runs on the background and a new icon is displayed on the taskbar (in this case, of Windows) as shown in Figure 7-25.

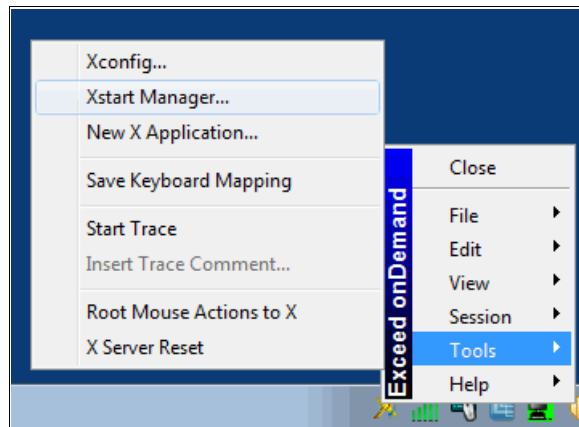


Figure 7-25 EoD menu on the taskbar

Click **Tools** → **Xstart Manager** to start a manager to select the appropriate Xstart file to run an ANSYS remote session. In the example environment, this is an ANSYS.xs file as shown in Figure 7-26.

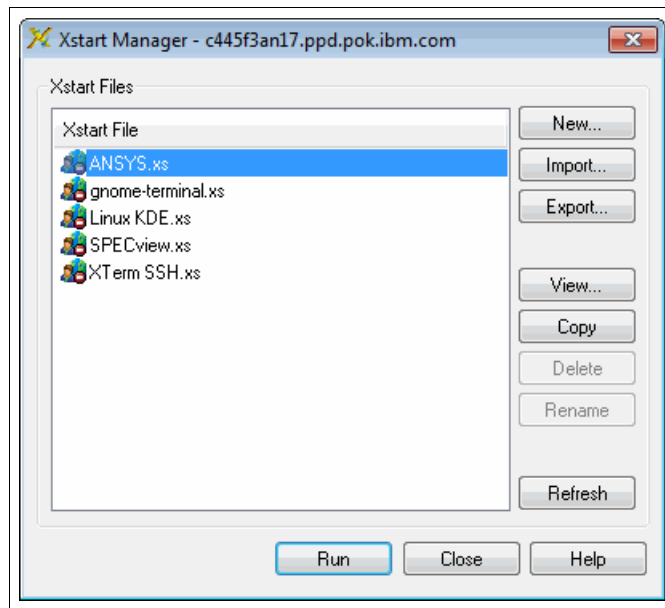


Figure 7-26 Starting by using EoD client

The Xstart file has a list of commands that can be passed as session parameters. Figure 7-27 shows the command used to start a session in the example environment.

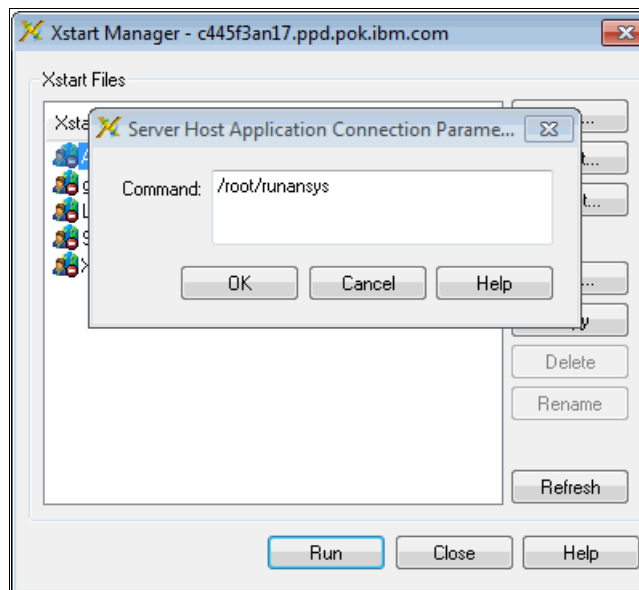


Figure 7-27 Starting script for ANSYS

After the user session is authenticated, EoD starts an ANSYS Workbench session such as the one illustrated in Figure 7-28. If the session disconnects, it can be reconnected by running the EoD provided by PAC.

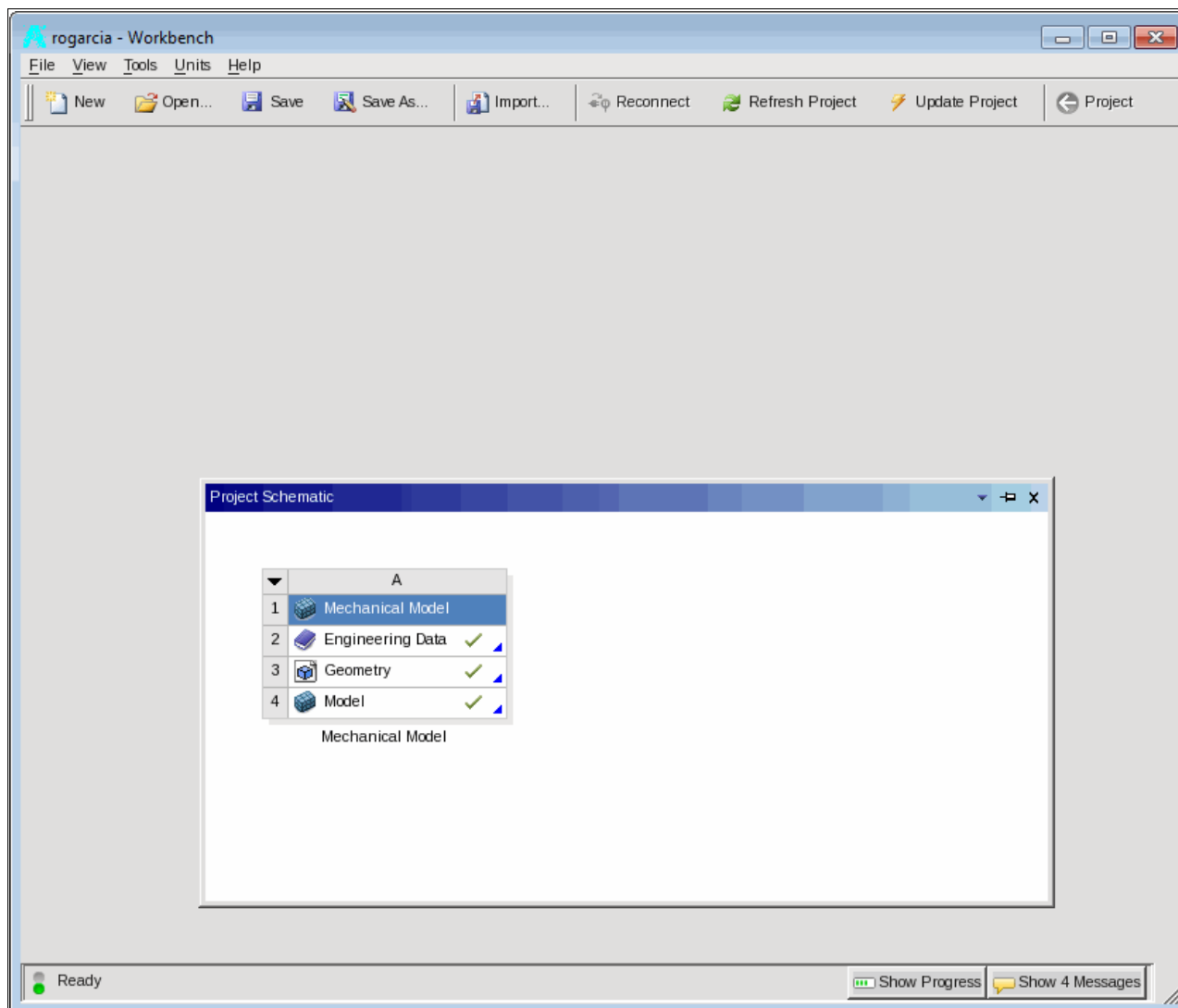


Figure 7-28 EoD virtualized session running ANSYS Workbench on the server side

Figure 7-29 shows the work on a mechanical model.

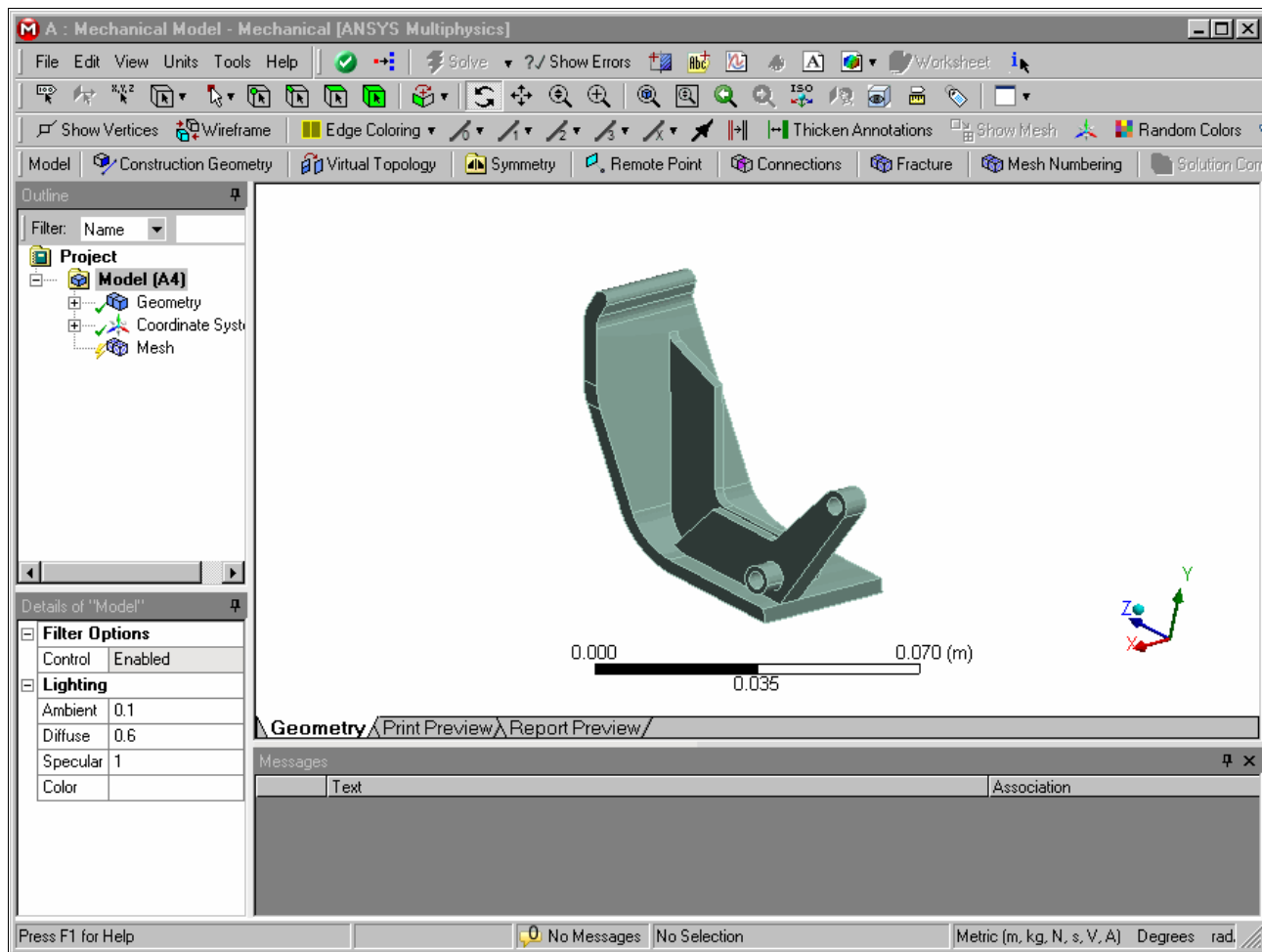


Figure 7-29 Working on a mechanical model

Remote collaboration

In a large globally dispersed enterprise, disperse the engineering 3D cloud deployment across the globe just like the engineering users. In other words, if you have multiple engineers in Tokyo, multiple engineers in Detroit, and multiple engineers in Berlin, consider building separate engineering 3D clouds in Tokyo, Detroit, and Berlin. These globally dispersed engineering clouds can (and should) be connected, and can even serve as a failover for the other cloud sites. In this case, the dispersed engineers get the best response and least network issues when using their local engineering cloud. Collaboration can still take place on a global basis, but distant collaborators might see some hesitation in model movement when in collaboration mode with an engineer in another location.

Scalability

Currently the upper limits of scalability of the Engineering 3D Cloud solution are unknown. The network characteristics of the client considering this solution are most likely the primary limiting factor for upward scalability. Because TCP/IP is used as the transport mechanism for the graphics images, network bandwidth and client network traffic must also be considered.



Solution for life sciences workloads

This chapter provides a brief introduction on the technical computing cloud solution for life sciences workloads. It provides an overview of the background for the solution, describes the reference architecture, and finishes with use case scenarios to explain how the solutions can help you solve your life science workload challenges.

This chapter includes the following sections:

- ▶ Overview
- ▶ Architecture
- ▶ Use cases

8.1 Overview

Dramatic advances occurring in the life sciences industry are changing the way that we live. These advances fuel rapid scientific discoveries in genomics, proteomics, and molecular biology that serve as the basis for medical breakthroughs, the advent of personalized medicine, and the development of new drugs and treatments.

Today, the typical life sciences company needs to access and analyze petabytes (10^{15} bytes) of data to further their research efforts. Dynamic market changes affecting life science organizations require a new approach to business strategy. The competitive advantage belongs to companies that can use IT resources efficiently and manage imminent growth.

Figure 8-1 describes today's vision where life sciences leaders see research and development as a key transformative area.

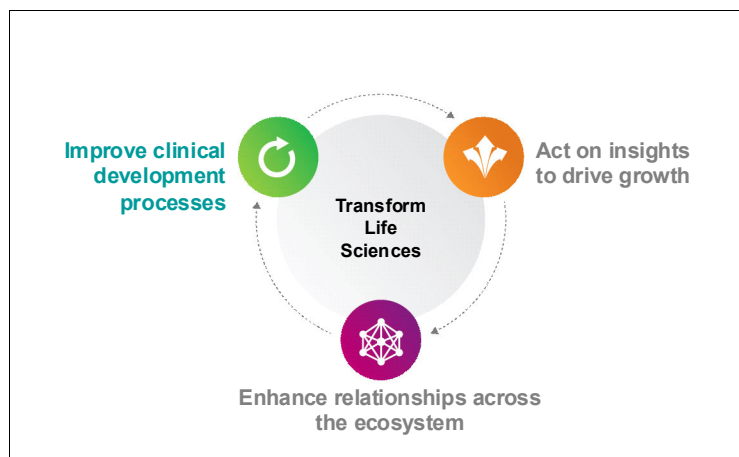


Figure 8-1 Redefining value and success in life sciences

8.1.1 Bioinformatics

This section introduces bioinformatics as one of the areas that is redefining life sciences, driving the need for technical cloud-computing infrastructure.

Next generation sequencing

Finding new ways to perform faster and more reliable sequence assembly and mapping of the human genome is an ongoing industry challenge. As the data from next generation sequencing (NGS) technologies continues to increase, deploying efficient software tools, and high-performance computing (HPC), and storage technology becomes essential to accelerate drug research and development.

NGS technologies have been instrumental in significantly accelerating biological research and discovery of genomes for humans, mice, snakes, plants, bacteria, virus, cancer cells, and so on. Researchers now process immense data sets, build analytical deoxyribonucleic acid (DNA) models for large genomes, use reference-based analytic methods, and further their understanding of genomic models. This is useful for drug discovery, personalized medicine, toxicology, forensics, agriculture, nanotechnology, and other emerging use cases.

NGS technologies parallelize the sequencing process, producing thousands or millions of sequences at a time. These technologies are intended to lower the cost of sequencing beyond what is possible with standard dye-terminator methods. High-throughput sequencing technologies generate millions of short reads from a library of nucleotide sequences. Whether

they come from DNA, RNA, or a mixture, the sequencing mechanism of each platform does not vary.

Translational medicine

Modern medicine focuses on data integration, using genomic data and the analytics that are required to identify biomarkers to understand disease mechanisms and identify new medical treatments.

This translational field provides a deeper understanding of genome and disease biology that is key for major advances in medicine.

Personalized health care

Advancements in translational medicine, accelerated by NGS technologies, enable health professionals to deliver evidence-based therapeutic intervention to improve the effectiveness of treatments and outcomes.

8.1.2 Workloads

Table 8-1 shows the life sciences workloads, purpose, workload characteristics, and applications that you can use to understand these workloads.

Table 8-1 Technical computing workloads in life sciences

Discipline	Purpose	Workload characteristics	Major applications
Bioinformatics - sequence analysis	Searching, alignment and pattern matching of biological sequences (DNA and protein)	Structured Data. Integer dominant, frequency-dependent, large caches, and memory BW not critical, some algorithms are suited to single instruction, multiple data (SIMD) acceleration	<ul style="list-style-type: none"> ▶ NCBI BLAST, wuBLAST ▶ ClustalW, HMMER ▶ FASTA, Smith-Waterman ▶ SAM tool, GATK
Bioinformatics - sequence assembly	Align and merge DNA fragments to reconstruct the original sequence	Usually have large memory footprint, for de novo assembly	<ul style="list-style-type: none"> ▶ Phrap/phred, CAP3/PCAP ▶ Velvet, ABySS, SOAPdenovo ▶ Newbler, MAQ, BOWTIE, ▶ BFAST, SOAP, BioScope ▶ GAP, pGAP (TAMU)
Biochemistry - drug discovery	Screening of large database libraries of potential drugs for ones with the wanted biological activity	Mostly floating point, compute intensive, highly parallel	<ul style="list-style-type: none"> ▶ Dock, Autodock, GLIDE ▶ FTDock, Ligandfit, Flexx
Computational chemistry - molecular modeling & quantum mechanics	Modeling of biological molecules using Molecular Dynamics and Quantum Mechanics techniques	Very floating point intensive, latency critical, frequency dependent, scalable to low 100s	<ul style="list-style-type: none"> ▶ CHARMM / CHARMM, ▶ GROMACS ▶ Desmond, AMBER, NAMD ▶ Gaussian, GAMESS, Jaguar, ▶ NWCHEM

Discipline	Purpose	Workload characteristics	Major applications
Proteomics	Interpreting mass spectrometry data and matching the spectra to protein database	Mostly Integer dominant, frequency dependent. Not communication intensive	<ul style="list-style-type: none"> ► Mascot, Sequest ► ProteinProspector ► X!Tandem, OMSSA

8.1.3 Trends and challenges

One of the life sciences industry's most difficult challenges is transforming massive quantities of highly complex, constantly changing data, from many data sources into knowledge. The challenge of harnessing this substantial data into life sciences insights by transforming information into knowledge is increased by the exponential increase in data that are created in every domain. Somewhere within the mountains of information are answers to questions that can prevent and cure disease. Questions such as what proteins are encoded by the over 30,000 human genes? What biological pathways do they participate in? Which proteins are appropriate targets for the development of new therapeutics? What molecules can be identified and optimized to act as therapeutics against these target proteins?

Managing very large-scale computing

Based on current data growth expectations, computing hundreds of petaflops will be a reality by 2018. What will house it? The *Power Utilization Efficiency* of data centers becomes as important as the "green solution" you put in it. And, how do you keep it fully utilized?

The data deluge

Big data and big data management are problems for researchers. There are very large worldwide projects where data is measured in the hundreds of petabytes. Analytic solutions must scale. Many Natural Language Processing (NLP) and statistical analyses packages cannot scale to the extent needed. The performance of the file system and the ability to transparently store data on the correct storage from SSD to tape are key to cost effective storage management.

Managing many HPC applications

The NGS pipeline revolves around moving data across several applications to reach a sequencing output that can be used as information for medicine. Managing these different data types as they flow through a complex pipeline of applications can be tackled by cloud management software. Doing so provides both high throughput computing and high performance (capability) computing using a shared environment. Costs are reduced when you build a central condominium facility where researchers can contribute.

Bioinformatics pipeline

Bioinformatics pipelines are *data and compute intensive*:

- 1 Human Genome = 300 GB ~ 700 GB (short, deep reads)
- Mapping, annotation = 1000+ compute hours
- 1000 genomes x 1000 compute hours = 114 years

The bioinformatics pipelines need these characteristics:

- Vast scalable storage
- Parallel compute design
- Old and new bioinformatics tools interoperability

- ▶ Fault tolerance
- ▶ Ability to share dynamic resource pools:
 - To run mixed workloads simultaneously
 - To meet changing LOB needs and SLAs

Traditional approach

The key to increasing R&D effectiveness and remaining competitive in today's fast-paced scientific community is data integration. The ability to tap into multiple heterogeneous data sources and quickly retrieve clear, consistent information is critical to uncovering correlations and insights that lead to the discovery of new drugs and agricultural products.

Traditional approaches to production bioinformatics pipelines such as data warehousing and point-to-point connections between specific applications and databases have strong limitations. Data warehousing (placing data into a centralized repository) works well in situations where information is relatively static and data types are not too diverse. However, building and maintaining enterprise-wide warehouses that contain hundreds of data sources can be costly and risky to implement.

Similarly, the technical effort and costs that are associated with writing customized point-to-point connections to multiple data sources and applications can result in time-consuming processes for companies with limited IT resources.

Lack of comprehensive pipeline management software often results in dedicated clusters for each specific workload type.

Typical file system solutions are inadequate:

- ▶ Expensive, not easily scalable, slow, unreliable
- ▶ Limited archive/backup
- ▶ Poor performance
- ▶ Many file systems impede research

Recent big data experiments have been performed, but they are restricted to “closet clusters” running Hadoop on limited infrastructure, or using expensive and unreliable public cloud resources. This generates poorly optimized cluster silos and siloed applications, decreasing the overall efficiency and effectiveness of your R&D operations.

Add to these issues the need to work within existing laboratory and business computing environments, and the challenges facing today's life sciences industry are almost overwhelming.

8.1.4 New possibilities

In response to these challenges, life sciences companies are redefining their research methodologies and retooling their IT infrastructures. The traditional trial-and-error approach is rapidly giving way to a more predictive science based on sophisticated laboratory automation and computer simulation. The technology that is used in the new life sciences discovery models is critical to laboratory productivity and time to market. The following are transformations that technical computing solutions can enable in the life sciences industry:

- ▶ More efficiently use compute resources while gaining faster time to results.
- ▶ Integrate new technologies into heterogeneous research environments with fewer failures.

- ▶ Boost performance of complex workloads such as de novo assembly and improve clinical collaboration.
- ▶ Accelerate execution of resource-intensive applications that demand big data management and analytics capabilities within resource and cost constraints.
- ▶ More quickly enable secure, integrated cloud environments.
- ▶ Sharing and pooling information across global resources while maintaining security.
- ▶ Retrieving and integrating diverse data across many scientific domains.
- ▶ Adding new data sources without new software development or complete redeployment of the solution.
- ▶ Acquiring experimental data from industrial-style laboratory activities 24 hours a day, 7 days a week.
- ▶ Enabling continuous real-time access to data without building and managing database warehouses.
- ▶ Developing new ways to collaborate among research teams by using shared research to focus efforts.

8.2 Architecture

IBM technical computing end-to-end infrastructure solutions use the cloud benefits to provide the scalable tools and systems to help life sciences companies access, manage, and develop content. The life sciences industry needs flexible, scalable, reliable systems that can easily adapt as needs change.

IBM solutions for knowledge management, data integration, high-performance computing, and storage deliver the powerful capabilities that are needed in life sciences laboratories:

- ▶ Knowledge management tools for transforming life sciences data into knowledge.
- ▶ Data integration for extracting information and identifying patterns from multiple data sources and across diverse data domains.
- ▶ High-performance computing for computational modeling, simulation, and visualization.
- ▶ Industry-leading, supercomputing performance for scientific workloads, including genome sequencing, protein structure sequencing, and drug target identification.
- ▶ Storage and retrieval technologies and tools for managing data easily.
- ▶ Security and data management to help protect the privacy of research data.

An important, fast growing business within the life sciences industry is focused on the compilation of genomic information into databases, and the sale of information through subscriptions to drug companies and biotech research institutions. To help identify and analyze patterns within genetic data for viability as diagnostics and pharmaceutical products, drug discovery companies need powerful, high performance solutions.

High speed, high performance computing power and industrial strength databases perform a wide range of data intensive computing functions. These include mapping genetic and proteomic information, data mining to identify patterns and similarities, and text mining using large libraries of information. All of these activities require high-speed computer infrastructures with integrated storage systems.

Figure 8-2 illustrates the reference architecture for genomic medicine and translational research. This architecture uses IBM hardware and software solutions as well as open source and third-party application suites to provide an end-to-end solution for the life sciences industry.

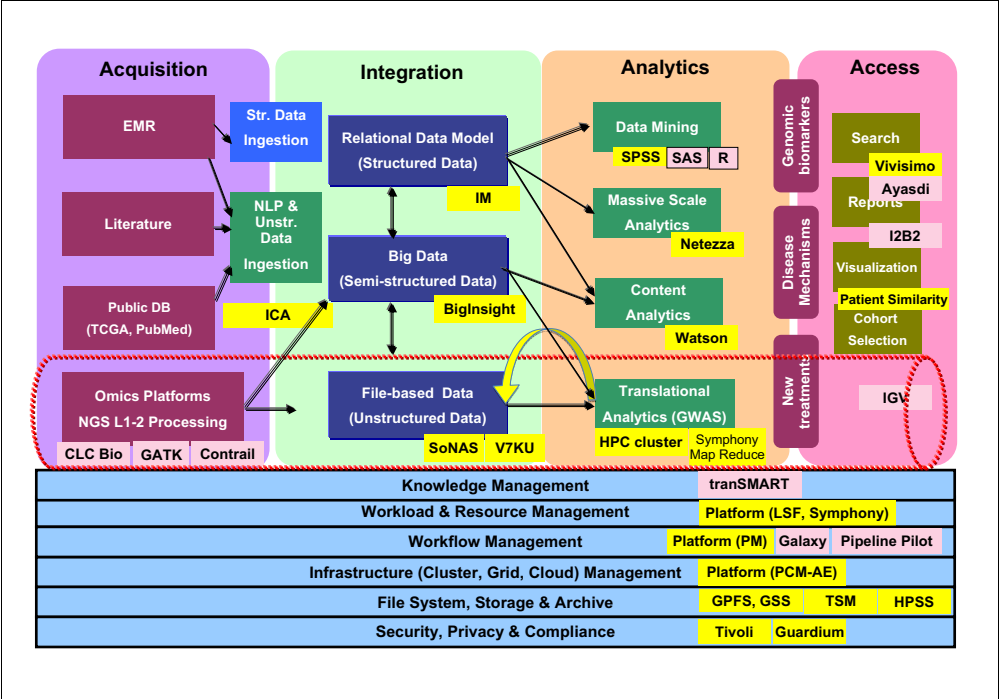


Figure 8-2 Reference architecture for genomic medicine and translational research

8.2.1 Shared service models

Today’s life sciences businesses require solutions with the flexibility to adapt and extend mission-critical applications to meet customer demands and the stability to smoothly absorb these changes.

The IBM data integration strategy provides hardware, software, and services to enable successful research and development in life sciences laboratories. A shared services model refers to an infrastructure management platform that enables mixed workloads running on a shared grid and sophisticated SLAs among multiple business units.

By combining IBM Platform Computing solutions with IBM high performance computing systems and software, organizations can accelerate application performance, improve infrastructure, and reduce time to results. Figure 8-3 shows a high-level architecture of a genomic sequencing pipeline, where multiple MapReduce jobs from the mapping, alignment, and variation detection steps, use the low latency provided by IBM Platform Symphony SOA services. This configuration optimizes resource utilization and shares reference genome data, resulting in improved performance when compared to siloed models.

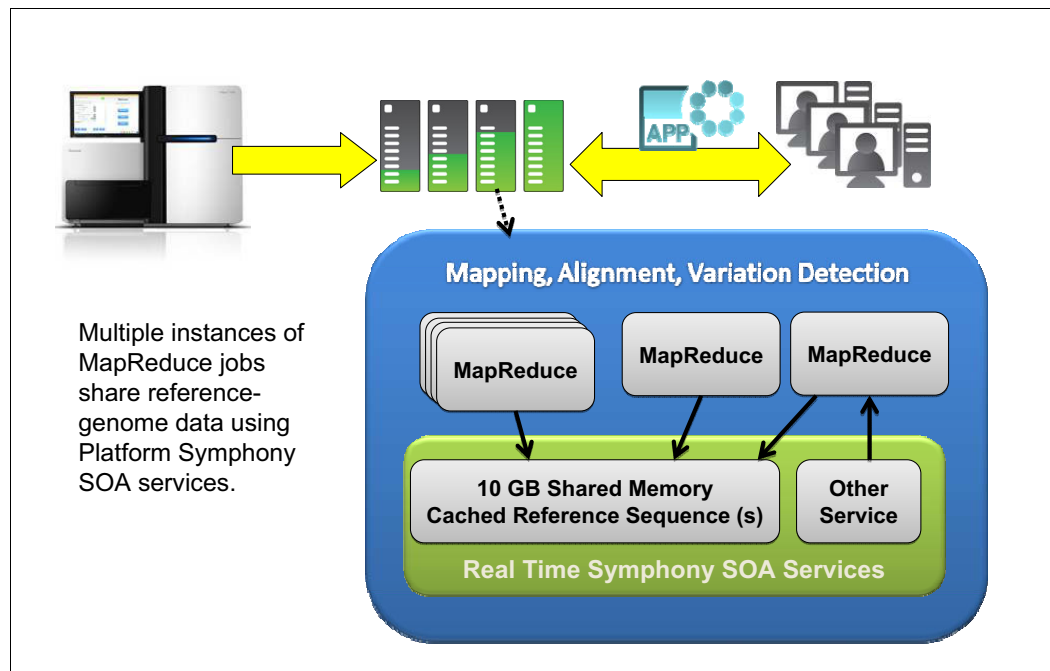


Figure 8-3 Platform Symphony and real-time SOA solution for ultra fast shared memory

IBM has combined technology, industry expertise, best practices, and leading analytical partner applications into a tightly integrated solution. With this solution, research institutions and pharmaceutical companies can easily manage, query, analyze, and better understand integrated genotypic and phenotypic data for medical research and patient treatment. They can perform these tasks:

- Organize, integrate, and manage different kinds of data to enable focused clinical research, including diagnostic, clinical, demographic, genomic, phenotypic, imaging, environmental, and more.
- Enable secure, cross-department collection and sharing of clinical and research data.
- Ensure flexibility and growth with open and industry-standards based architecture.

Figure 8-4 illustrates an example of a full solution architecture for genome sequencing.

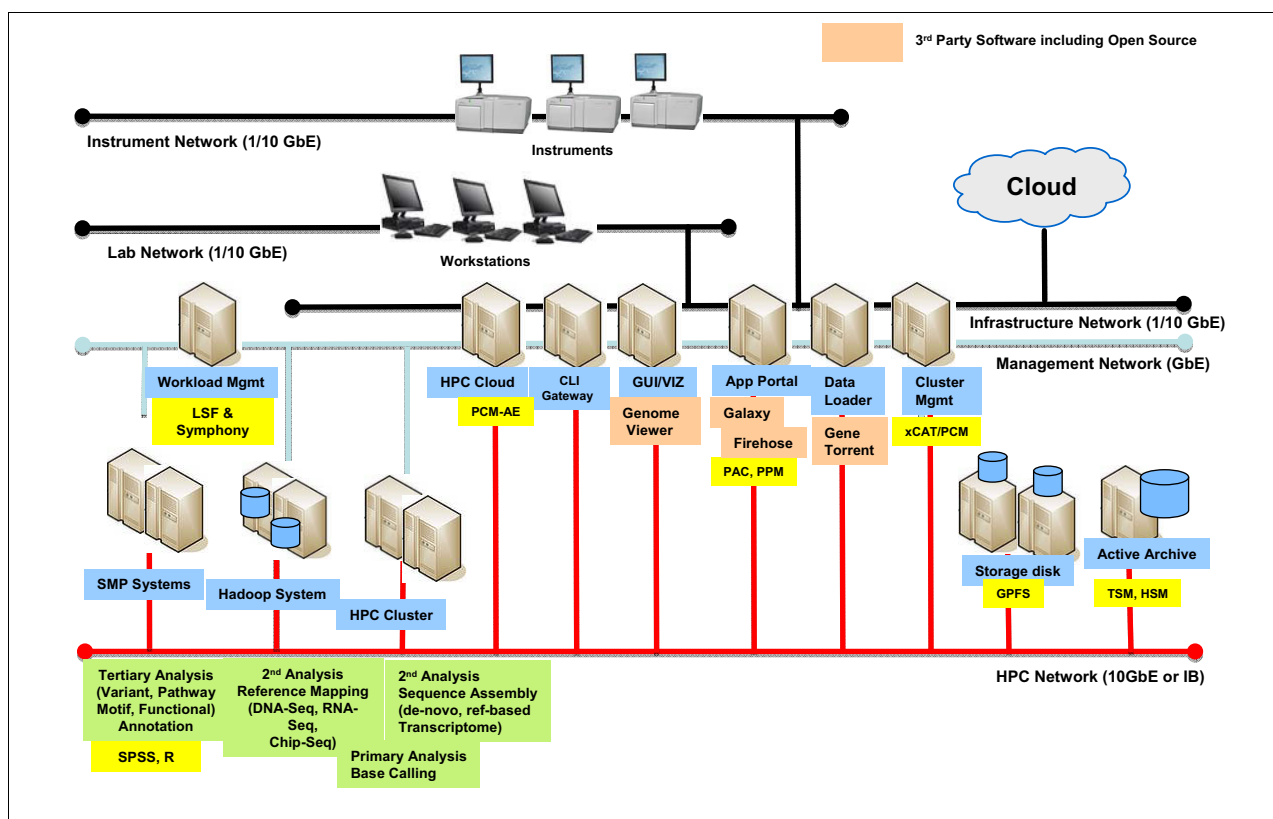


Figure 8-4 Example of system architecture

8.2.2 Components

This section describes the architecture components of the solution.

Hardware

IBM delivers complete solutions for searching vast quantities of genomic data from many sources and running thousands of jobs simultaneously. A wide range of server solutions enable drug discovery companies and biotechnical researchers to improve the value of the data while maintaining control over the analysis phase.

Figure 8-5 illustrates the different types of hardware components that are needed to deliver a complete NGS solution. It covers the full cycle of data acquisition, processing, storage, and analytics. The amount of data that is generated by sequencers is growing so fast that *information lifecycle management* (ILM) has become an essential part of this pipeline. Therefore, hierarchical storage management, using tape and General Parallel File System (GPFS) ILM are key to building a high performance genomics solution.

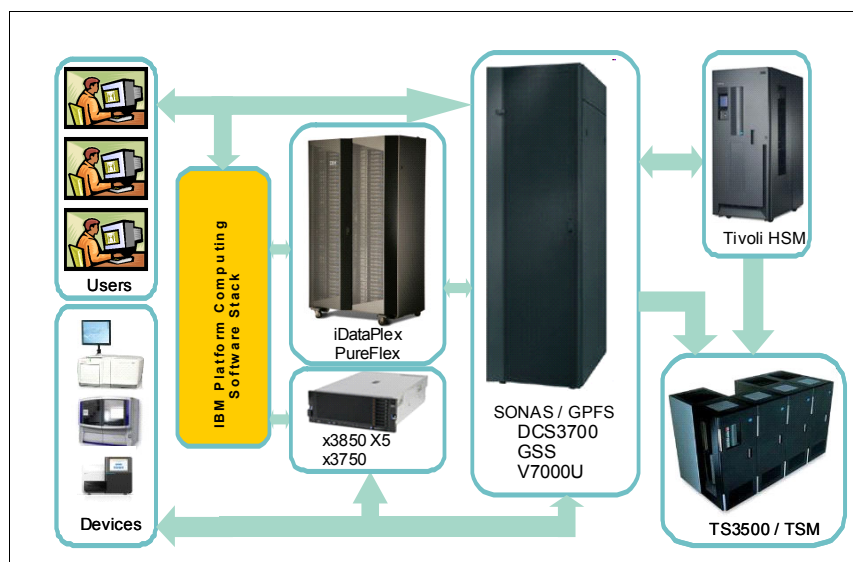


Figure 8-5 IBM Next Generation Sequencing solution system components

The characteristics of the hardware that is involved depend on application and type of workload that is being run. Figure 8-6 shows a high-level summary of the preferred hardware based on workload type.

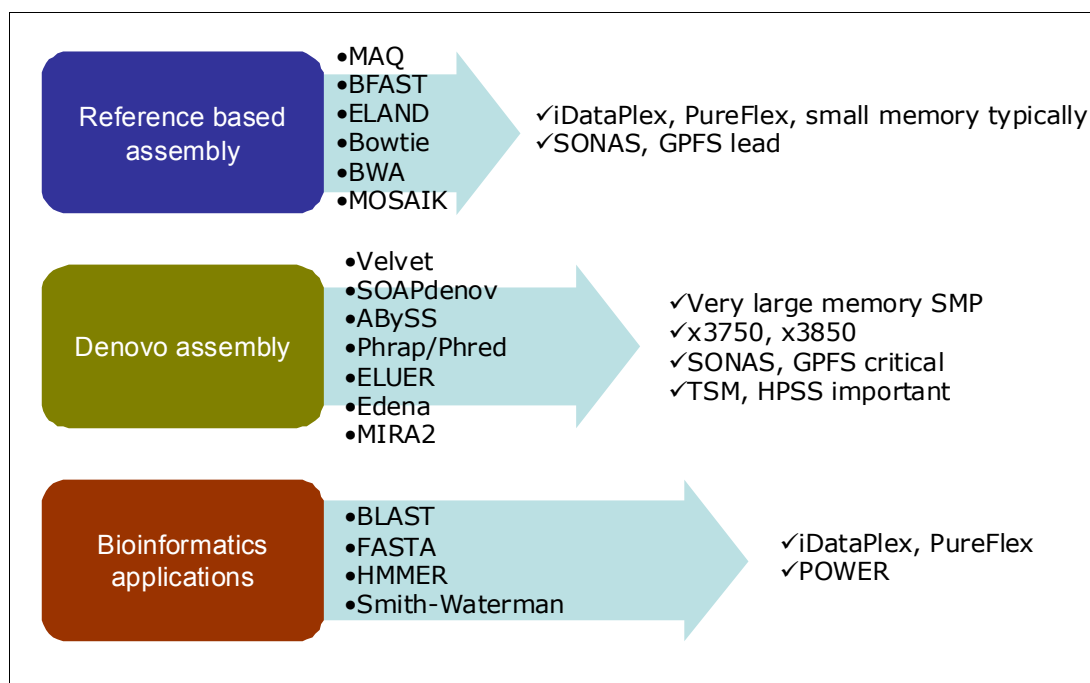


Figure 8-6 Preferred hardware based on application characteristics

Data management

Many drug discovery processes, including clinical trials, require maximum efficiency for data sharing and knowledge management functions such as patient record mining across companies. IBM has the end-to-end open source infrastructure solutions to help optimize data sharing and information management.

A key component of a high-quality life sciences solution is reliable, disaster-proof storage. IBM storage hardware, software, and services can help maximize laboratory productivity and minimize operating costs. Researchers can store results obtained from collaborative research and data mining in “pools” of commonly shared knowledge that are administered from a centralized point. Laboratories can increase capacity without interruptions by using these scalable storage systems, and reduce backup time because only modified data must be transferred.

IBM Storwize V7000 Unified

Many users have deployed storage area network (SAN) attached storage for their applications that require the highest levels of performance, while separately deploying network-attached storage (NAS) for its ease of use and lower-cost networking. This divided approach adds complexity by introducing multiple management points, and also creates islands of storage that reduce efficiency.

The Storwize V7000 Unified system allows you to combine both block and file storage into a single system. By consolidating storage systems, multiple management points can be eliminated and storage capacity can be shared across both types of access. This configuration helps improve overall storage utilization. The Storwize V7000 Unified system also presents a single, easy-to-use management interface that supports both block and file storage, helping to simplify administration further.

Scale Out Network Attached Storage

The IBM Scale Out Network Attached Storage Gateway system is designed to manage vast repositories of information in enterprise environments that require large capacities, high levels of performance, and high availability.

Scale Out Network Attached Storage Gateway uses a mature technology from the IBM HPC experience. It is based on the IBM General Parallel File System (GPFS), a highly scalable clustered file system. Scale Out Network Attached Storage Gateway is an easy-to-install, turnkey, modular, scale out NAS solution. It provides the performance, clustered scalability, high availability, and functionality that are essential for meeting strategic multi-petabyte and cloud storage requirements.

Note: The difference between the IBM Storwize V7000 Unified and Scale Out Network Attached Storage Gateway systems lies in the workloads that each system can support. The Storwize V7000 Unified system can support smaller and medium-size workloads. Scale Out Network Attached Storage Gateway system can deliver high performance for extremely large application workloads and capacities, typically for the entire enterprise.

8.3 Use cases

There are extraordinary challenges and opportunities ahead for the life sciences industry. The scientific challenges in this emerging industry are matched by the challenges associated with managing data integration and developing the computing technology and tools needed to provide solutions for the laboratory.

8.3.1 Mixed workloads on hybrid clouds

Figure 8-7 illustrates the basic components of a life sciences hybrid cloud model. It consists of a private cloud to handle both batch-oriented workflows and near real time, service oriented workloads. Using a reliable middleware that support both MapReduce and non-MapReduce applications, the infrastructure of this private cloud manages a wide array of workloads. It can keep the utilization ratio of local resources at its maximum.

Using self-service portals to deliver easy access to researchers, the private cloud accelerates time to results. The cloud either provides infrastructure as a service (IaaS) or platform as a service (PaaS) for big data genomics applications.

The private cloud is able to expand and connect to public cloud resources to provide easy access to results across institutions. Another key aspect of the public cloud integration is the ability to provide additional computational power to cope with peak demands or experimental projects, without increasing capital expenses.

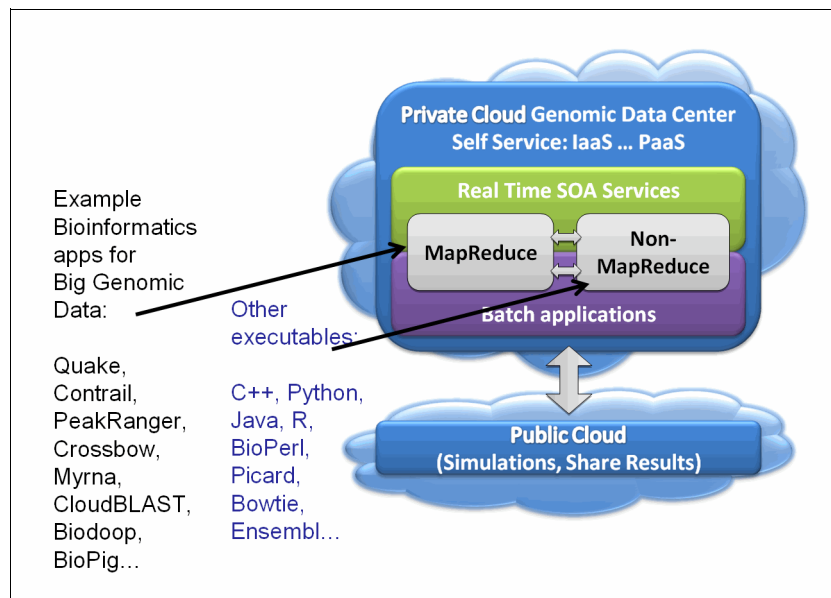


Figure 8-7 Hybrid cloud reference architecture model

Figure 8-8 on page 185 details the use case of a genomics pipeline using IBM and open source software components to deliver a high performance end-to-end solution. Moving from left to right there are five phases in the pipeline:

- ▶ Sequence
- ▶ Queue
- ▶ Assemble/map
- ▶ Annotate
- ▶ Store

In each phase, you make use of the underlying infrastructure to process the necessary data and provide input for the next phase. Several applications are involved in the process to take data from its initial raw stage (sequencer output), and transform it into relevant genomic information. The data is usually stored in a high performance shared file system layer, allowing applications to process workflows concurrently. Extra abstraction layers provide schema and scripting capabilities to work on top of the stored data, using MapReduce to extract information faster and in a scalable manner.

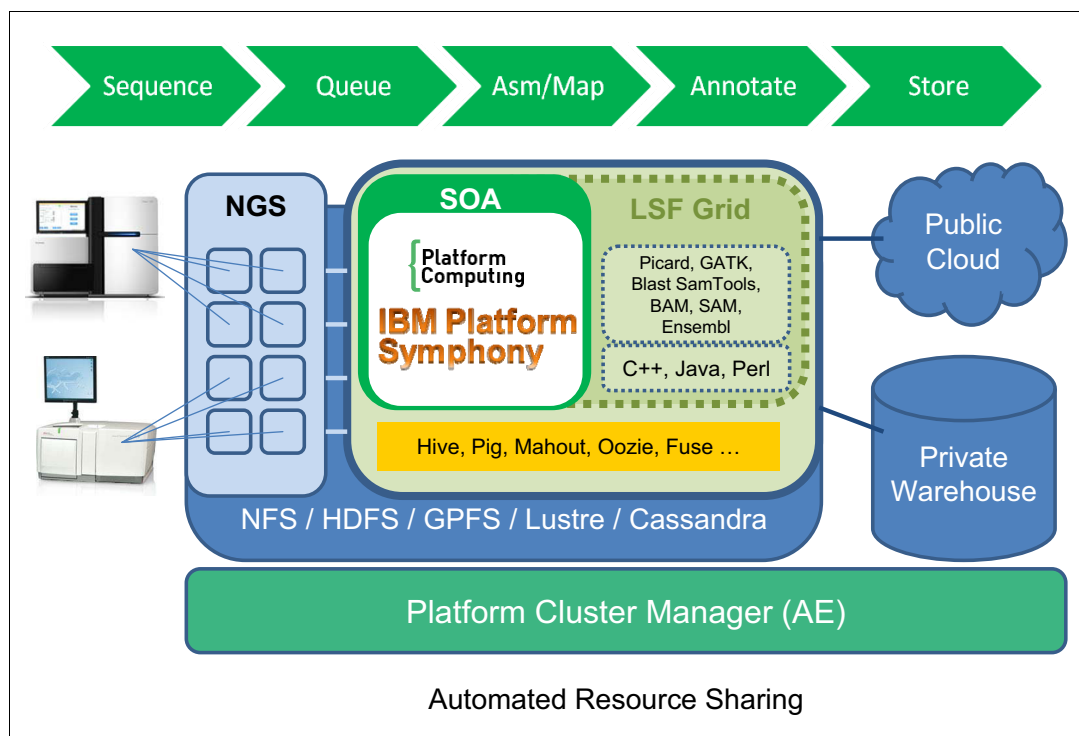


Figure 8-8 Mixed workloads pipeline

8.3.2 Integration for life sciences private clouds

The focus of the work done for this book was on the integration of life sciences application into a private cloud. The example evaluates an infrastructure setup that was able to provide very good results for sequence, assemble, map, data merge, and variant calling phases in a genomics pipeline (Figure 8-9).

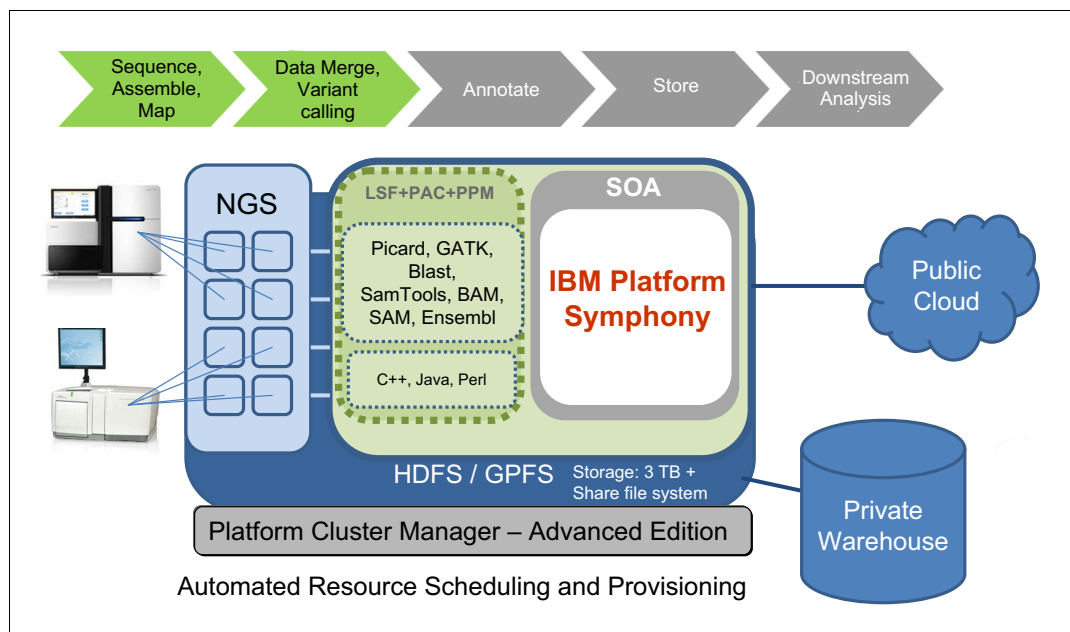


Figure 8-9 Scope of the evaluated solution

Open source tools

This section describes the open source tools available to help life sciences workloads.

BWA (Burrows-Wheeler Alignment Tool)

A software package for mapping low-divergent sequences against a large reference genome, such as the human genome.

BWA is an open source, high-performance tool, and is available freely, with no software licensing restrictions. It is an efficient program that aligns relatively short nucleotide sequences against a long reference sequence such as the human genome. It implements two algorithms, BWA-SHORT and BWA-SW. The former works for query sequences shorter than 200 base-pairs, and the latter for longer sequences up to around 100,000 base-pairs. Both algorithms do gapped alignment. They are usually more accurate and faster on queries with low error rates.

SAMTOOLS (Sequence Alignment/Map Tool)

Provides various utilities for manipulating alignments in the SAM format, including sorting, merging, indexing, and generating alignments in a per-position format.

PICARD

Consists of Java-based command-line utilities that manipulate SAM files, and a Java API (SAM-JDK) for creating new programs that read and write SAM files. Both SAM text format and SAM binary (BAM) format are supported.

GATK-lite (Genome Analysis Toolkit open source version)

Software package that was developed at the Broad Institute to analyze next-generation resequencing data with a primary focus on variant discovery and genotyping with a strong emphasis on data quality assurance.

Workflow

The genomic sequencing pipeline can be efficiently implemented by mapping a series of interdependent tasks into workflows. However, these workflows tend to become complex and, without automation, difficult to maintain. Table 8-2 shows the required transformation of data to reach the format wanted in a variant call workflow. The sequence must be carefully observed because specific input and output formats are required by the open source tools that are employed in the process.

Table 8-2 Input and output flow

FASTA (FastAlignment format)	FASTQ (biological sequence and its quality data)
SAI (Alignment index file)	
SAM (Standard Alignment/ Map format)	
BAM (Binary Alignment / Map format)	
VCF (Variant call format)	
Final output = VCF format	

IBM Platform Process Manager (PPM)

To manage the complex workflows that are required by the genomic sequencing pipeline, use IBM Platform Process Manager (PPM) to create and manage a variant detection workflow for a genomic sequencing experiment.

PPM is a workflow management tool for users to automate their business processes in UNIX and Windows environments by creating and managing flow definitions. A flow definition is a collection of jobs, job arrays, subflows, and their relationships that represents work items and their dependencies.

The basic requirement to generate a workflow is to develop the specific commands for the open source tools to generate a vcf file. Because various tools are required to handle all the intermediate formats, PPM is extremely helpful when managing job dependencies.

The flow editor client component of the Platform Process Manager is a tool that can be used to create a complete workflow that can be deployed to a Platform Load Sharing Facility (LSF) cluster. With the flow editor, you can create jobs and their relationships, and define dependencies based on files or time. The example environment uses a complete Variant Call Format (VCF) file creation workflow for demonstration purposes.

Note: Platform Process Manager workflow creation is not described in detail in this publication. For more information, see *IBM Platform Computing Solutions*, SG24-8073, and *IBM Platform Computing Integration Solutions*, SG24-8081.

Figure 8-10 shows the flow manager being used to visualize the workflow created for the variant calling demonstration. The right pane shows a visual representation of this flow definition.

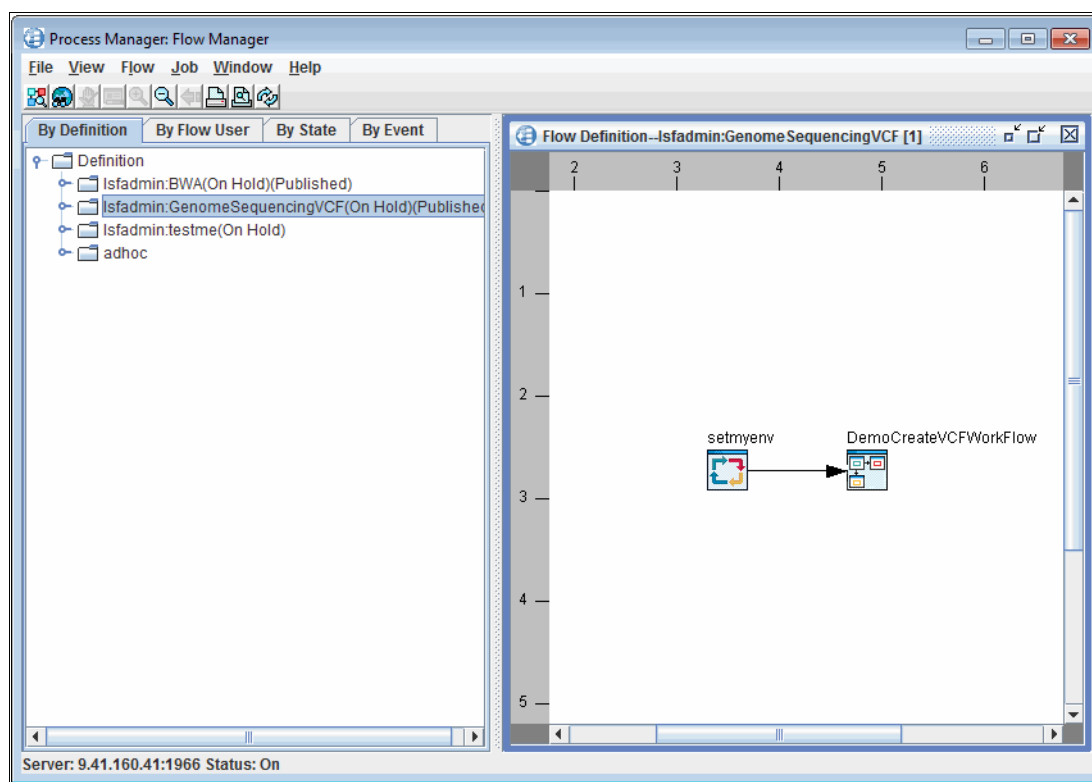


Figure 8-10 Using the flow manager to access available flows on the Process Manager server

The *DemoCreateVCFWorkflow* block shown in Figure 8-10 on page 187 is actually a subflow that can be expanded as shown in Figure 8-11. The right pane describes the complete flow of interdependent job arrays that are required to reach the final vcf format. BWA, Picard, and GATK jobs must be run in a certain order to achieve the result wanted.

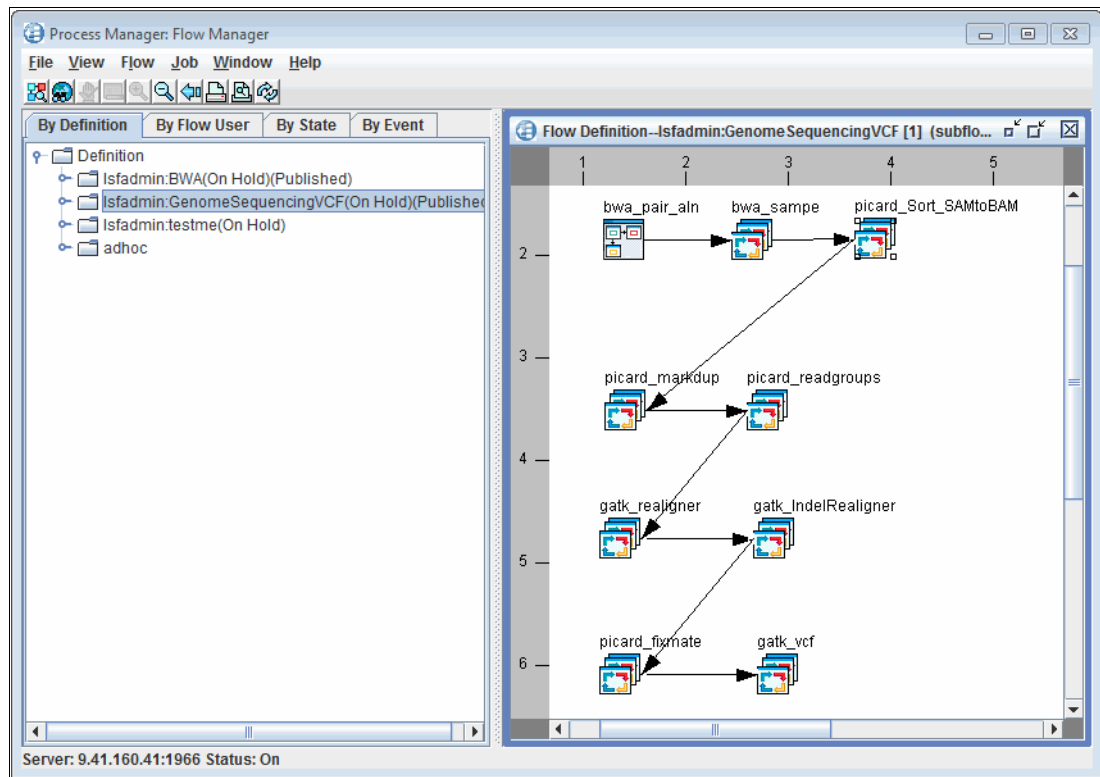


Figure 8-11 Expanding a subflow inside flow manager

PPM provides fine grained control of the relationships between blocks within a flow. Users can create job flow definitions in the Process Manager Client, and then submit them to the Process Manager Server. The Process Manager Server manages job dependencies within the flow and controls submission to the IBM Platform LSF master host. The IBM Platform LSF master host provides resource management and load balancing, runs the job, and returns job status to the Process Manager Server.

Figure 8-12 shows the detailed definition for *gatk_vcf*, which is the last step in the creation of the vcf file.

Figure 8-12 Job definition details about a job array

Platform LSF and Platform Application Center (PAC)

Workload managers and resource orchestrators help manage and accelerate workload processing and help ensure completion across a distributed, shared, IT environment. They also help fully utilize all HPC resources, regardless of operating system, vendor, or architecture. By improving utilization, resources are more readily available, helping researchers to get more work done in a shorter amount of time. This can free up time for collaboration across the clinical development value chain for better insights and superior results. IBM Platform LSF fits this role perfectly. For more information, see Chapter 2, “IBM Platform Load Sharing Facilities for technical cloud computing” on page 13.

Scheduling policies

IBM Platform LSF includes flexible scheduling capabilities to ensure that resources are allocated to users, groups, and jobs in a fashion consistent with service level agreements (SLAs). With extended SLA-based scheduling policies, these software tools simplify administration and ensure optimal alignment of business SLAs with available infrastructure resources.

Fair share scheduling features allow you to fine-tune the algorithms that determine user priority and enable different fair share policies by project, team, or department. Job preemption controls help maximize productivity and utilization by avoiding preempting jobs that are almost complete. This system enables researchers to run significantly more analyses and tackle more complex computations in less time.

Features such as bulk job submissions, dynamically adjustable swap space estimates, flexible data handling and smarter handling of dependencies in job arrays allow users to spend less time waiting for cluster resources. This gives them more time focused on their

research. This ultimately contributes to more streamlined development processes in life sciences and can speed patenting, discovery, and time-to-market for new drugs.

Application templates

IBM PAC provides a complete self-service portal for users to start genomic sequencing workflows without dealing with complex submission scripts. The application templates can be customized by using the PAC visual interface to produce rich job submission forms, simplifying researchers daily tasks and experiments. The forms promote a high level of asset reuse, which makes parameter variation jobs easier to submit and automate. Figure 8-13 shows a simple submission form that requires only a few input parameters to start a BWA job as part of a genome sequencing workflow.

The screenshot displays the IBM Platform Application Center 9.1 web interface. On the left, a navigation pane shows a tree structure under 'Jobs' and 'Resources'. The 'Jobs' section is expanded, showing 'Submission Forms' and 'Flow Forms'. Under 'Flow Forms', 'GenomeSequencingVCF' is selected and highlighted in green. The main content area is titled 'Submit a job: GenomeSequencingVCF'. It features a top bar with 'Submit', 'Save As', and 'Delete' buttons. Below this is a 'Flow Parameters' section with several input fields: 'Job Flow Description' (empty), 'Reference Directory *' (with a 'Browse...' button), 'FASTQ Directory *' (with a 'Browse...' button), 'BWA number of threads [-t] option' (set to '4' with 'Max Threads=4' next to it), and 'User specified work directory' (with a 'Browse...' button). At the bottom of the form are 'Submit' and 'Revert' buttons. The top right of the interface shows the user 'demouser (User)', a 'Log Out | Help' link, and a timestamp 'May 28, 2013 16:10:55 EDT'.

Figure 8-13 Submitting a job using an application template to start a genome sequencing workflow

Integrating PPM and PAC

This section describes the integration between PPM and PAC to run life sciences workflows.

Figure 8-14 shows the flow management capabilities that are incorporated into PAC after it is integrated with the PPM server. The workflows can be managed from PAC in the same manner as in the flow manager client. While running flows, the user can pinpoint the exact execution step on workflow graph inside PAC, and link to any corresponding LSF jobs. That helps debugging eventual problems in large, complex workflows.

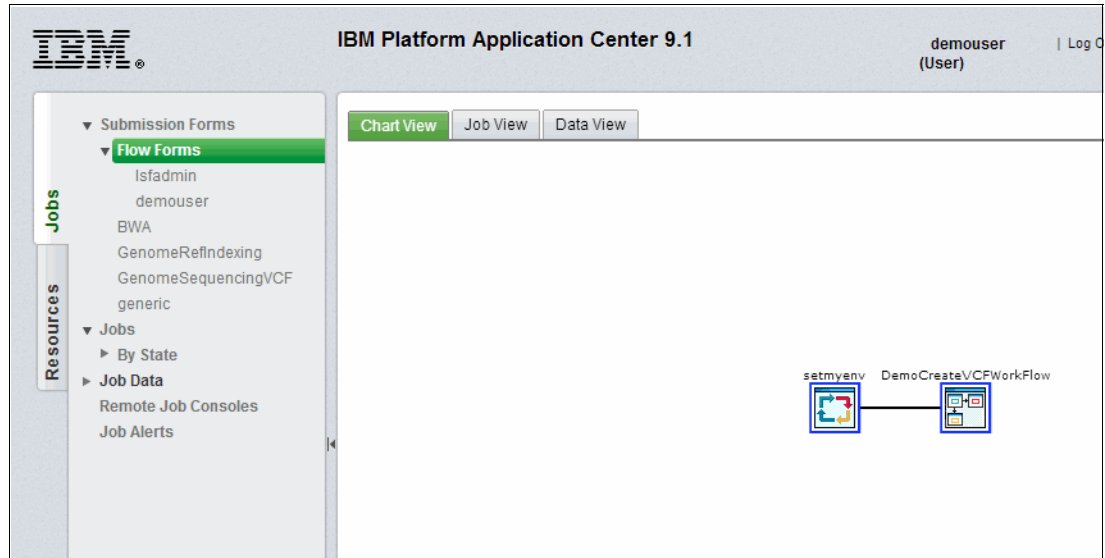


Figure 8-14 Visual representation of a workflow inside PAC

When submitting a flow using the PAC interface, there are extra tools to track the status. Figure 8-15 shows the available tabs in the parent job.

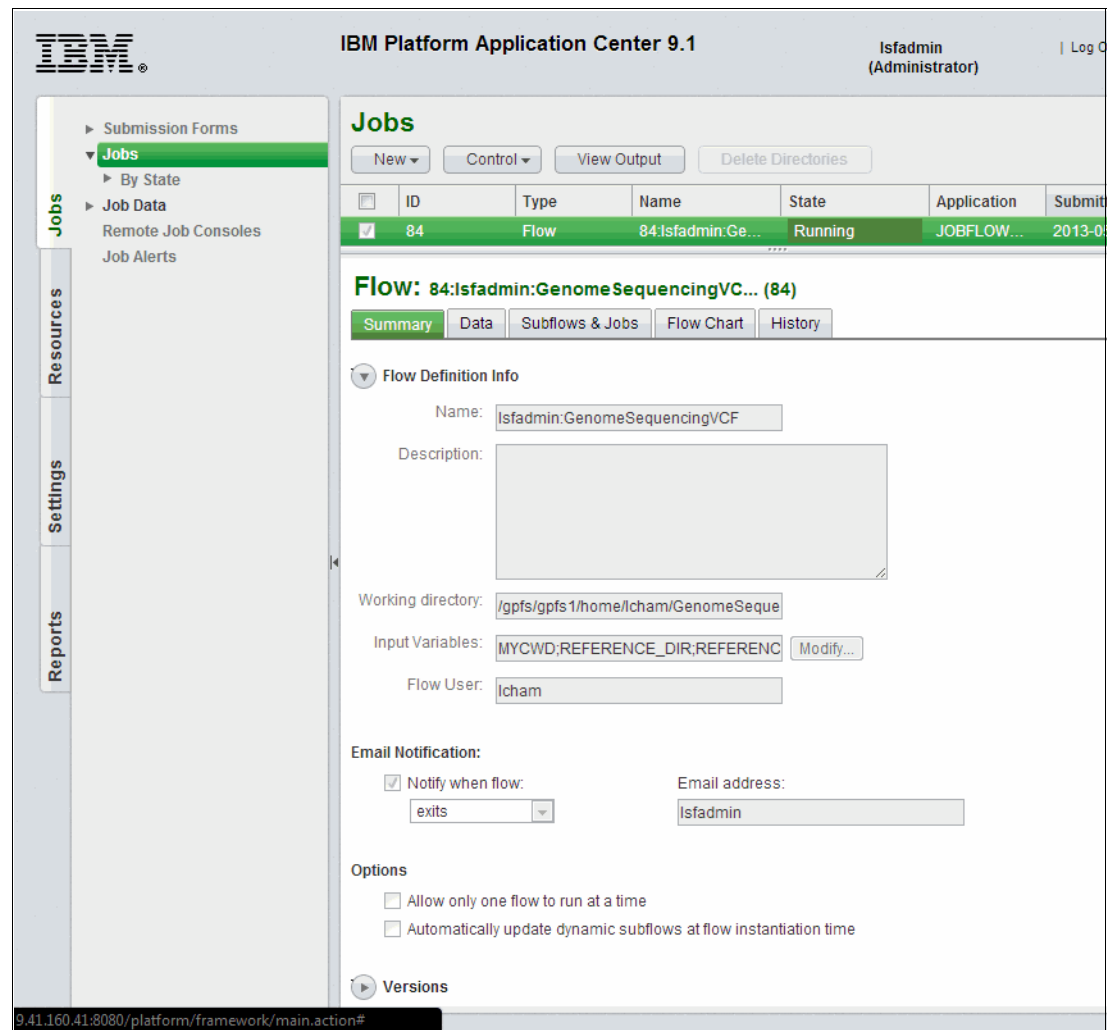


Figure 8-15 Flow running on LSF

Test environment

The test environment has seven compute nodes in a storage rich configuration that is connected over 10 Gb Ethernet. The shared file system used is IBM GPFS, which is also connected to the compute nodes through 10 Gb Ethernet links.

Hardware configuration

Each compute node has the following configuration:

- ▶ IBM System x iDataPlex dx360 M3 server
- ▶ Mellanox ConnectX-2 EN Dual-port SFP+ 10 GbE PCIe 2.0 adapter
- ▶ 1GE on-board adapter (management)
- ▶ 128 G RAM: 16x 8 GB(1x8 GB, 2Rx4, 1.5 V) PC3-10600 CL9 ECC DDR3 1333 MHz LP RDIMM
- ▶ 2x Intel Xeon Processor X5670 6C 2.93 GHz 12 MB Cache 1333 MHz 95w
- ▶ 12x IBM 3 TB 7.2 K 6 Gbps NL SAS 3.5" HS HDD

Software configuration

The following are the software components:

- ▶ Platform Application Center 9.1
- ▶ Platform Process Manager 9.1
- ▶ Platform LSF 9.1
- ▶ General Parallel File System 3.5

Life sciences software components:

- ▶ BWA 6.2
- ▶ Picard 1.79
- ▶ SAMTools 1.18
- ▶ GATK-lite 2.3.9

8.3.3 Genome sequencing workflow with Galaxy

This section provides a genome sequence workflow using Galaxy.

Deploying a Galaxy cluster

Figure 8-16 shows the IBM Platform Cluster Manager Advanced Edition window used to create a cluster definition for an LSF Galaxy cluster.

The screenshot displays the IBM Platform Cluster Manager Advanced Edition web interface. On the left is a navigation sidebar with sections: Resources (Cockpit, Definitions, Policies, Alarms), Clusters, Accounts, and System. The main content area is titled 'Cluster Definitions' and includes buttons for New, Copy, Modify, Delete, and Manage. A table lists several cluster definitions, with 'LSF Galaxy cluster' highlighted. Below this, the 'Cluster Definition: LSF Galaxy cluster' details are shown, including a Summary tab and a Properties section with fields for Name, Description, and Status. A Tiers section contains a table with two tiers: LSFMaster and LSFCompute.

Name	Account Name	Creator	Status	Description
Streams Cluster	AdminAccount	Admin	Unpublished	A basic Streams cluster
Galaxy VM	AdminAccount	Admin	Unpublished	Install Galaxy in a Web-DB Tier configuration
Test Streams Cluster	AdminAccount	Admin	Unpublished	A basic Streams cluster
LSF Galaxy cluster	AdminAccount	Admin	Unpublished	An LSF cluster with 1 master. See /opt/platform/icm/conf/cloudApplications/samples/lsf/README.txt

Tier Name	Operating System Template	Number of Machines	Number of CPUs
LSFMaster	rhels6.2-x86_64-install-PMTTools	1 - 2	1 - 32
LSFCompute	rhels6.2-x86_64-install-PMTTools	1 - Unlimited	1 - 32

Figure 8-16 Creating a cluster definition for an LSF Galaxy cluster

Figure 8-17 shows the cluster designer menu to change the LSF cluster definition to provision the Galaxy cluster.

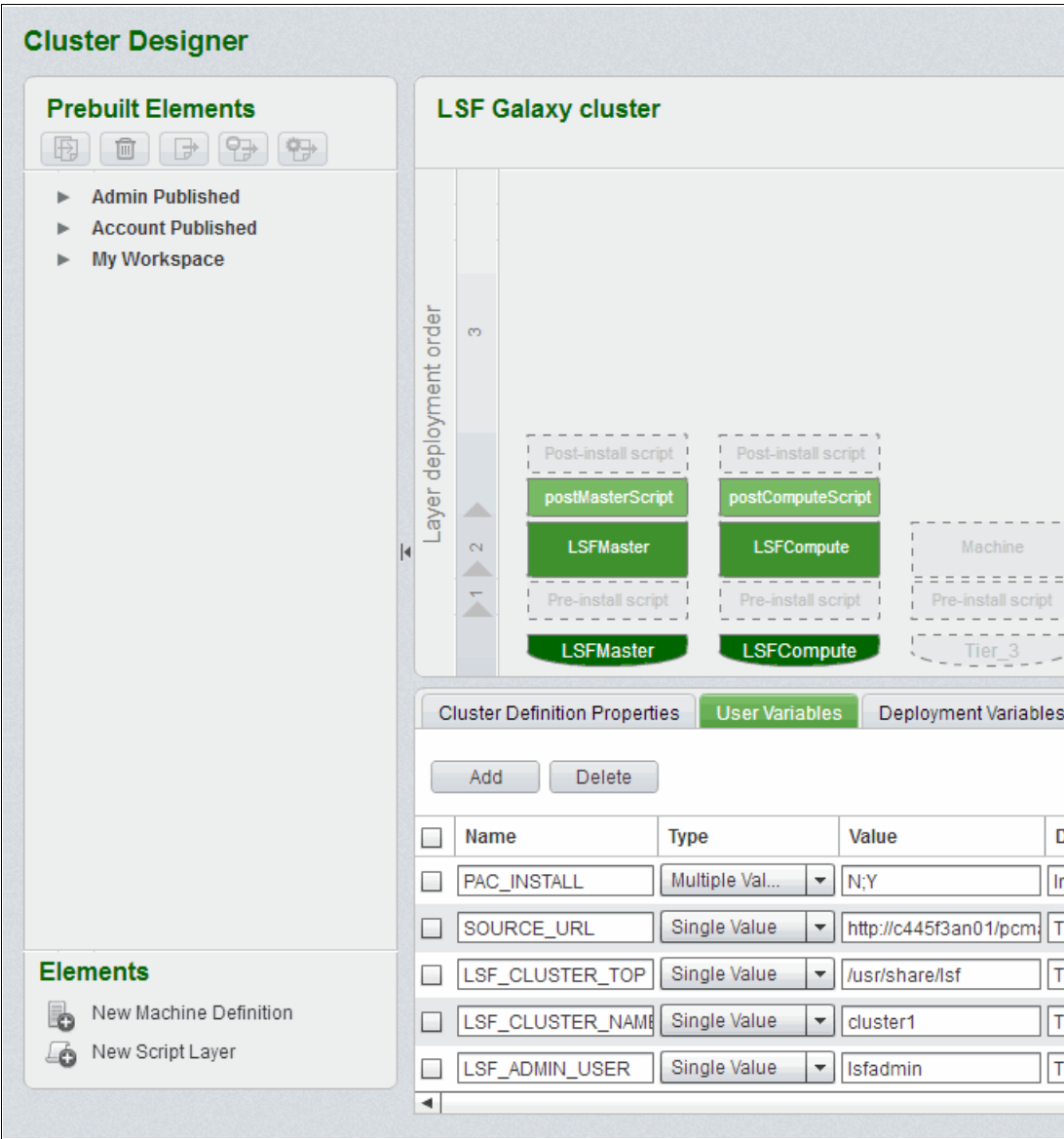


Figure 8-17 Using the cluster designer to modify the default LSF cluster definition to provision Galaxy

Figure 8-18 shows a three node Galaxy cluster deployed.

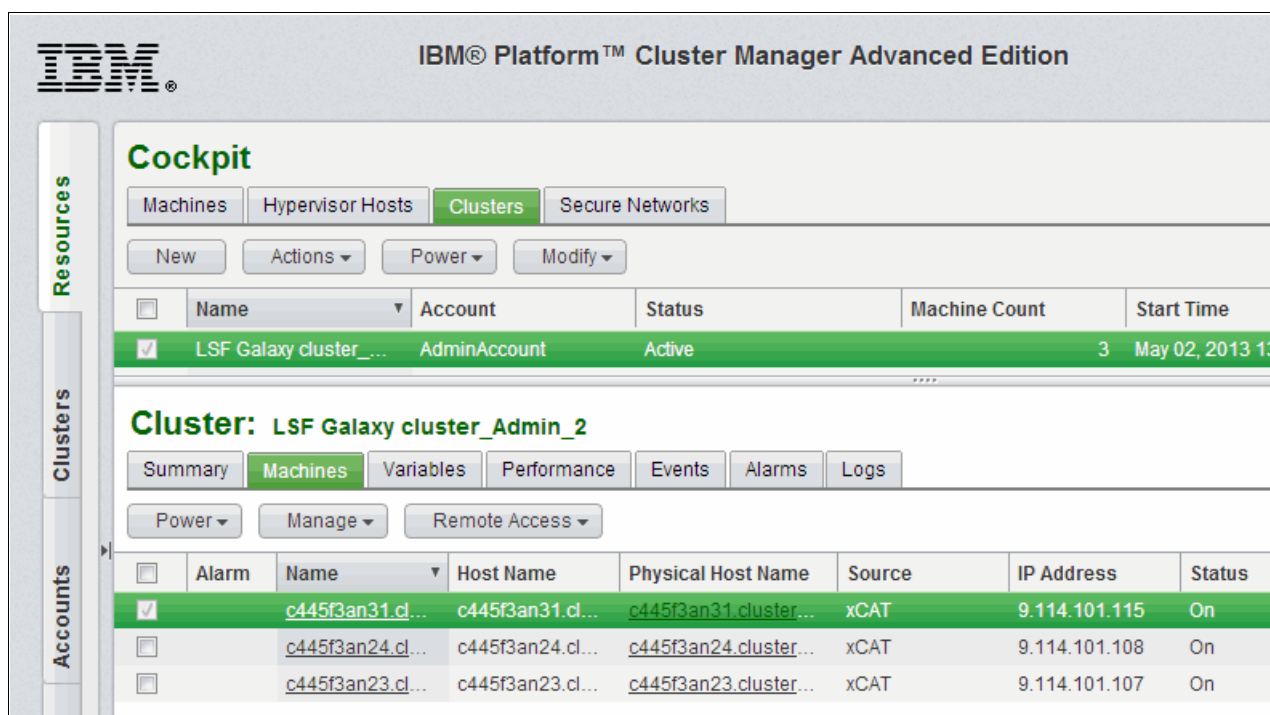


Figure 8-18 A three node Galaxy cluster deployed

Figure 8-19 shows the Galaxy interface.



Figure 8-19 Accessing the Galaxy interface

Figure 8-20 shows how to edit workflows in Galaxy.

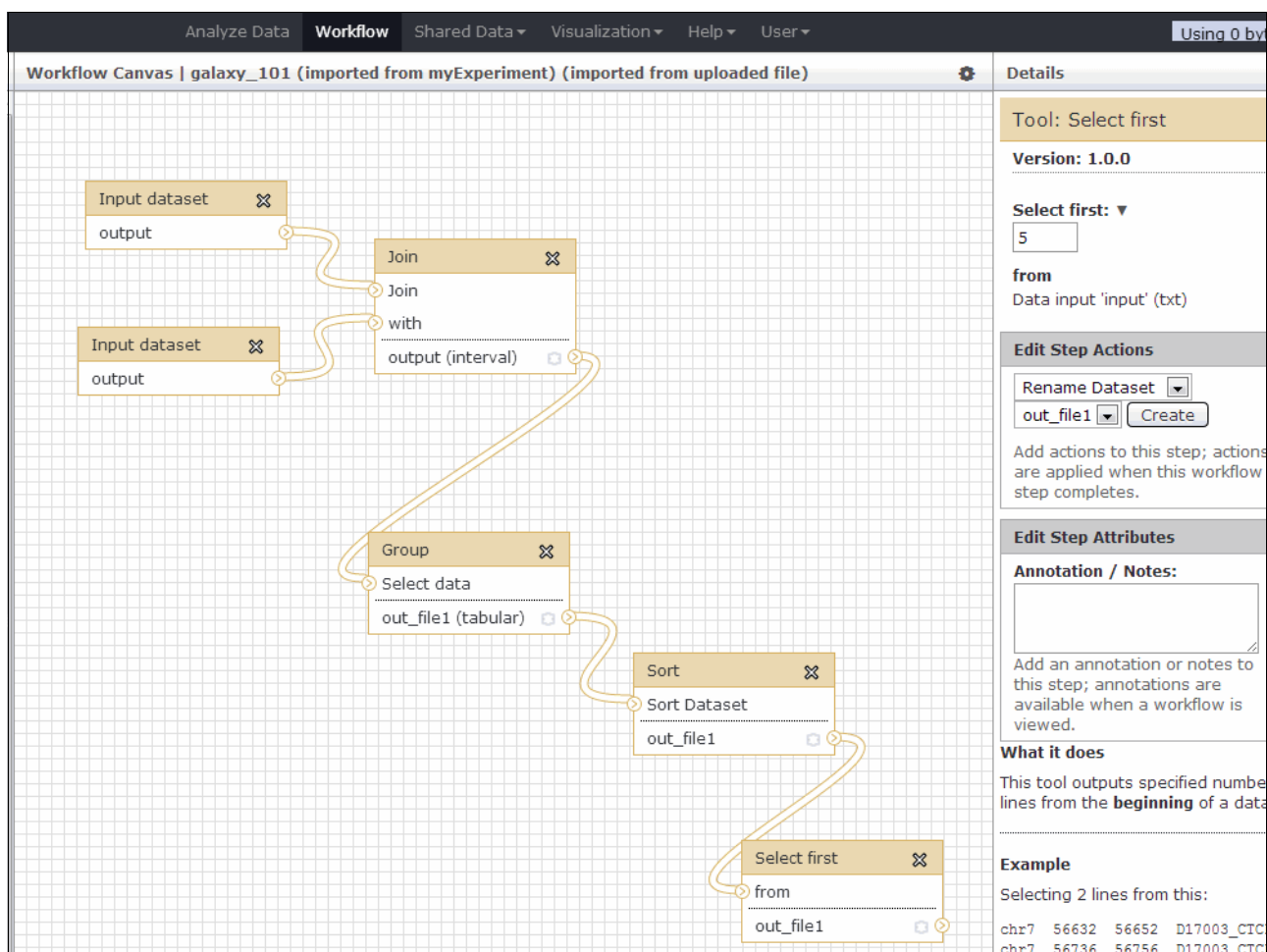


Figure 8-20 Editing workflows in Galaxy

Platform LSF integration

This section describes the LSF integration aspects with Galaxy.

Test environment

This section shows the characteristics of the test environment.

The c445 cluster has the following characteristics:

- ▶ Job runs on 1G Ethernet
- ▶ GPFS internal disk configuration
- ▶ GPFS 12 disks from server
- ▶ SAS controller
- ▶ GPFS connect to compute through IB QDR
- ▶ 3 M2 compute nodes

Hardware configuration

The hardware configuration of the node includes these components:

- ▶ IBM System x iDataPlex dx360 M2 server
- ▶ 1x 1GE on-board adapter (management)
- ▶ Mellanox Technologies MT26428 ConnectX VPI PCIe 2.0 5 GT/s - IB QDR

- ▶ 24 G RAM
- ▶ 2 x Intel Xeon Processor X5500 4C
- ▶ 220 GB IDE HDD12 * IBM 3 TB 7.2 K 6 Gbps NL SAS 3.5" HS HDD

Software configuration

The environment uses these high performance computing software components:

- ▶ Platform Cluster Management Advance Edition 4.1
- ▶ Platform LSF 9.1
- ▶ General Parallel File System 3.5

The environment uses these life sciences software components:

- ▶ BWA 6.2
- ▶ Picard 1.56
- ▶ SAMTools 1.18
- ▶ GATK-lite 2.3.9
- ▶ Galaxy 788cd3d06541 distribution level w/20130502 update



Solution for financial services workloads

This chapter describes challenges that are faced by financial institutions, and how technical cloud computing can be used to help solve them. It also describes solution architecture topics, and provides use case scenarios to help solve financial workloads.

This chapter includes the following sections:

- ▶ Overview
- ▶ Architecture
- ▶ Use cases
- ▶ Third-party integrated solutions

9.1 Overview

In today's world, financial institutions are under increasing pressure to solve certain types of problems. Online transaction fraud has increased over time, and accounts for billions of dollars already. Money laundering costs governments to lose billions of tax dollars that could be invested in infrastructure or services for the country.

Simulation of scenarios based on past data or real-time analysis are in high demand. Today, institutions base their decisions much more on heavy data analysis and simulation than on feeling and experience alone. They need as much data as possible to guide these decisions to minimize risks and maximize return on investment.

Current regulatory requirements nowadays also push for a faster risk analysis such as Counterparty Credit Risk (CCR), which requires the use of real-time data. Regulations ALSO require comprehensive reports that are built from a large amount of stored data.

In essence, financial institutions must be able to analyze new data as quickly as it is generated, and also need to further utilize this data later on. These are only a few examples of problems that can be tackled by Platform Computing and business analytics solutions. These solutions can scale up their processing to synthesize, as quickly as required by the business, the amount of data that are generated by or available to these institutions.

9.1.1 Challenges

In addition to these problems and trends, this section provides an insight on some of the challenges that are currently faced by customers in the financial arena:

- ▶ Financial institutions face the need to model and simulate more scenarios (Monte-Carlo) to minimize uncertainty and make better decisions. The number of scenarios required for a certain precision can require very large amounts of data and processing time. Optimization of infrastructure sharing for running these multiple scenarios is also a challenge.
- ▶ Given the costs, it is prohibitive to think about having an isolated infrastructure to provide resources for each business unit and application. However, users fear that they might lose control of their resources and not meet their workload SLAs when using a shared infrastructure.
- ▶ Some problem types are more efficiently solved with the use of MapReduce algorithms. However, the business might require the results in a timely manner, so there is the need for a fast response from MapReduce tasks.
- ▶ Using programming languages such as R and Python to make use of distributed processing. R is a statistical language that is used to solve analytics problems. Python has been increasingly used to solve financial problems due to its high performance mathematics libraries. However, writing native code that is grid-distributed aware is still a difficult and time-consuming task for customers.
- ▶ For data intensive applications, the time to transfer data among compute nodes for processing can exceed calculation times. Also, network can get saturated as the number of data applications that need to be analyzed grows rapidly.
- ▶ Customers want to use the idle capacity of servers and desktops that are not part of the grid due to budgetary constraints. However, the applications that run on these off-grid systems cannot be affected.

IBM has a range of products that can be used to help solve these problems and challenges. These include BigData applications, Algorithmics, and IBM Platform Symphony along with its MapReduce capabilities.

9.1.2 Types of workloads

Workloads can be classified according to two different categories when it comes to the world of finance:

- ▶ Real-time versus long-running (batch)
- ▶ Data intensive versus computing intensive

Figure 9-1 depicts a diagram classifying some of the tasks that are run by a bank according to these categories.

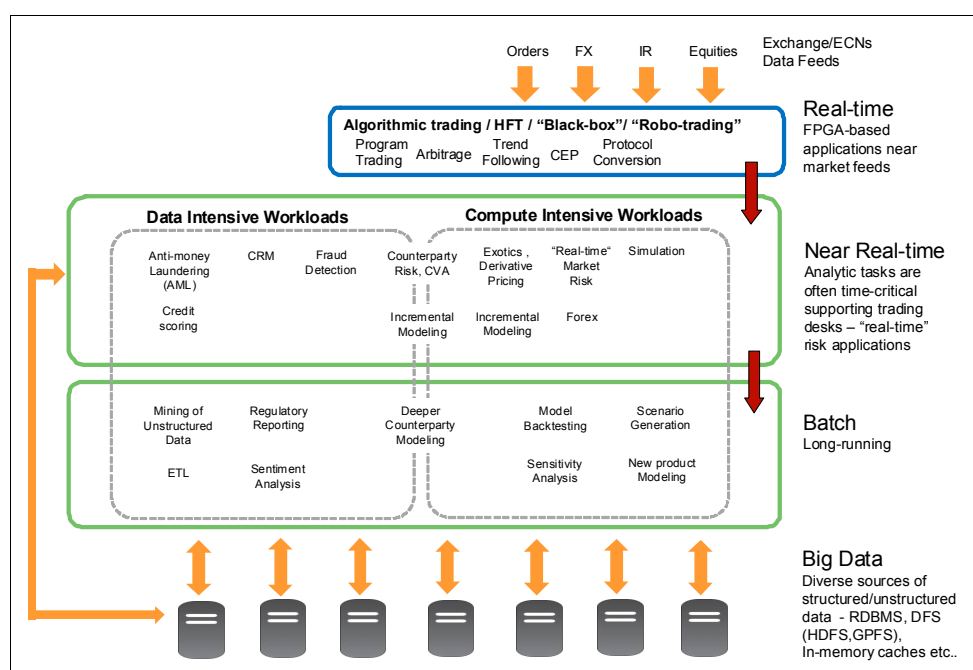


Figure 9-1 Diagram classifying tasks according to workload types at a bank

Notice in Figure 9-1 that the environment is constantly fed with data such as electronic orders, foreign exchange rates, interest rates, and equity values. This information is used to feed real-time applications such as algorithmic trading. This type of rapid, massive data analysis can be run with, for example, IBM InfoSphere Streams.

In the next layer, decreasing in terms of need for time-critical results, come the near real-time workloads. The acquired data, after they are analyzed by real-time applications, can be used to perform analysis such as fraud detection, run Anti-Money Laundering (AML) routines, near real-time market risk analysis, simulations, and others. In this layer, you can still split the workloads into two groups: Compute intensive and data intensive. Long running applications are not the only ones that must deal with large amounts of data. Near real-time ones such as AML need to be able to quickly detect fraud or laundering activities by analyzing large amounts of data generated by millions of financial transactions. Some workloads are mixed in that sense, such as counterparty risk analysis, which is both compute and data intensive.

Going down one layer, you reach the non-real-time, long-running, batch jobs responsible for creating reports, mining unstructured data, model back-testing, scenario generation, and others. Again, workloads can be classified as compute or data intensive, or both.

The lowest layer of Figure 9-1 on page 201 depicts a diverse set of hardware and technologies that are used as infrastructure for all of these workloads. Below them all lies a vast amount of data that was not able to be analyzed until now. The finance world has become a truly BigData environment.

The following is a more comprehensive list of workloads that are commonly tackled by financial institutions:

- ▶ Value at risk (VaR)
- ▶ Credit value adjustments (CVAs) for counterparty CCR
- ▶ Asset liability modeling (ALM)
- ▶ Anti-money Laundering
- ▶ Fraud detection
- ▶ Sensitivity analysis
- ▶ Credit scoring
- ▶ Mortgage analytics
- ▶ Variable annuity modeling
- ▶ Model back testing
- ▶ Portfolio stress testing
- ▶ Extraction, transformation, and load (ETL)
- ▶ Strategy mining
- ▶ Actuarial analysis
- ▶ Regulatory reporting
- ▶ Mining of unstructured data

9.2 Architecture

The example software architecture to engage common financial workloads is shown in Figure 9-2. The architecture is based on BigData components that use IBM Platform Symphony's ability to effectively manage resource grids and provide low latency to applications.

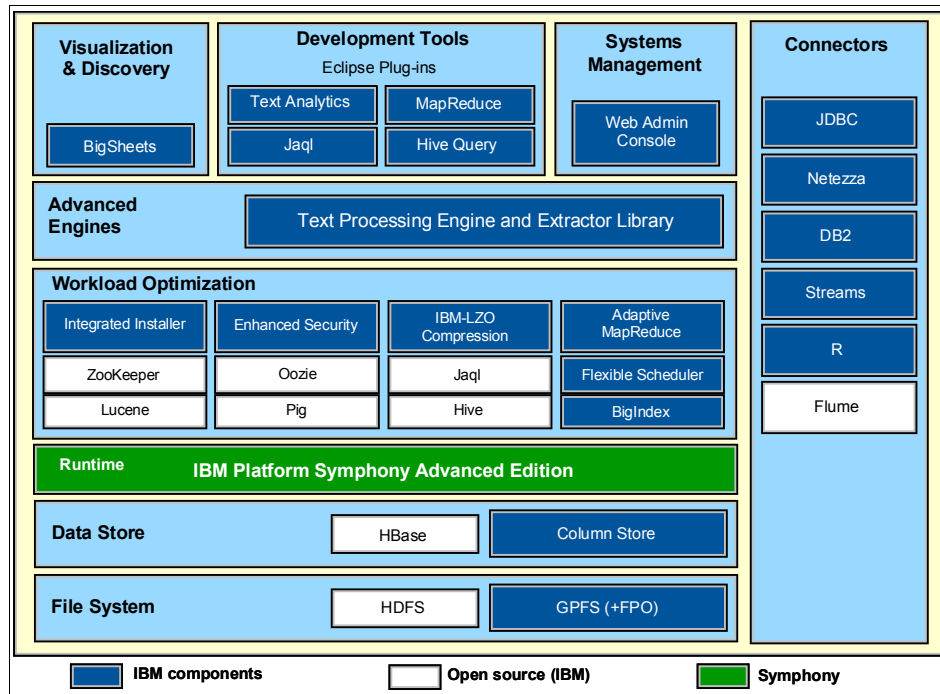


Figure 9-2 Platform Symphony-based software architecture for running financial workloads

The architecture depicted in Figure 9-2 is composed of multiple layers. Notice that some of the layers offer the option to use open source software, which are identified as white boxes. The blue boxes denote IBM components.

As a middleware, Platform Symphony appears in a middle layer of the architecture. It is able to control and schedule computational resources within the grid and effectively share them among applications. The software components underneath it are related to data handling such as file system software components and data store software components. For file systems, the architecture can be built using GPFS or Hadoop. Known technologies for storing data, especially in a BigData environment, are HBase and Column Store.

Above Platform Symphony is the application layer that uses its low-latency and enhanced MapReduce capabilities. As you can see in Figure 9-2, many technologies can be integrated into this Platform Symphony-based reference architecture.

Finally, the architecture uses connectors to allow communication with other data sources (Netezza®, IBM DB2®, Streams, and others) as depicted in Figure 9-2.

9.2.1 IBM Platform Symphony

IBM Platform Symphony can be applied as a middleware layer to accelerate distributed and analytics applications running on a grid of systems. It provides faster results and a better utilization of the grid resources.

With Platform Symphony, banking, financial markets, insurance companies, among other segments, can gain these benefits:

- ▶ Higher quality results are provided faster.
Run diverse business-critical compute and data intensive analytics applications on top of a high performance software infrastructure.
- ▶ Reduction of infrastructure and management costs.
A multi-tenant approach helps achieve better utilization of hardware, minimizing the costs of hardware acquisition, cost of hardware ownership, and simplifying systems management tasks.
- ▶ Quick responses to real-time demand.
Symphony uses push-based scheduling models that save time compared to polling-based schedulers, allowing it to respond almost instantly to time-critical businesses. This can provide a great boost to MapReduce based applications.
- ▶ Management of compute-intensive and data-intensive workloads under the same infrastructure.
Financial businesses workloads are diverse. However, you do not need to create separate environments to be able to process each type of workload separately. Symphony can efficiently schedule resources of the same computing grid to meet these workload requirements.
- ▶ Harvesting of desktop computers, servers, and virtual servers with idle resources.
Symphony can use desktop and server resources that are not part of the computing grid to process workloads without impacting their native applications. Tasks are pushed to these extra resources when they are found to be idle. Virtual servers such as VMware and Citrix Xen farms can also be harvested.
- ▶ Integration with widely used programming languages in the financial world, such as R and Python.
Through Symphony, customer written code can use the grid middleware to handle some aspects of distributed programming.
- ▶ Data-aware scheduling of compute and data intensive workloads.
Platform Symphony schedules tasks to nodes where the data is found to be local whenever possible, which improves performance and efficiency.

IBM Platform Symphony can suit both classifications that were introduced in 9.1.2, “Types of workloads” on page 201: Real-time versus long-running, and compute versus data-oriented workloads.

Figure 9-3 describes which layers Platform Symphony can act upon from a time-critical classification point of view. If you compare it to Figure 9-1 on page 201, you can see that there is a match between them. The second and third stages of data flow in Figure 9-3 correspond to the second and third layers (delimited by the green rectangles) in Figure 9-1 on page 201.

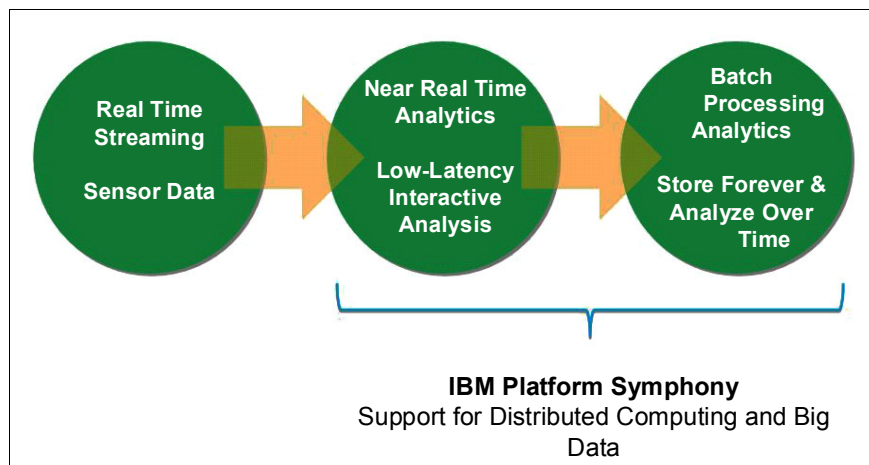


Figure 9-3 Platform Symphony's support for time-critical data processing

Similarly, there is a match between Platform Symphony's component architecture as depicted in Figure 9-4 with Figure 9-1 on page 201. Symphony can provide a low-latency service-oriented application middleware for compute intensive workloads, and also an enhanced, highly performing framework for massive data processing that is based on MapReduce algorithms. Notice that Platform Symphony is also able to provide both of these for workloads that are both compute and data intensive.

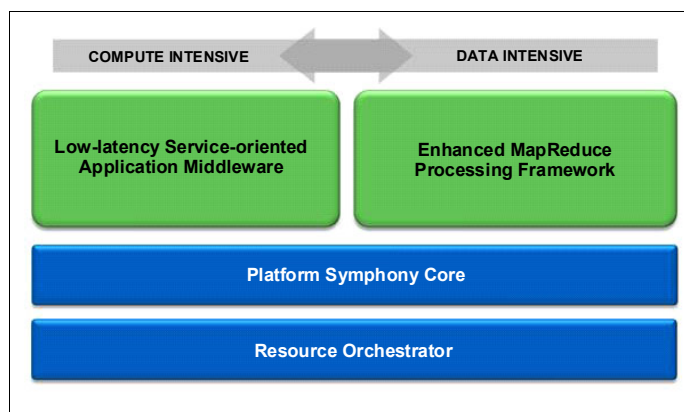


Figure 9-4 Platform Symphony middleware for compute and data intensive workloads

Details about Symphony's internal components can be found in Chapter 9, "Solution for financial services workloads" on page 199 and Chapter 4, "IBM Platform Symphony MapReduce" on page 59.

IBM Platform Symphony MapReduce

IBM Platform Symphony contains its own framework for dealing with MapReduce processing. This framework allows multiple users to have access to the grid resources at the same time to run the MapReduce jobs of an application. This is an improvement over Hadoop's MapReduce framework, where jobs are scheduled to run sequentially, one at a time, having a single job consume grid resources as much as possible. Platform Symphony can schedule

multiple jobs to the grid at the same time by sharing the grid resources based on job priority, job lifetime (shorter, longer), user SLAs, and so on.

Another advantage of using Platform Symphony as an architecture component is that it can manage the co-existence of both MapReduce and non-MapReduce applications in the same grid. This characteristic avoids the creation of resource silos by allowing different workload types to use the grid resources. Therefore, a financial institution does not need to have different grids to process its variety of workloads. Instead, it can use a single analytics cloud to run all of its workloads as depicted in Figure 9-5.

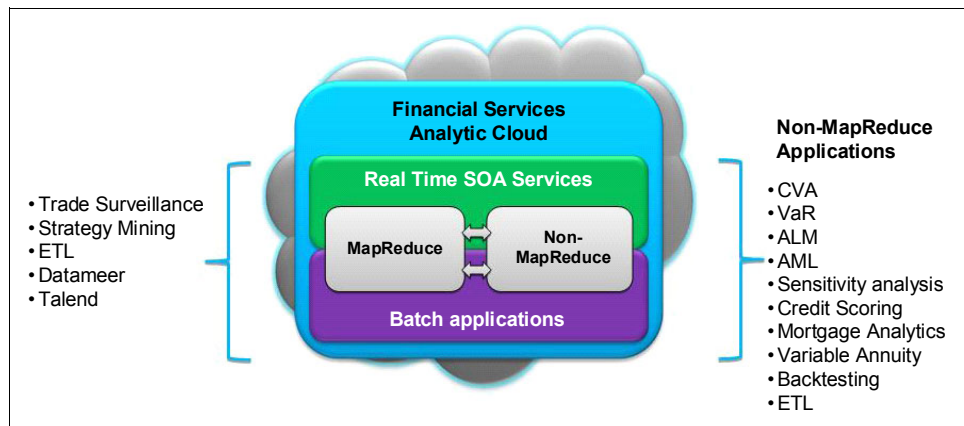


Figure 9-5 Single financial services analytic cloud

9.2.2 General Parallel File System (GPFS)

With a large quantity of financial problems using MapReduce algorithms to get to results, many frameworks use the Hadoop Distributed File System (HDFS) to store data.

HDFS is well suited for what it is intended to do: Serve data intended for MapReduce tasks in a cluster of machines. However, HDFS lacks capabilities for file system operations, so it cannot be used as a multi-purpose file system:

- Data accessed through Java application programming interfaces (APIs).

To access data, users must interface their own code with APIs, or use a set of utilities for doing so (FS shell). In either case, there is no direct linkage between the files and the operating system.

- HDFS is not built with POSIX standards.

There is no direct linkage between the files within HDFS and the operating system. Consequently, users must load their HDFS space with the data before consuming it. The time required can be significant for large amounts of data. Also, users might end up with the data stored twice: Inside HDFS for consuming it with MapReduce applications, and on the operating system for manipulating the data easily.

User-space file systems technologies integrated to the Hadoop MapReduce framework might alleviate data handling operations, but then the performance of a user-space file system becomes a disadvantage.

- Single-purpose built file system.

HDFS was built to serve as Hadoop's file system for providing data to MapReduce applications. It is not suited to be used as a file system for other purposes.

- Optimized for large data blocks only.

Small or medium sized files are handled in the same way as large files, making HDFS not so efficient at handling them. Because there are numerous sources of varying characteristics for BigData today, this also adds more disadvantages to the file system.

- HDFS metadata is handled in a centralized way.

Although it is possible to provide a certain level of metadata high availability with the primary and secondary name nodes for HDFS, this data is restricted to these nodes only.

To overcome these characteristics, GPFS with its new File Place Optimizer (FPO) feature can be considered as an enterprise-class alternative to HDFS. FPO makes GPFS aware of data locality, which is a key concept within Hadoop's MapReduce framework. With that, compute jobs can be scheduled on the computing node for which the data is local. The new FPO feature is explained in 6.4.2, "File Placement Optimizer (FPO)" on page 129.

GPFS has an established reputation as a distributed file system compliant with POSIX standards. Therefore, it is part of the operating system, and applications can use it by using the same system calls used to manage data with any file system (open, close, read, write, and so on). There is no need to load data onto a MapReduce framework before consuming it. Captured BigData from multiple sources are stored in GPFS and are immediately available for consumption.

Also, GPFS is a multi-purpose file system. As such, different types of application can consume the same resident data on the storage devices. There is no need to replicate data depending on how it is supposed to be consumed. This means that you can have, for example, both MapReduce and non-MapReduce based applications of a workflow using the same set of data, avoiding the need for duplication.

GPFS provides data access parallelism, which represents an increase in performance when running multiple simulation scenarios that access the same data. This means that financial institutions can run more scenarios at once, increasing decision accuracy because more simulation results are available.

Support for both large and small blocks is available in GPFS, as well as support for various data access patterns, giving applications a high level of performance.

Finally, GPFS provides automated data replication to remote sites, thus being an integrated solution for both high performing data I/O and data backup to ensure business continuity.

9.2.3 IBM Platform Process Manager (PPM)

PPM can automate the execution of jobs by the creation of flow definitions. This helps organize tedious, repetitive tasks and their interconnections. Sophisticated flow logic, subflows, alarm conditions, and scriptable interfaces can be managed by PPM. By doing so, Process Manager minimizes the risk of human errors in processes and provides reliability to processes.

Platform Process Manager is part of the Platform LSF suite, and can be integrated on a grid that is managed by IBM Platform LSF, or a multiheaded grid that is managed by Platform LSF and Platform Symphony as described in Figure 9-6.

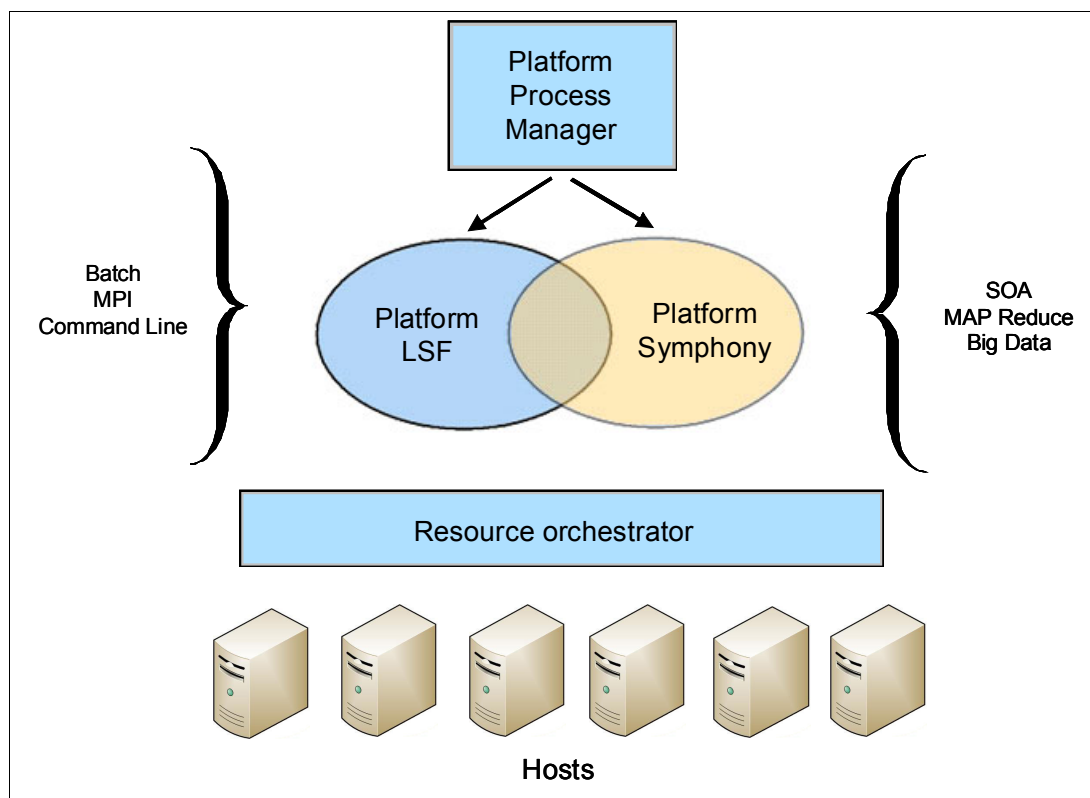


Figure 9-6 Platform Process Manager in an LSF and Symphony multiheaded grid architecture

Platform Process Manager is Platform LSF aware. That is, its internal components are able to understand Platform LSF constructs such as job arrays, and choose processing queues. PPM is deployed to work more with Platform LSF in most of the cases.

However, a PPM flow step can interact with Platform Symphony by starting a client that in turn connects to Platform Symphony for job execution. For example, imagine a flow in which the first step is the creation of simulation scenarios. A second step can then get these scenarios and pass them along to an application running on top of Platform Symphony so they can be run.

The flexibility of using both Platform LSF and other applications such as Platform Symphony-based applications, is suited to environments that are composed of both batch and service-oriented jobs. Platform LSF can be used to manage batch-oriented jobs, whereas Platform Symphony can be used to run MapReduce or other SOA framework-based applications.

For more information about how to create a multiheaded grid environment, see *IBM Platform Computing Solutions*, SG24-8073.

Platform Process Manager can use the advantages of a distributed grid infrastructure as it provides these benefits:

- ▶ Resource-aware workflow management.

Platform Process Manager works by qualitatively describing the resources that it needs to run a workflow. Therefore, jobs are not tied to particular host names. This creates a more efficient workflow execution because you can use resource schedulers, such as Platform Symphony or Platform LSF, to deploy jobs on other available grid nodes that can satisfy their execution instead of waiting for a particular node to become available.

- ▶ Built-in workflow scalability.

As your grid has its amount of resources increased, PPM can dynamically scale the workflow to use the added resources. No changes to the workflow definitions are required.

- ▶ Multi-user support.

Platform Process Manager can handle multiple flows from different users at the same time. Flow jobs are sent to the grid for execution, and each flow can use part of the grid resources.

The following bullets summarize the benefits of using PPM for automating flow control:

- ▶ Integrated environment for flow designing, publishing, and managing.

- ▶ Reliability provided by rich conditional logic and error handling.

- You can inspect variables status and define roll-back points in case of failure. Roll-back points are useful to avoid users having to treat errors that can be resolved by trying the particular task again.

- ▶ Modular management that provides flows and subflows with versioning control.

- ▶ Flow execution based on schedule or triggering conditions.

- ▶ Intuitive graphical interfaces as shown in Figure 9-7 on page 210.

- No programming skills required.

- Graphical dynamic flow execution monitoring.

- ▶ Self documenting of flows.

- ▶ XML-based format is used to save workflows.

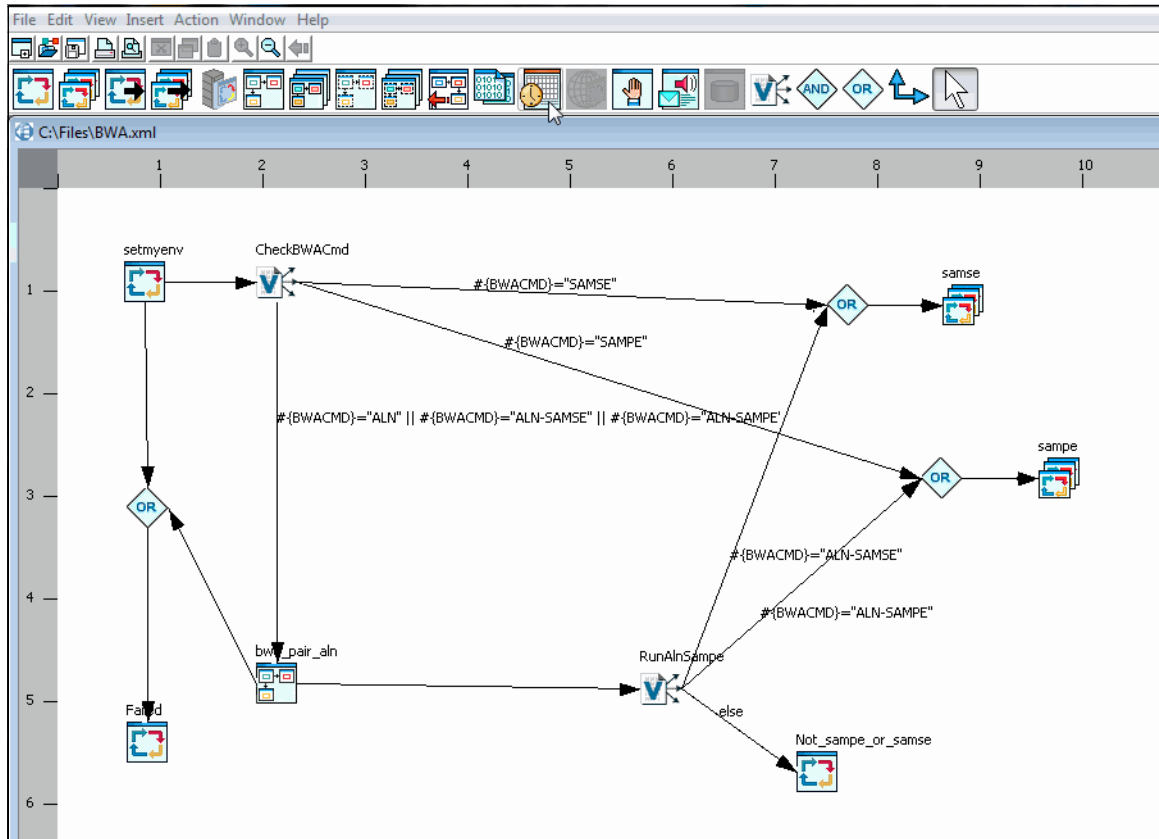


Figure 9-7 Platform Process Manager flow editor.

For a comprehensive user guide on Platform Process Manager, see *Platform Process Manager Version 9 Release 1* at:

<http://www.ibm.com/shop/publications/order>

9.3 Use cases

This section provides use case scenarios to complement the theoretical details described in previous sections.

9.3.1 Counterparty CCR and CVA

An active enterprise risk management strategy relies on having an up-to-date aggregate view of corporate exposures. These include accurate valuations and risk measurements, reflecting CVAs of portfolios and new transactions. To understand risks enterprise-wide, pricing and risk analytics can no longer be done in silos or in an ad hoc fashion. The need to apply accurate risk insights while making decisions throughout the enterprise is driving firms to consolidate risk systems in favor of a shared infrastructure.

The CVA system is an enterprise-wide system that needs to take input from both commercial and proprietary risk/trading systems (seeing what is in the portfolios). It then aggregates the input to determine the counterparty risk exposures. Monte Carlo simulations are the best way to do the CVA calculation. Platform Symphony grid middleware is particularly suited for running Monte Carlo because of its high scalability and throughput on thousands to tens of

thousands of compute nodes. As new regulations and new financial products become available, the CVA system must adapt to respond to these changes.

Key requirements for enterprise risk management solutions

The following highlights these key considerations:

- ▶ Enterprise-scale, across asset classes and deal desks
- ▶ Provide full Monte Carlo Simulations
- ▶ Support proprietary and third-party risk/trading systems
- ▶ Aggregation of results with netting
- ▶ Intraday or faster with high throughput
- ▶ Cost efficient solution
- ▶ Agile by enabling change/update of new models

Applicable markets

This list highlights the applicable solution markets:

- ▶ Tier-one investment banks running predominantly in-house analytic models
- ▶ Tier-two banks running ISV applications and in-house models
- ▶ Hedge funds, pension funds, and portfolio managers
- ▶ Insurance companies and exchanges

IBM Platform Symphony for integrated risk management solution

An active risk management approach requires a highly scalable infrastructure that can meet large computing demands, and allow calculations to be conducted in parallel. Grid computing is a key enabling, cost-effective technology for banks to create a scalable infrastructure for active risk management. Grids are used in various areas, including pricing of market and credit risk, compliance reporting, pre-trade analysis, back testing, and new product development.

Past technical limitations with some applications and grid technologies often created multiple underused compute “silos”. This results in increased cost and management complexity for the entire organization. Newer applications and grid technologies have overcome these limitations, allowing for effective resource sharing by maximizing utilization while containing costs.

With effective grid management in place, grids should be available instantly and on demand to the users with the highest priority. The grid applications can dynamically borrow computing resources from lower priority work already in process, achieving a higher overall utilization. In addition, priority requests are served quickly at a lower overall cost.

The solution is based on IBM Platform Symphony, which provides a solution for data intensive and distributed applications such as pricing, sensitivity analysis, model back-testing, stress-testing, fraud-detection, market and credit risk, what-if analysis, and others.

Platform Symphony also uses its data affinity feature to optimize workload and data distribution to remove bottlenecks and maximize performance.

Another important feature of Platform Symphony is the ability to oversubscribe each computation slot to handle recursive type jobs over the data already available on a particular node. This parent-child relationship of created jobs prevents unnecessary data movement and is key to achieving maximum performance.

Using IBM Algorithmics as an example (Figure 9-8), the Algorithmics products and Platform Symphony are integrated to enable faster time-to-completion of complex analytics workloads such as CVA. This is particularly useful for compute intensive tasks such as simulation (integration of IBM RiskWatch® and Platform Symphony).

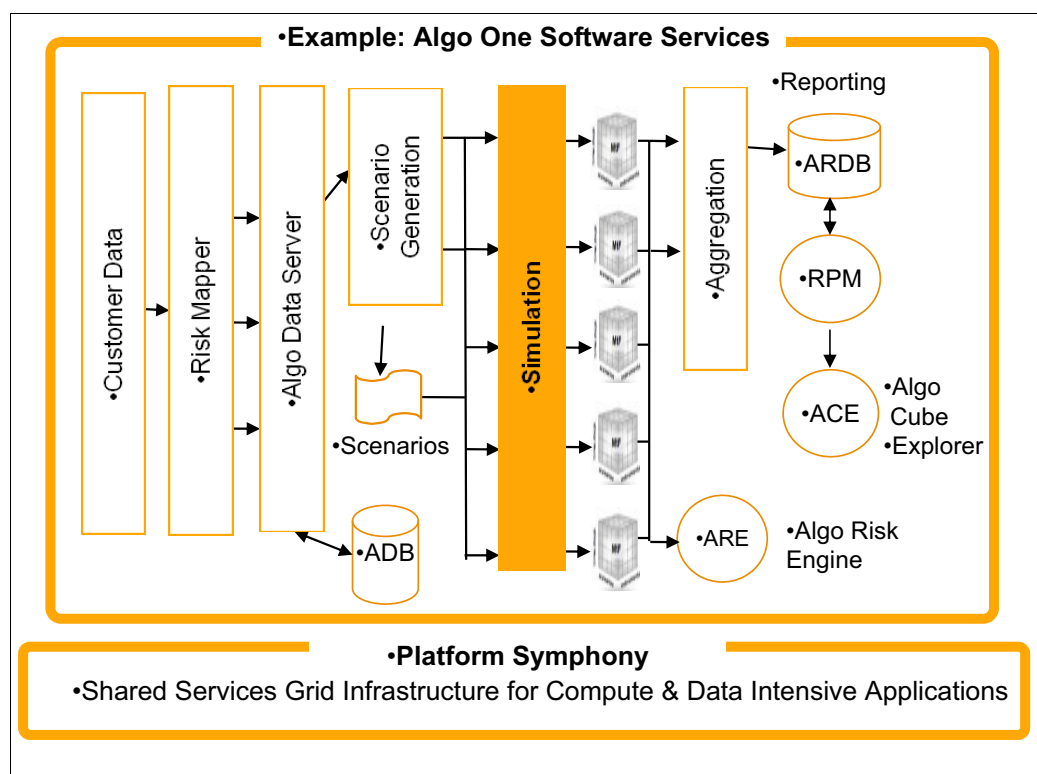


Figure 9-8 Risk management solution: Platform Symphony and IBM Algo One® software services

Platform Symphony also supports high availability. If a node fails, an automated failover restarts only the task that failed rather than the entire job. In testing, this integration has been proven to lower deployment risks and costs. Both Algorithmics and IBM Symphony support heterogeneous systems, allowing you a choice of where to run the integrated solution.

Benefits

This section describes the benefits of the solution:

Scalability

CVA applications demand both scale and low latency. Platform Symphony enables higher fidelity simulations and better decisions making in less time.

Agility

Platform Symphony is unique in its ability to respond instantly to changing real-time requirements such as pre-deal analysis and limit checks, and hedging.

Resource sharing

Platform Symphony enables the same infrastructure to be shared between line of business (LOBs) and applications with flexible loaning and borrowing for sensitivity analysis, stress runs, convergence testing, and incremental CVA.

Smarter data handling	The efficient built-in data distribution capability combined with intelligent data affinity scheduling meet data handling demands for risk exposures calculation across multiple LOBs.
Reliability	Scheduling features and redundancy help ensure critical tasks are run within available time windows.

9.3.2 Shared grid for high-performance computing (HPC) risk analytics

The shared grid solution for risk analytics is a platform as a service (PaaS) infrastructure for scalable shared services. In today's global economic scenario, where IT requirements are increasing but budgets are flat, the pressure to deploy more capability without incremental funding is always present in the financial services sector.

Here are some IT challenges the shared grid model aims to address:

- ▶ Internal customers need to self-provision infrastructure faster and cheaper.
- ▶ Costly and slow to deploy new applications.
- ▶ Need to preserve SLAs, leading to incompatible technology “silos” that are underused and costly to maintain.
- ▶ LOB peak demands are either unsatisfied or cause over-provisioning of infrastructure to meet demand and low utilization.
- ▶ Effective approach to share and manage resources across geographically dispersed data centers.
- ▶ Business units and application owners are reluctant to get onboard with a shared infrastructure project for fear of losing control and jeopardizing core functions.

Applicable markets

The following is a list of applicable markets:

- ▶ Financial organizations that seek to build a private cloud infrastructure to support risk analytics environments.
- ▶ Service providers that offer infrastructure as a service (IaaS) or software as a service (SaaS) solutions that are related to risk analytics.
- ▶ Banks, funds and insurance companies that seek to reduce internal IT costs.

Solution architecture

Platform Symphony helps to consolidate multiple applications and lines of business on a single, heterogeneous, and shared infrastructure. Resource allocations are flexible, but resource ownership is guaranteed. Resources are allocated to improve performance and utilization while protecting SLAs.

Platform Symphony can harvest off-grid resources such as corporate desktops, workstations, and virtual machine hypervisors to expand the available resource pool. Platform Symphony also supports the Platform Analytics plug-in for chargeback accounting and capacity planning to address IT-related risk.

Platform Symphony also supports a wide range of optimized application integrations, including IBM Algorithmics, Murex, R, SAS, and multiple third-party ISV applications.

Benefits

The following are the benefits of the solution:

Reliability	Platform Symphony delivers critical software infrastructure to enable a PaaS for enterprise risk applications
Dynamic provisioning	Application services are deployed rapidly based on real-time application demand and subject to policy.
Multi-tenancy	Platform Symphony enables multiple LOBs with diverse workload to efficiently share resources with commercial applications, homegrown applications, Platform LSF, Platform MPI, Corba, JMS applications, and so on.
High utilization	Symphony maximizes use of data center assets, and can opportunistically harvest capacity on desktops, production servers, and VMware or Citrix server farms without impacting production applications.
Instrumentation	Clear visibility to assets and applications, and usage patterns within the data center or around the globe.
Heterogeneity	Platform Symphony runs across multiple operating environments and supports multiple APIs including C, C++, C#/.NET, Java, R, and Python. It also supports popular IDEs, enabling rapid integration of applications at a lower cost than competing solutions.

9.3.3 Real-time pricing and risk

It is common that traders and analysts lack the simulation capacity needed to adequately simulate risk, thus leading them to use less-precise measures. This leads to missing market opportunities because of the inability to compute risk adequately and in a timely fashion. This inability to quickly simulate the impact of various hedging strategies on transactions reflects directly on reduced profitability.

An agile and flexible infrastructure for time critical problems based on real-time pricing and risk analysis is key for financial institutions struggling to maintain an up-to-date view of enterprise-wide risk.

Applicable markets

The following are the applicable markets for this solution:

- ▶ Investment banks
- ▶ Hedge funds
- ▶ Portfolio managers
- ▶ Pension funds
- ▶ Insurance companies
- ▶ Exchanges

Solution architecture

IBM Platform Symphony allocates target “shares” of resource by application and line of business, ensuring appropriate allocations based on business need. Each application can flex to consume unused grid capacity, enabling faster completion for all risk applications.

In a time-critical requirement, Platform Symphony can respond instantly, preempting tasks and reallocating over 1,000 service instances per second to more critical risk models, temporarily slowing (but not interrupting) less time critical applications.

Benefits

This solution provides the following benefits:

“Instant-on”

Platform Symphony is unique in its ability to rapidly preempt running simulations and run time-critical simulations rapidly with minimal impact to other shared grids.

Low latency

The combination of massive parallelism and ultra-low latency is critical to responding at market speed.

Rapid adjustments

Symphony can rapidly run simulations and dynamically change resources that are allocated to running simulations. For urgent requirements, Platform Symphony can respond faster and get results faster.

9.3.4 Analytics for faster fraud detection and prevention

Due to the increasing growth of Internet-based and credit-card-related fraud, financial institutions have established strong measures to address loss prevention. This includes investing in data analytics solutions to help detect fraud and act as early as possible to record patterns to prevent fraud.

Analytics solution for credit card company

Figure 9-9 illustrates an architecture to provide an end-to-end analytics solution for credit card data. The major components in the colored boxed map are the software and hardware solutions in the IBM big data analytics portfolio.

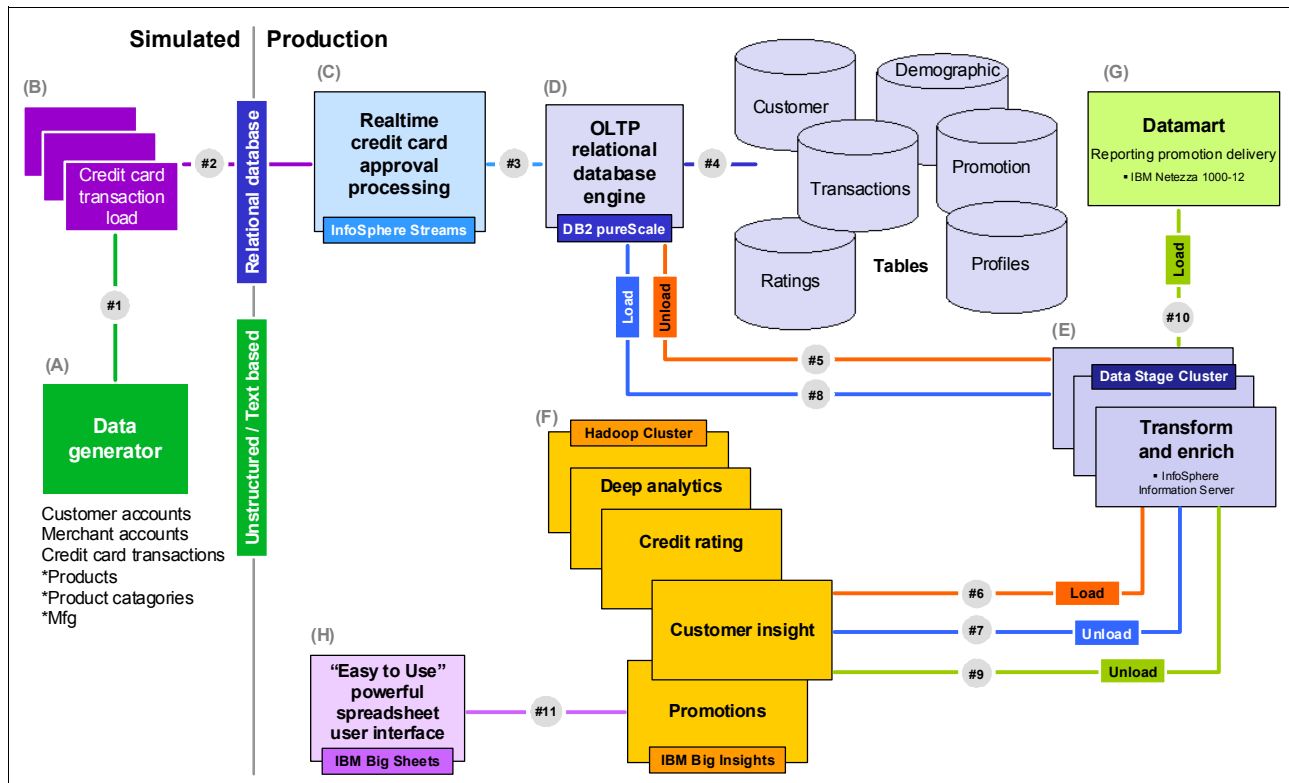


Figure 9-9 Example use case for credit card fraud detection

IBM Platform Symphony and IBM InfoSphere BigInsights integrated solution

This sample use case was built by IBM for show at the Information On Demand (IOD) conference in October of 2012. It was based in a fictional credit card company and its solution for big data analytics workloads. The software was developed to generate synthetic credit card transactions.

A DB2 database stored details such as customer accounts, merchant accounts, and credit card transactions. To handle a high volume of transactions, IBM InfoSphere Streams was used to make real-time decisions about whether to allow credit card transactions to go through.

The business rules in Streams were updated constantly based on credit scoring information in the DB2 database, reflecting card holder history and riskiness of the locale where transactions were taking place.

To automate workflows, and transform data into needed formats, IBM InfoSphere DataStage® was used to guide key processes.

IBM InfoSphere BigInsights is used to run analysis on customer credit card purchases to gain insights about customer behaviors, perform credit scoring more quickly, improve models related to fraud detection, and to craft customer promotions. IBM InfoSphere BigInsights runs its big data workloads on a grid infrastructure that is managed by IBM Platform Symphony.

Continuous analysis run in the IBM InfoSphere BigInsights environment posts results back into the PureScale database. The results of the analytics jobs are also stored in a data mart where up-to-date information is accessible both for reporting and promotion delivery.

By using this less costly architecture, the business is able to gain insights about their operations, and use this knowledge for business advantage (Figure 9-10).

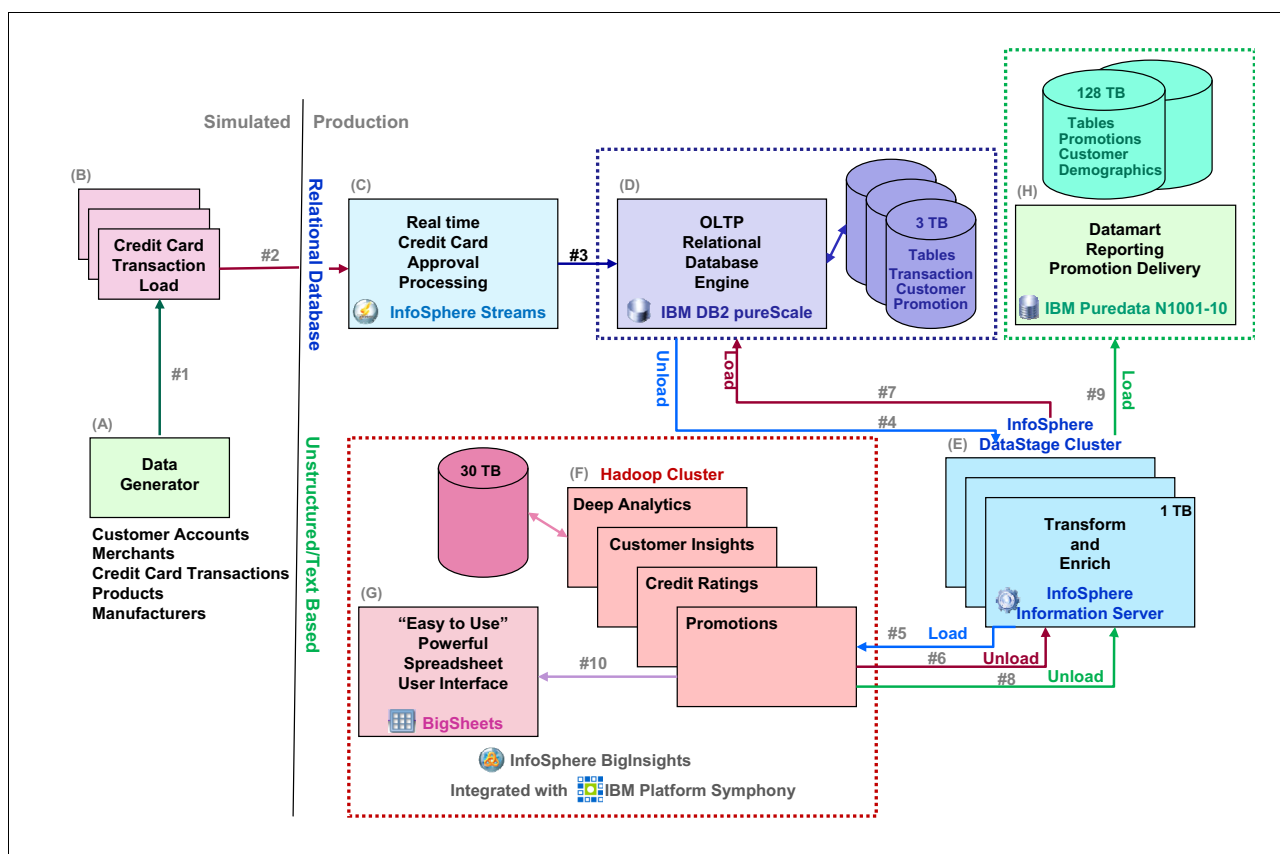


Figure 9-10 Back office integration

Description

The following are the labeled sections in Figure 9-10.

- (A) Data generator generating transaction data
- (B) Credit card transaction load to InfoSphere Streams
- (C) InfoSphere Streams for transaction fraud detection
- (D) DB2 PureScale for credit card transaction processing and storage
- (E) DataStage transforms and enriches relational data:
 - Update master data, unload promotion and reporting information from IBM InfoSphere BigInsights, load it to IBM PureData™ for Analytics
 - Unload customer risk rating from IBM InfoSphere BigInsights, and update master data in DB2 IBM pureScale®
- (F) IBM InfoSphere BigInsights Hadoop cluster for deep analysis of customer buying patterns to generate customer credit risk ratings and promotion offers using Platform Symphony for low-latency scheduling and optimal resource sharing

(G) BigSheets easy-to-use spreadsheet interface to build analytic applications

(H) IBM PureData data warehouse appliance for deep analysis of reporting data that sends out emails for promotions

Data flow

The following are the steps shown in Figure 9-10 on page 217:

1. The data generator generates credit card transactions
2. Credit card transaction approval requests
3. Streams add approved and rejected transactions into DB2 pureScale
4. DataStage unloads transaction data from DB2 pureScale
5. DataStage loads transformed and enriched transaction data into IBM InfoSphere BigInsights
6. DataStage unloads customer credit risk ratings from IBM InfoSphere BigInsights
7. DataStage updates DB2 pureScale with customer credit risk ratings
8. DataStage unloads promotion offers from BigInsights
9. DataStage loads transformed and enriched reporting data to IBM PureData for Analytics data warehouse appliance
10. BigSheets analytic applications access and process customer credit card transaction history

Benefits

The solution has the following benefits:

- ▶ Multi-tenancy
- ▶ Performance
- ▶ Heterogeneity
- ▶ Improved flexibility

9.4 Third-party integrated solutions

This section provides an overview of the independent software vendor (ISV) software that can be used to solve common financial workloads, and how it can be used with the reference architecture presented in 9.2, “Architecture” on page 203.

9.4.1 Algorithmics Algo One

Algorithmics is a company that provides software solutions to financial problems to multiple customers around the world through its ALGO ONE framework platform. Their goal is to allow users to simulate scenarios and understand risks that are associated with them so that a better decision can be made for minimizing risks.

The solutions that are provided by the ALGO ONE platform can be divided into four categories as shown in Table 9-1.

Table 9-1 Software services provided by the ALGO ONE platform

Scenario	Simulation	Aggregation	Decision
<ul style="list-style-type: none"> ► Stress scenarios ► Historical scenarios ► Conditional scenarios ► Monte Carlo scenarios 	<ul style="list-style-type: none"> ► Riskwatch ► Specialized simulators ► Custom models ► Hardware acceleration 	<ul style="list-style-type: none"> ► IBM Mark-to-Future® ► Netting and collateral ► Portfolios ► Dynamic Re-balancing 	<ul style="list-style-type: none"> ► Risk & Capital Analytics ► Real Time Risks & Limits ► Optimization ► Business Planning and What-If

IBM Platform Symphony can be used as a middleware to the ALGO ONE platform of services to use Platform Symphony’s benefits as a multi-cluster, low-latency scheduler. As a result, all of Platform Symphony’s advantages that were presented in 9.2.1, “IBM Platform Symphony” on page 203 can be used by ALGO ONE services.

Figure 9-11 illustrates the interaction of ALGO ONE on a Platform Symphony managed grid.

Note: Other grid client applications (solutions other than Algorithmics) can also use the grid. Therefore, you do not need to create a separate computational silo to run ALGO ONE services on. It can be deployed on top of an existing Symphony grid.

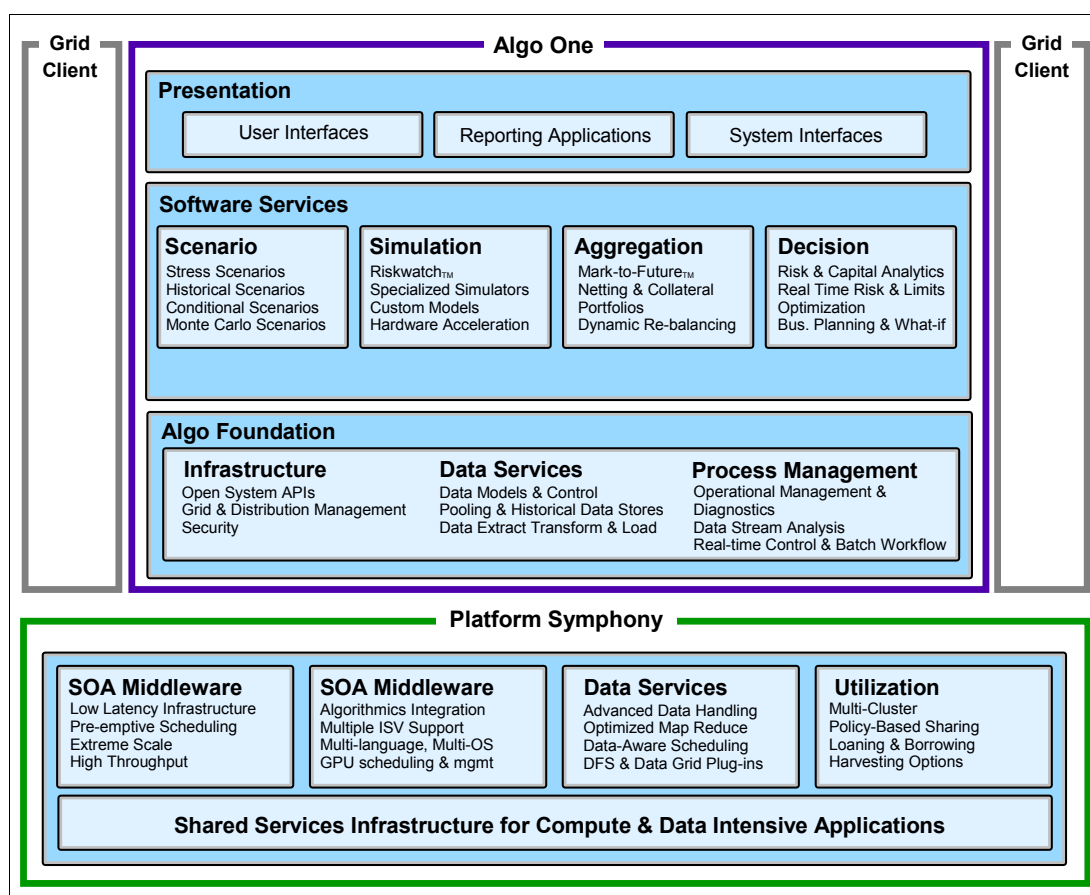


Figure 9-11 A grid managed by Platform Symphony serving ALGO ONE and other client platforms

The following is a list of benefits of integrating ALGO ONE and Platform Symphony:

- ▶ Provides better resource utilization because the grid can be used for diverse multiple tasks at the same time, avoiding the creation of processing silos.
- ▶ Can flex grid resources depending on task priority (bigger, less critical tasks can be flexed down in terms of resources to give room to smaller, more critical tasks such as “What-If” workloads).
- ▶ Allows for the consolidation of risk analysis to provide enterprise-wide results.
- ▶ Can intelligently schedule jobs to nodes that are optimized for a particular task. This is good for credit-risk calculations that require the use of specific lightweight simulators (SLIMs) which require particular operating system and hardware configurations to run.

In summary, ALGO ONE and Platform Symphony allows financial institutions to run more rigorous simulations in shorter time and respond more quickly to time-critical events. They do this while using an easier to manage and flexible grid environment.

9.4.2 SAS

SAS is known for providing business analytics solutions. Financial institutions can use its portfolio of products to aid in decision making, credit-risk analysis, scenario simulation, forecasting of loan losses, probability of defaults on mortgages, and others.

These workloads are heavy due to the amount of data they work with, and can also be compute intensive. To address that, SAS offers its SAS Grid Computing framework for high performance analytics. SAS Grid gives you the flexibility of deploying SAS workloads to a grid of computing servers. It uses the same concept of job dispatching to a set of systems, and so it uses a resource scheduler and resource orchestrator.

SAS Grid is able to use a middleware layer composed of IBM Platform products that manages, and schedules the use of the computing resources. Figure 9-12 illustrates this architecture from a software point of view.

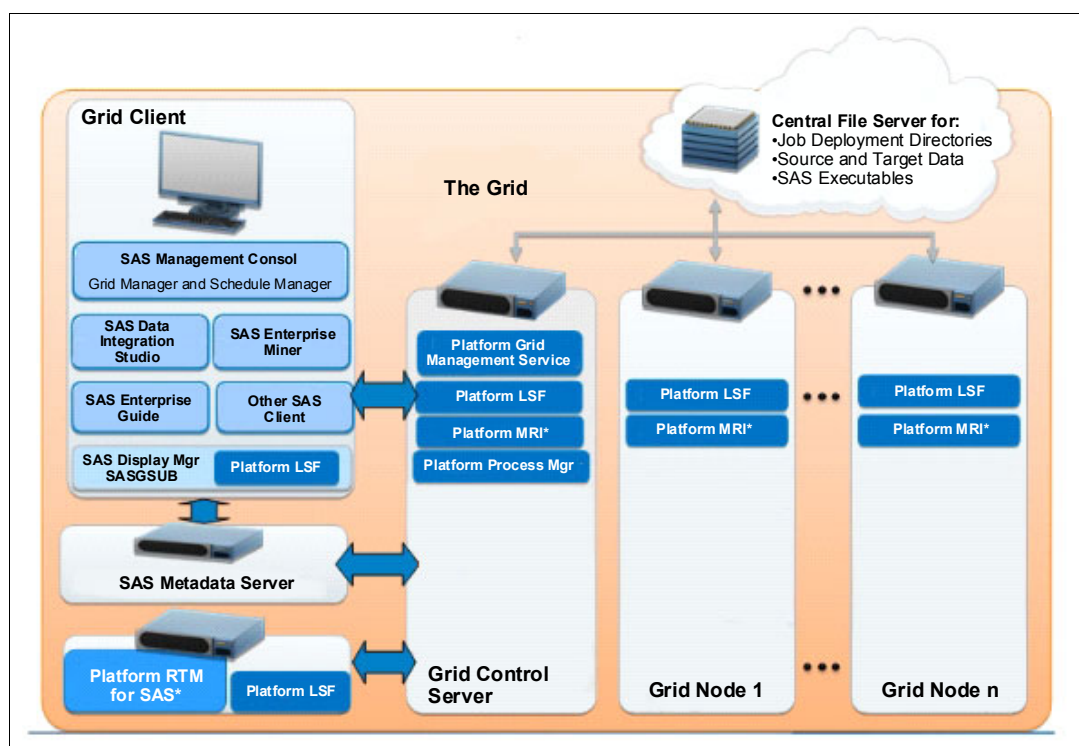


Figure 9-12 SAS Grid software architecture with IBM Platform Computing products

IBM Platform LSF is used as the job scheduler to spread jobs to the grid computing nodes. SAS workloads can make full use of Platform LSF by having it create a full session. They can also use it to, for example, query the load of a particular grid node. With this capability, it is possible for SAS applications to run workload balancing on the grid.

Platform RTM, another component in the architecture shown in Figure 9-12, can provide a graphical interface for controlling grid properties. Platform RTM can be used to perform these tasks:

- ▶ Monitor the cluster
- ▶ Determine problems
- ▶ Tune the environment performance by identifying idle capacity and eliminating bottlenecks
- ▶ Provide reporting
- ▶ Provide alerting functions specific to the Platform LSF environment

Platform Process Manager provides support for creating workflows across the cluster nodes. Platform Process Manager handles the flow management, but uses Platform LSF to schedule and run the steps of a workflow.

Lastly, GPFS can be used as a file server for job deployment directories, source and target data, and SAS executable files.

In summary, these are the benefits of using SAS Grid with the described architecture:

- ▶ Reduced complexity
- ▶ SAS workload management with sophisticated policy controls

- ▶ Improved service levels with faster analysis
- ▶ Provide high availability to SAS environments, which ensures reliable workflows and the flexibility to deploy SAS applications on existing infrastructure



Solution for oil and gas workloads

This chapter provides an architecture reference for oil and gas workloads to be deployed on a technical cloud-computing environment.

This chapter includes the following sections:

- ▶ Overview
- ▶ Architecture
- ▶ Workloads
- ▶ Application software
- ▶ Components

10.1 Overview

The current economic scenario drives the oil and gas industry to pursue new approaches to improve discovery, production, and recovery rates.

There is a complex set of industry forces that are acting in today's oil and gas companies:

- ▶ Energy supply and demand volatility places pressure on production effectiveness.
- ▶ Production complexities push the limits of current technologies and best practices.
- ▶ A changing workforce requires increased productivity and knowledge transfer.
- ▶ New technologies provide opportunities, but require operational changes.
- ▶ Capital and operating expense uncertainty makes it difficult to assess economic viability of assets.
- ▶ Risk and compliance pressures intensify as regulations tighten.
- ▶ Environmental concerns put industry practices under extreme scrutiny.
- ▶ Rising energy needs require new approaches to extend the life of existing fields while also finding and developing new fields.

The demand for innovation creates opportunities to push information technology (IT) boundaries:

- ▶ Greater access to shared, centralized resources for faster insight to complex challenges, with improved efficiency and reduced costs.
- ▶ Improved collaboration by using remote access to increase knowledge sharing, unlocking skills from location and using portable devices.
- ▶ Improved data management and security through centralized tiered storage solutions.
- ▶ Increased operational agility and resiliency through expert integrated systems to support faster time to value.

Figure 10-1 illustrates the trends in the oil and gas industry caused by these driving factors.

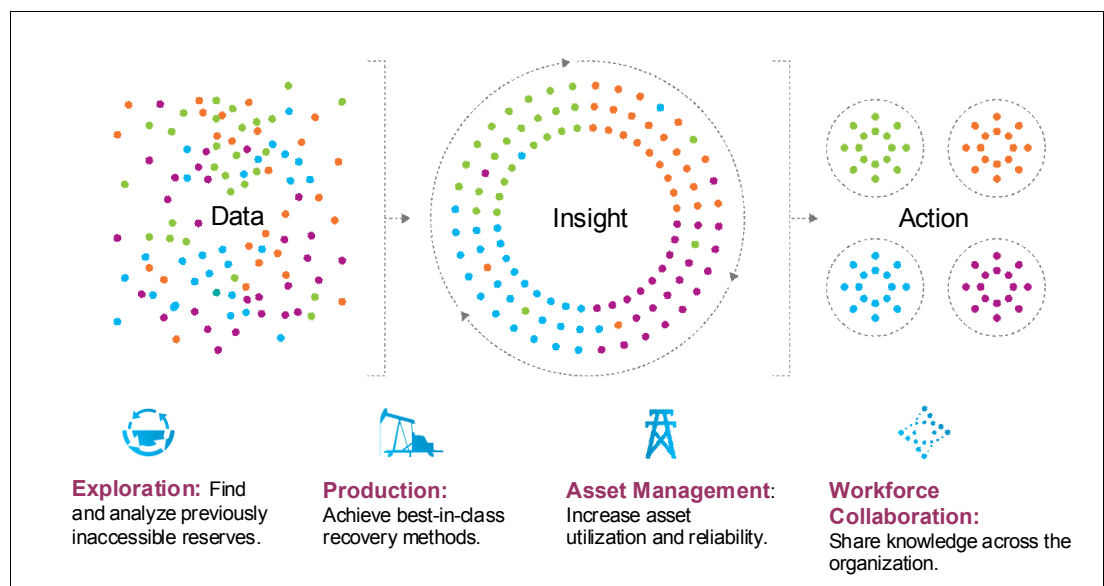


Figure 10-1 Trends in the oil and gas industry

10.1.1 Enhance exploration and production

A smarter computing approach to exploration and production results in four focus areas that match the needs that are described in Table 10-1.

Table 10-1 Focus areas to address oil and gas exploration and production needs

Focus area	Need	Purpose
User access	► Clients for 2D/3D remote engineering desktops, standard browsers	Supporting anytime, anywhere, collaboration work
Process management of control systems	► General computing on consolidated, virtualized, expert integrated systems	Greater agility, scalability, efficiency, and availability at reduced cost
Numerical analysis and 3D visualization	► Technical computing clusters and clouds, with support for accelerators and coprocessors ► Strong workload management	Improved time to market by using improved analytical and operational insight
Global file systems and scalable storage (block and file)	► Shared, centralized, tiered storage ► Active file management between strategic data center locations	Efficient content movement, management, and worldwide access to valid data sources

Technical computing clouds target these specific focus areas, and address some of the critical needs of the clients in the oil and gas industry.

10.1.2 Workloads

The upstream oil and gas industry focuses on the discovery and harvesting of reservoirs of petroleum around the world.

A component of their stock price is in their reserves. In other words, the amount of petroleum that they have discovered and that has a certain revenue potential.

Every year oil companies sell a certain number of millions of barrels of oil. These companies hope to find and explore reserves in excess of the current production to show that their business will grow in the future. There is a lot of pressure on reserve replacement because oil reserves are getting harder to find.

The data coming from oil exploration is exploding, doubling every year in terms of the data footprint. Furthermore, this trend is expected to go up dramatically.

Converting field data to 3D images

The largest application area is *seismic imaging*, which is the development of 3D pictures of the subsurface of the earth so that geophysics can determine the presence of oil or gas reservoirs. The next step in the process is called *reservoir simulation*, which simulates the recovery of oil from that reservoir. This is a commercial exercise to determine that the reservoir is commercially viable in terms of the kind of expenses that people must make to recover it.

Seismic imaging

This is an acoustic based process where, in the case of marine-based or ocean-based seismic imaging, large vessels use an acoustic gun to fire sound waves through the surface of the earth. The waves are reflected back up, captured, and the associated acoustic data is then run through software algorithms that convert the time base acoustic image into a 3D image of the subsurface.

This process uses traditional Fast Fourier Transform (FFT) algorithms that were invented in the 1800s. Many of these applications were envisioned in the 1930s and have been improved in terms of better resolution of the layers of the earth. The algorithm is extremely parallel, and can achieve much better performance running on large high-performance computing (HPC) cluster with little internode messaging requirements.

The resolution quality has always depended on the speed of computers available to run it. But as computers get faster, the quality of the imaging can be better and better by using more complex algorithms.

In fact, these algorithms have already been envisioned through all the way to what is called direct inversion. Today companies typically use an algorithm called *Reverse Time Migration (RTM)*, which is fairly complex. To run adequate volumes requires the use of large computing facilities.

Note: *Reverse Time Migration (RTM)* is the most common high end application that is used in upstream petroleum. It is characterized by modeling both the downward and upward acoustic waves as they travel down through the layers of the earth and are reflected back by the various layers.

RTM might be the answer to many of today's problems, but researchers must also run a larger list of applications in the future to run imaging with better resolution than today.

The following is a list of applications that oil companies research departments use:

- ▶ Full waveform inversion
- ▶ 3D internal multiples attenuation
- ▶ Iso/VTI FWI
- ▶ Integrated imaging/modeling
- ▶ TTI FWI
- ▶ Real-time 3D RTM
- ▶ Viscoelastic FWI
- ▶ Inverse scattering series
- ▶ Direct inversion (Real FWI)

Also, the research divisions of oil companies have started application development road maps that will result in exascale projects by the end of the decade. So on the seismic imaging side, the road map is going to require processing power up a thousand times greater than used today.

Reservoir simulation

The economic and technical model of a reservoir are fundamental for determining drill decisions and planning. The algorithms that are used are less parallel, and usually demand large amounts of memory per node. Large shared memory systems and high-bandwidth interconnect such as InfiniBand are required to deliver the best results.

On the reservoir simulation side, growth is driven by the need to create simulations with a greater resolution, but also rerunning these simulations many times in a Monte Carlo type approach to provide a better analytical approach.

10.1.3 Application software

Seismic imaging is very different from reservoir simulation because many oil companies develop their own seismic imaging products in-house because they feel they can achieve a significant competitive advantage by doing so. Companies with larger development budgets have research departments that develop new seismic imaging algorithms and implement them into software that they use to develop their own seismic imaging processes and products.

On the reservoir simulation side, companies tend to accept independent software vendor (ISV) software more readily. There is also ISV software for seismic imaging, but the ISV software adoption is much more widespread on the reservoir simulation side.

Table 10-2 shows the leading vendors in the upstream oil and gas industry. Schlumberger WesternGeco does seismic imaging, and Schlumberger Information Solutions provides reservoir simulation and desktop software for geophysics and geologists. Similarly, Halliburton Landmark Graphics is a full range software vendor, and Computer Modeling Group is predominately a reservoir simulation provider. Paradigm Geophysics is predominately a seismic imaging provider.

Table 10-2 Independent software vendors and applications for seismic imaging

ISV	Applications
Schlumberger WesternGeco	▶ OMEGA family of seismic processing software (200 programs)
	▶ RTM ported to GPGPUs
Schlumberger Information Solutions	▶ ECLIPSE Reservoir Simulation Software
	▶ Petrell
	▶ Intersect
Halliburton Landmark Graphics	▶ SeisSpace
	▶ ProMax
	▶ VIP
Computer Modeling Group (CMG)	▶ STARS Heavy Oil Simulator
Paradigm GeoPhysics	▶ EPOS IV Seismic Imaging

10.2 Architecture

There is much interest in employing the cloud model (in technical computing) to the oil and gas industry. However, little cloud implementation has occurred thus far because these systems tend to be so massive that current commercial cloud offerings do not have the capacity for their computing requirements.

Figure 10-2 shows that petroleum exploration pipeline is quite complex and poses some considerable challenges to end-to-end cloud implementation.

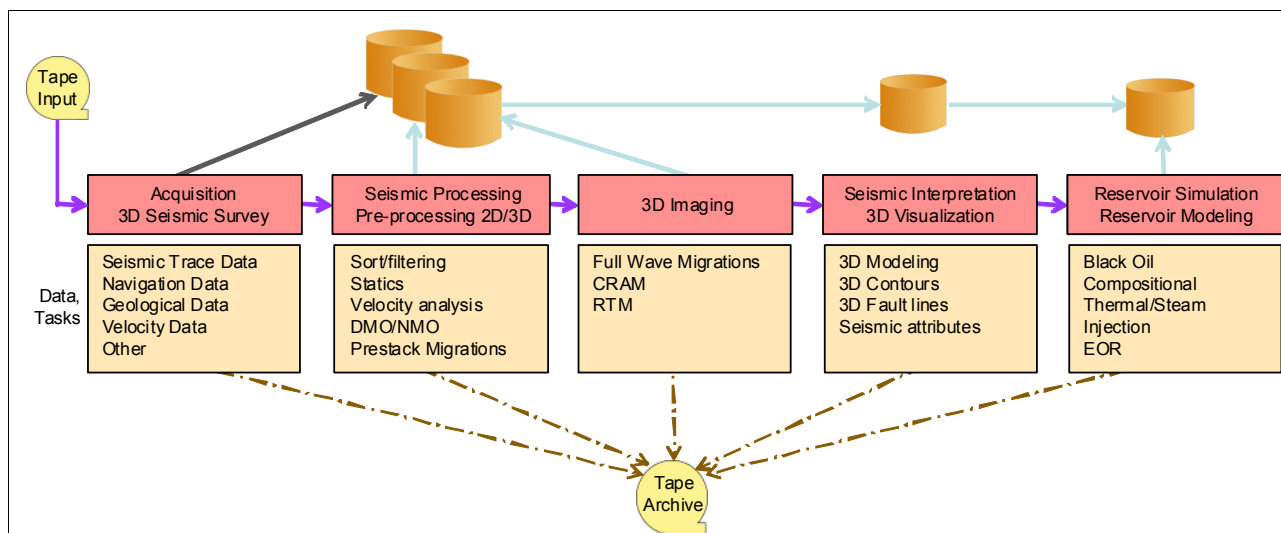


Figure 10-2 Petroleum exploration and production processing and data lifecycle

However, there are considerable benefits in applying a private cloud model to some steps of the process, specially in the area of remote 3D visualization and collaboration. Most of the visualization software and tools used in both seismic imaging and reservoir simulation can use the 3D desktop virtualization architecture described in Chapter 7, “Solution for engineering workloads” on page 139.

10.2.1 Components

This section describes the solution components.

IBM Platform Application Center (PAC) remote visualization

IBM Platform Application Center Standard Edition provides not only basic job submission, and job and host monitoring, but also default application templates, role-based access control, reporting, customization, and remote visualization capabilities.

PAC can help reduce application license cost by increasing license utilization. With fine-tuned licensing scheduling policies, PAC and licensing scheduler can help companies optimize use of expensive reservoir simulation software licenses.

Application templates are used to integrate applications. You can use the built-in templates to immediately submit jobs to the specific applications.

Note: Generally, the name of the template indicates the name of the application to which jobs can be submitted.

Tested applications

PAC provides some built-in application templates for oil and gas industry applications. Table 10-3 lists the versions of applications that have been tested with Platform Application Center.

Table 10-3 Tested applications for the oil and gas industry

Applications	Tested versions
CMGL_GEM	▶ 2008.12 ▶ 2009.13
CMGL_IMEX	▶ 2008.11 ▶ 2009.11
CMGL_STARS	▶ 2008.12 ▶ 2009.11
ECLIPSE	▶ 2009.1 ▶ 2010
STAR-CCM+	▶ 6.02

Note: These are tested application versions. Job submission forms can be customized to support other versions.

Submission forms and submission scripts

Application templates have two components:

- Submission form** The job submission form is composed of fields. Each field has a unique ID associated with it. IDs must be unique within the same form.
- Submission script** The job submission script uses the same IDs as the submission form to pass values to your application.

Customizing application templates

You can customize application templates by adding or removing fields, rearranging fields, and entering default values for fields. You can also change field names and add help text for fields. Figure 10-3 shows the submission form editing window for the ECLIPSE built-in template.

In addition, you can create hidden fields, which are fields that are only visible to you, the administrator. These can hold default values for the submission forms. Users cannot see hidden fields in their forms.

The screenshot shows the 'ECLIPSE' submission form editing window in the IBM Platform Application Center 9.1. The window has a title bar with 'IBM Platform Application Center 9.1', 'Isfadmin', and buttons for 'Log Out', 'Help', and 'Refresh'. Below the title bar, the window is titled 'ECLIPSE' and shows 'Template Name: ECLIPSE', 'Type: Built-in', and 'Application: ECLIPSE'. There are three tabs: 'Submission Form' (selected), 'Submission Script', and 'Help Documentation'. On the left side, there is a vertical navigation bar with 'Jobs', 'Resources', 'Settings', and 'Reports'. The main content area is divided into three sections: 'Application Parameters', 'Cluster Parameters', and 'Application Data Files'. Each section has 'Add', 'Delete', and 'Edit' icons. The 'Application Parameters' section contains: 'Simulator *' with a dropdown menu showing 'E100' and ID '[ID:SIMULATOR]'; 'Version' with a dropdown menu showing '2010.2' and ID '[ID:RELEASE]'; 'MPI' with a dropdown menu showing 'PMPI' and ID '[ID:MPI]'; 'Use local configuration *' with a dropdown menu showing 'LOCAL_NO' and ID '[ID:LOCAL]'; and 'Additional Parameters' with a text input field and ID '[ID:OTHER_OPTIONS]'. The 'Cluster Parameters' section contains: 'Queue' with a dropdown menu showing 'normal' and ID '[ID:QUEUE]'. The 'Application Data Files' section is currently empty.

Figure 10-3 Application template for ECLIPSE

Figure 10-4 shows the submission script editing window for the CFX built-in template in PAC.

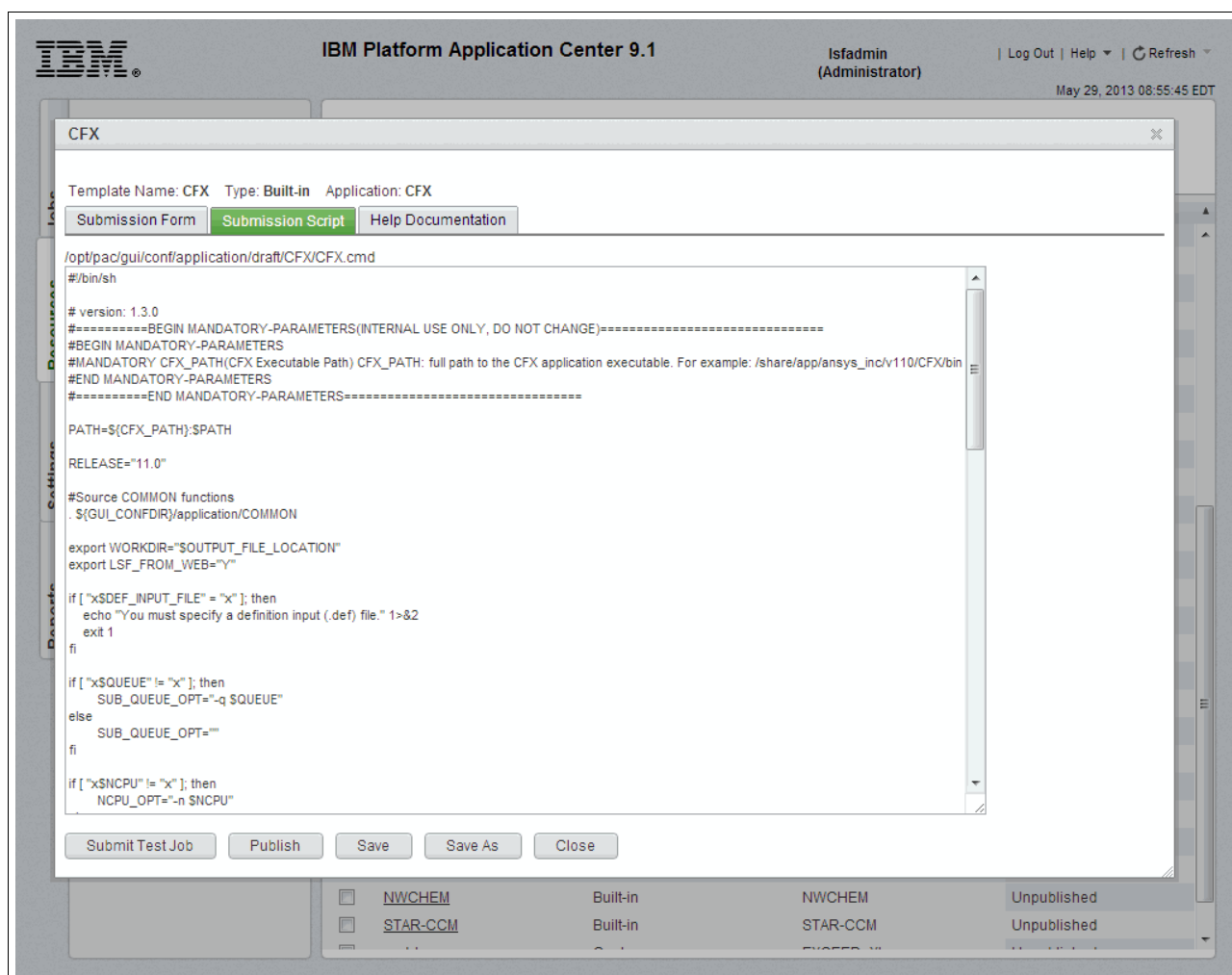


Figure 10-4 CFX submission script editing window

Note: For more information about PAC application templates configuration and remote visualization setup, see *Administering Platform Application Center*, SC22-5396-01.



Solution for business analytics workloads

This chapter provides an architecture reference for business analytics clusters to be deployed in a technical computing cloud. The solution uses IBM InfoSphere BigInsights as the environment for the cluster.

This chapter includes the following sections:

- ▶ IBM InfoSphere BigInsights advantages for business analytics
- ▶ Deploying a BigInsights environment within a PCM-AE managed cloud
- ▶ The concepts behind NoSQL databases

11.1 MapReduce

MapReduce defined a methodology that enables the analysis of large amounts of data by the use of parallel computing. This method is used by most BigData applications. It consists basically of having each node analyze the amount of data that is local to it to avoid the transfer of data among nodes. This is referred to as the “map” phase of the analysis. Then, the results of each node are checked against each other to drop duplicate results, as data chunks processed on different nodes might have the same value. This is referred to as the “reduce” phase of the methodology.

The spreading of data analysis throughout the nodes of a grid without the need for data transfer, then the verification of a much smaller set of data to create a final result, allows MapReduce to provide answers to BigData analysis much faster. For more information about MapReduce, see Chapter 4, “IBM Platform Symphony MapReduce” on page 59.

Applications currently exist that use the MapReduce paradigm to analyze lots of unstructured data. IBM InfoSphere BigInsights is one of them.

11.1.1 IBM InfoSphere BigInsights

In a world that is heading towards an increasing amount of data generated at a fast rate every minute, technologies to analyze large volumes of data of varied types are being engaged. These include the MapReduce paradigm explained in the previous section. Today, frameworks such as Apache’s Hadoop use MapReduce to extract meaningful information from tons of unstructured data.

IBM InfoSphere BigInsights is a software platform that is based on the Hadoop architecture. It is the IBM solution for companies wanting to analyze their big data. The combination of IBM-developed technologies and Hadoop, packaged in an integrated fashion, provide an easy-to-install solution that is enterprise ready. Other technology components such as Derby, Hive, Hbase, and Pig are also packaged within IBM InfoSphere BigInsights.

The following are the benefits of an IBM InfoSphere BigInsights solution for your business or big data environment:

- Easy, integrated installation

The installation of IBM InfoSphere BigInsights is performed through a graphical user interface (GUI), and does not require any special skills. A check is run at the end of installation to ensure the correct deployment of the solution components. All of the integrated components have been exhaustively tested to ensure compatibility with the platform. You have support for a multi-node installation approach, thus simplifying the task of creating a large IBM InfoSphere BigInsights cluster.

- Compatible with other data analysis solutions

IBM InfoSphere BigInsights can be used with existing infrastructure and solutions for data analysis such as the IBM PureData System for Analytics (Netezza family) of data warehouse appliances, IBM Smart Analytics Systems, and IBM InfoSphere DataStage for ETL jobs.

Also, a Java Database Connectivity (JDBC) connector allows you to integrate it with other database systems such as Oracle, Microsoft SQL Server, MySQL, and Teradata.

- Enterprise class support

Enterprise class support means that you get assistance for your BigData analytics environment when you need it. There are two types of support, depending on the edition type acquired for IBM InfoSphere BigInsights: Enterprise and basic editions.

The enterprise edition provides a 24-hour support service and uses worldwide knowledge. The basic edition allows you to get the software for no extra fee for data environments up to 10 TB and still get access to online support.

- Enterprise class functionality

Businesses and research entities need highly available systems. This is why IBM InfoSphere BigInsights can be deployed on top of hardware that helps eliminate any single points of failure such as IBM servers.

Also, it provides you interfaces to manage and visualize jobs that are submitted to the cluster environment and to perform other administration tasks such as user management, authority levels, and content views.

- BigSheets

A browser-based analytic tool that enables business users and users with no programming knowledge to explore and analyze data in the distributed file system. Its interface is presented in a spreadsheet format so that you can model, filter, combine, and create charts in a fashion you are already familiar with. The resulting work can be exported to various formats such as HTML, CSV, RSS, Jason, and Atom.

- Text analytics

IBM InfoSphere BigInsights allows you to work with unstructured text data. You can store your data as it is acquired, and use this BigInsights component to directly analyze it without having to spend time preprocessing your text data.

- Workflow scheduling

IBM InfoSphere BigInsights can work with its own job scheduler for running MapReduce jobs. This brings advantages over Hadoop's Fair scheduler that works by providing equal processing shares to jobs. It allows you, for example, to prioritize some jobs over others or ensure that small jobs are run faster (users typically expect smaller jobs to finish quickly as they hope to use the results right away).

In addition, IBM InfoSphere BigInsights can be integrated with IBM Platform Symphony to control job scheduling. Platform Symphony brings a more efficient job management to the BigInsights solution. It is able to accelerate parallel applications, resulting in faster results and better utilization of the cluster, even under dynamically changing workloads. Also, Platform Symphony is able to scale to very large cluster configurations that reach up to thousands of processor cores.

For more information about IBM InfoSphere BigInsights features, components, and integration with other software, see *Implementing IBM InfoSphere BigInsights on System x*, SG24-8077, and *Integration of IBM Platform Symphony and IBM InfoSphere BigInsights*, REDP-5006.

11.1.2 Deploying a BigInsights workload inside a cloud

Customers that have diverse technical computing workloads can use the IBM technical computing clouds technology to quickly deploy a data analytics cluster. This can be done in a simple and user-oriented manner.

This section provides information about the hardware architecture and components, and also the software architecture and components used to run a quick BigInsights data analysis. The

basic foundations shown here can be used to deploy either a permanent or temporary BigInsights cluster with multi-user support. In the example scenario, the BigInsights cluster is limited to a single user, but its definition can be suited for a multi-user environment.

The following sections address the hardware and software layers that are used to build up the environment, how you interact with this architecture to create a BigInsights cluster, and a demonstration on how to access and use the created cluster. These references do not constrain how you can design your solution. For more information and other architecture references, see *Implementing IBM InfoSphere BigInsights on System x*, SG24-8077.

Hardware architecture

This section provides a description of a hardware architecture reference for deploying a BigInsights cluster inside of a Platform Cluster Manager - Advanced Edition (PCM-AE) managed cloud. This architecture provides you with the benefits and flexibility of dynamically creating a BigInsights cluster. This cluster can be later expanded or reduced based on workload demand, or even destroyed in the case of running temporary workloads.

Figure 11-1 illustrates how the hardware components were set up for this use case.

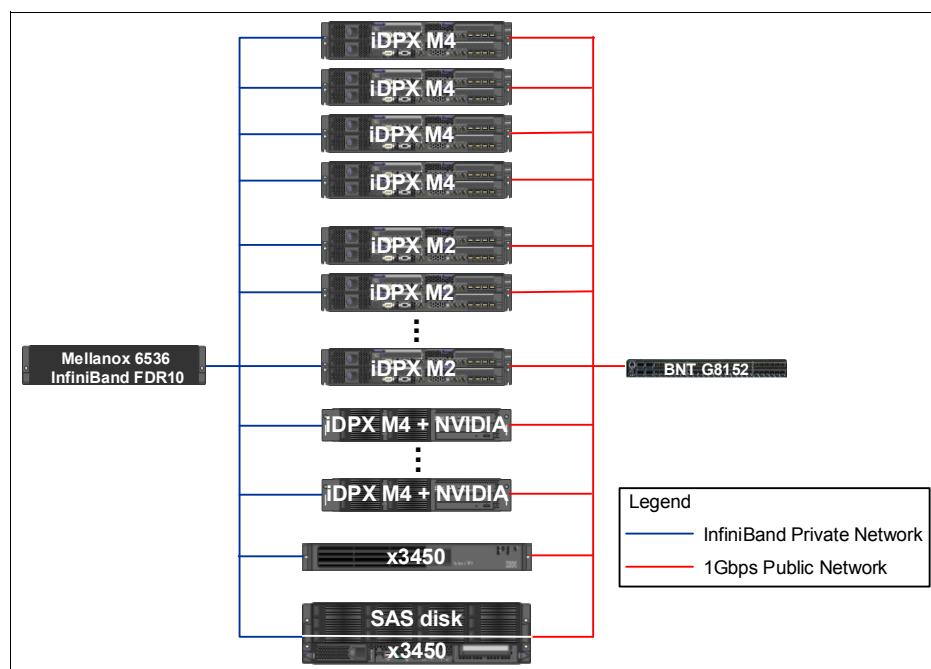


Figure 11-1 Lab hardware setup: PCM-AE environment to deploy other cloud solutions

The InfiniBand network serves as a high-speed connection between the nodes. The 1 Gbps network serves as a public gateway network for users to access the PCM-AE environment and the clouds within it.

The following is the hardware used for this use case:

- ▶ 8 iDataPlex M4 servers
- ▶ 13 iDataPlex M2 servers
- ▶ 4 iDataPlex servers with NVIDIA Quadro 5000 adapters
- ▶ 2 x3450 servers
- ▶ 1 Mellanox 6536 InfiniBand FDR10 switch
- ▶ 1 IBM BNT® G8152 Gigabit Ethernet switch
- ▶ 2 TB of shared storage for the IBM General Parallel File System (GPFS) (SAS disks)

This infrastructure was put together to create the PCM-AE managed cloud. Multiple high-performance computing (HPC) environments are running concurrently in this example scenario. One of the iDataPlex servers hosts the PCM-AE management server, one x3450 hosts the xCAT management node, and the other x3450 handles the 2 TB SAS disk storage area.

For this BigInsights use case, use two of the physical iDataPlex servers to host the master and compute nodes as explained in “Deploying a BigInsights cluster and running BigInsights workloads” on page 238.

Software architecture

This section provides a description of the software components architecture used to run the example r BigInsights use case scenario. Although there are multiple possible architectures to integrate all of the software pieces depending on the user’s needs, only the one used as an illustration is described.

Figure 11-2 depicts the software components of the example cloud environment.

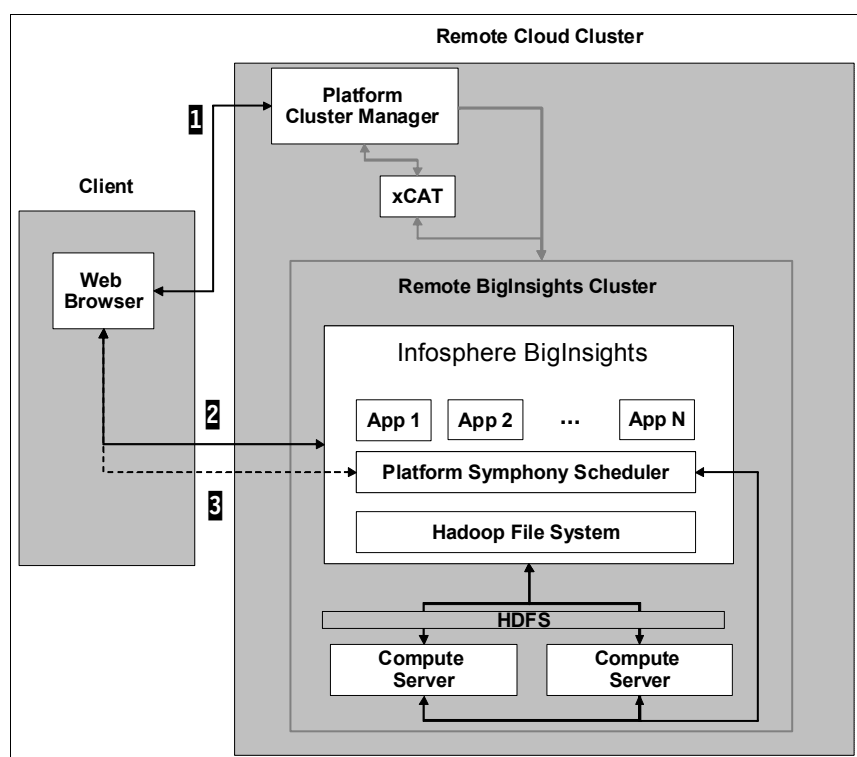


Figure 11-2 IBM InfoSphere BigInsights software components of the use case

IBM Platform Cluster Manager - Advanced Edition (PCM-AE) is used to orchestrate the creation and management of the BigInsights cluster. In the example scenario, the BigInsights servers (master and compute nodes) are physical machines. PCM-AE uses xCAT to deploy physical machines.

The BigInsights cluster is composed of the product itself, the Hadoop file system underneath it, and the analytics applications that are deployed inside BigInsights. However, Hadoop’s Fair scheduler is replaced with Platform Symphony.

Notice that in Figure 11-2 on page 237, the user interacts with the environment through an HTTP browser connection at two entry points, and optionally a third entry point:

- The PCM-AE environment

Users connect to PCM-AE to create the IBM InfoSphere BigInsights cluster, size it, and optionally resize it according to workload demands. This entry point is at number **1** in Figure 11-2 on page 237.

- The IBM InfoSphere BigInsights environment

After a cluster for BigInsights is active in the PCM-AE cloud environment, you can connect to it directly through the public network as explained in “Hardware architecture” on page 236. You can start analytics applications hosted within BigInsights. This entry point is at **2** in Figure 11-2 on page 237.

- The Platform Symphony environment

Optionally you can access Platform Symphony directly to check its configurations or use any of its report capabilities. This entry point is at **3** in Figure 11-2 on page 237.

Tip: Platform Symphony’s services run on port 18080, and can be accessed at address `http://<ip>:18080/platform`.

Deploying a BigInsights cluster and running BigInsights workloads

This section describes the process of deploying a BigInsights cluster from within PCM-AE and running bog data analysis on the provisioned cluster.

Log in to the PCM-AE web portal by pointing your browser to port 8080 on the management node. After logging in, click **Clusters** → **Cockpit** area as shown in Figure 11-3.

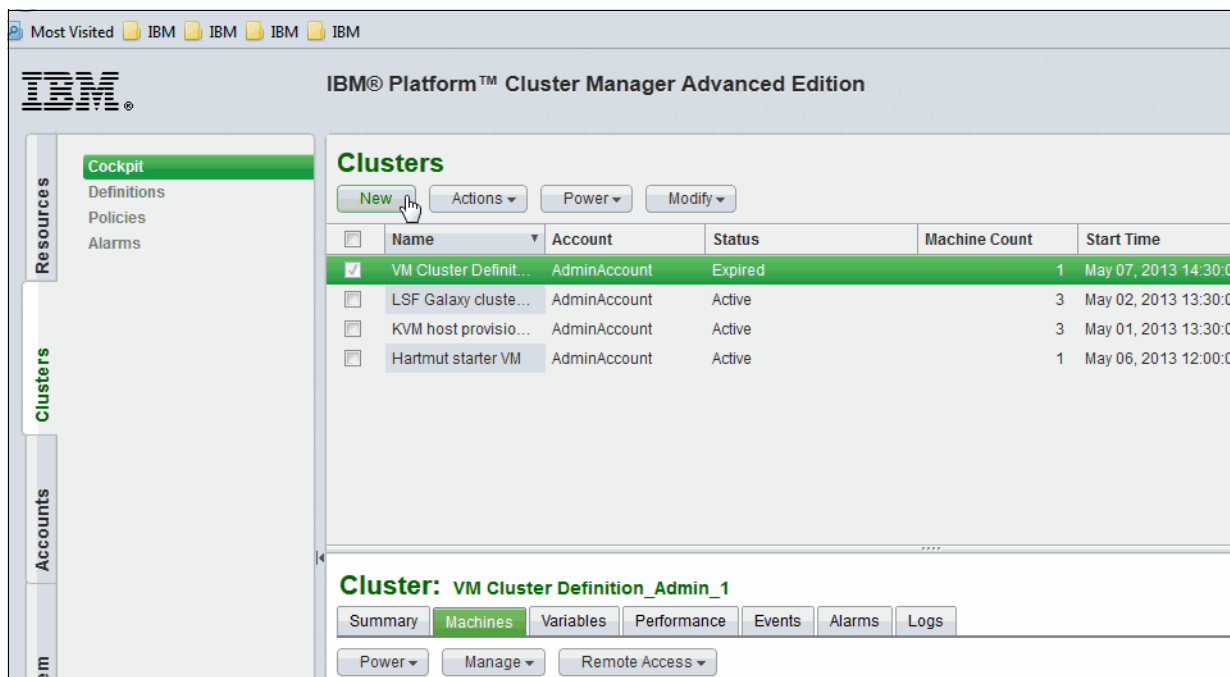


Figure 11-3 PCM-AE: Clusters tab (left side tab menu), cockpit area

To deploy a cluster, your PCM-AE environment needs to contain a cluster definition for the type of workload you want to deploy. A cluster definition holds information about operating

system and basic network configuration. It provides the ability to install extra software on the top of a base environment by using postscripts.

From an user point of view, after the PCM-AE administrator publishes the cluster definition for use, they just have to follow a guided wizard to create a cluster. In essence, the administrators have the knowledge to define the clusters whereas a user, simply has to know how to go through the simple creation wizard. Figure 11-4 shows the cluster definition of the test environment. It is based on the Red Hat Enterprise Linux operating system version 6.2, IBM InfoSphere BigInsights 2.0, and Platform Symphony 6.1. For more information about creating cluster definitions in PCM-AE, see *IBM Platform Computing Solutions*, SG24-8073.

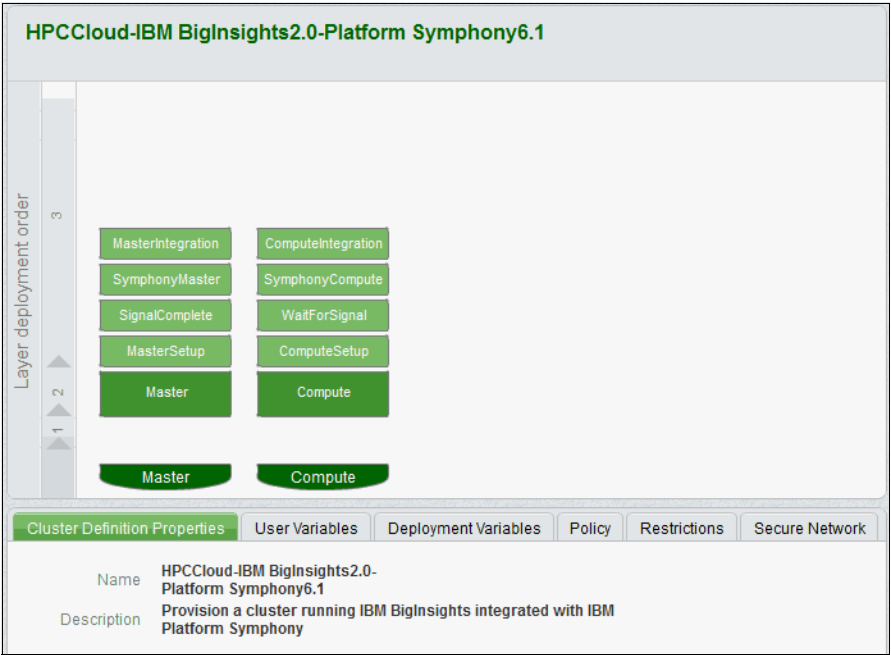


Figure 11-4 Cluster definition inside PCM-AE: Master and subordinate nodes

Click **New** as depicted in Figure 11-3 on page 238, then choose the appropriate cluster definition for the scenario as shown in Figure 11-5. Click **Instantiate**.

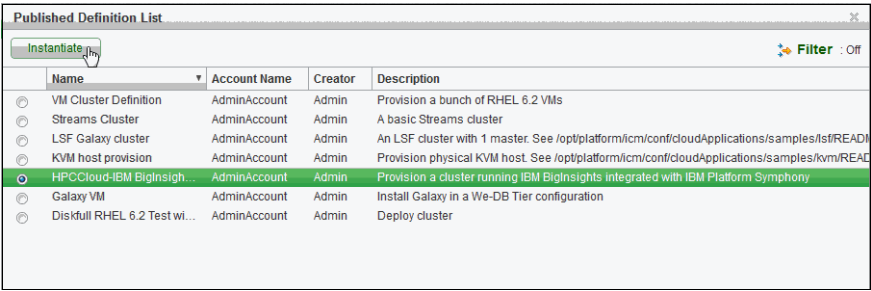


Figure 11-5 Instantiating a BigInsights and Platform Symphony cluster

The next wizard step is the definition of processor and memory parameters for the master and compute nodes, and also how many compute nodes to include in the cluster. In the example, just one compute node is defined as shown in Figure 11-6, plus the mandatory master node to create a non-expiring cluster.

Create Cluster

Cluster Properties

Name: HPCCloud-IBM BigInsights2.0-Platform
Description: Technical Computing Clouds Redbook
Definition Version: 2.0

Cluster Run Time

Start Time: May 10, 2013 17:00 EDT
End Time: May 11, 2013 17:00 EDT (No Expiry)
Recurrence Pattern: Only once

Machines | User Variables | Policy

Tier Name	Number of Machines	Number of CPU per Machine	Memory (MB)	Resource Selection Criteria
Master	1	1	512 (512 - 1024)	---
Compute	1	2	1024 (512 - 1024)	---

Create Cancel

Figure 11-6 Cluster creation wizard in PCM-AE: Resource definition

After you click **Create**, the cluster status is displayed as *Active (Provisioning)* on the **Clusters** → **Cockpit** interface of PCM-AE. Figure 11-7 shows the cluster in the provisioning state.

IBM Platform Cluster Manager Advanced Edition

Admin (Administrator) | Log Out | Help | Refresh | May 10, 2013 17:33:03 EDT

Clusters

New Actions Power Modify Filter: Off Options

Name	Account	Status	Machine Count	Start Time	End Time	Owner
VM Cluster Definiti...	AdminAccount	Expired	1	May 07, 2013 14:30:00 EDT	May 08, 2013 14:30:00 EDT	Admin
LSF Galaxy cluster...	AdminAccount	Active	3	May 02, 2013 13:30:00 EDT	No Expiry	Admin
KVM host provision...	AdminAccount	Active	3	May 01, 2013 13:30:00 EDT	No Expiry	Admin
Hartmut starter VM	AdminAccount	Active	1	May 06, 2013 12:00:00 EDT	No Expiry	Admin
HPCCloud-IBM Big...	AdminAccount	Active (Provisioning)	2	May 10, 2013 17:00:00 EDT	No Expiry	Admin

Cluster: HPCCloud-IBM BigInsights2.0-Platform Symphony6.1_A...

Summary Machines Variables Performance Events Alarms Logs

Power Manage Remote Access Filter: Off Options

Alarm	Name	Host Name	Physical Host Name	Source	IP Address	Status	CPU (%)	Memory (%)	Cluster Tier	Account
<input checked="" type="checkbox"/>	c445f3an27.c...	c445f3an27.d...	c445f3an27.cluster...	xCAT		Provisioning	0	0	Compute	AdminAc
<input type="checkbox"/>	c445f3an03.c...	c445f3an03.d...	c445f3an03.cluster...	xCAT		Provisioning	0	0	Master	AdminAc

Figure 11-7 PCM-AE: cluster provisioning

After the process is complete, you can check the IP address and host name of the master and compute nodes for accessing the IBM InfoSphere BigInsights cluster. This information is available in the cluster cockpit interface as shown Figure 11-7.

To verify the newly deployed BigInsights cluster, deploy and run the simple word count application. Access BigInsights user interface by pointing your web browser to the IP address of the master tier node on the deployed cluster using port 8080. Then, click the Applications tab and deploy the Word Count application as shown in Figure 11-8.

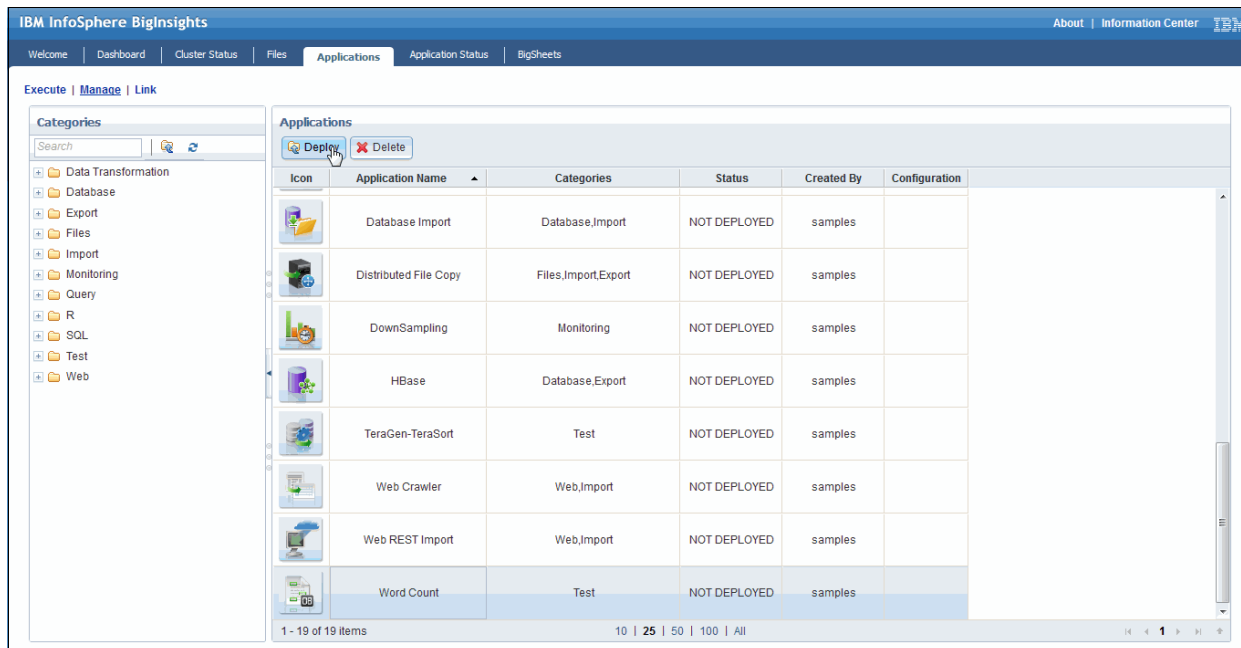


Figure 11-8 Deploying the word count application for use within IBM InfoSphere BigInsights

BigInsights users can now create Word Count jobs by clicking the Welcome tab and clicking **Run an application** as shown in Figure 11-9.



Figure 11-9 Running applications in IBM InfoSphere BigInsights

Figure 11-10 shows a simple input and output directory setup created for running this test case. The input directory contains a text file of an IBM publication.

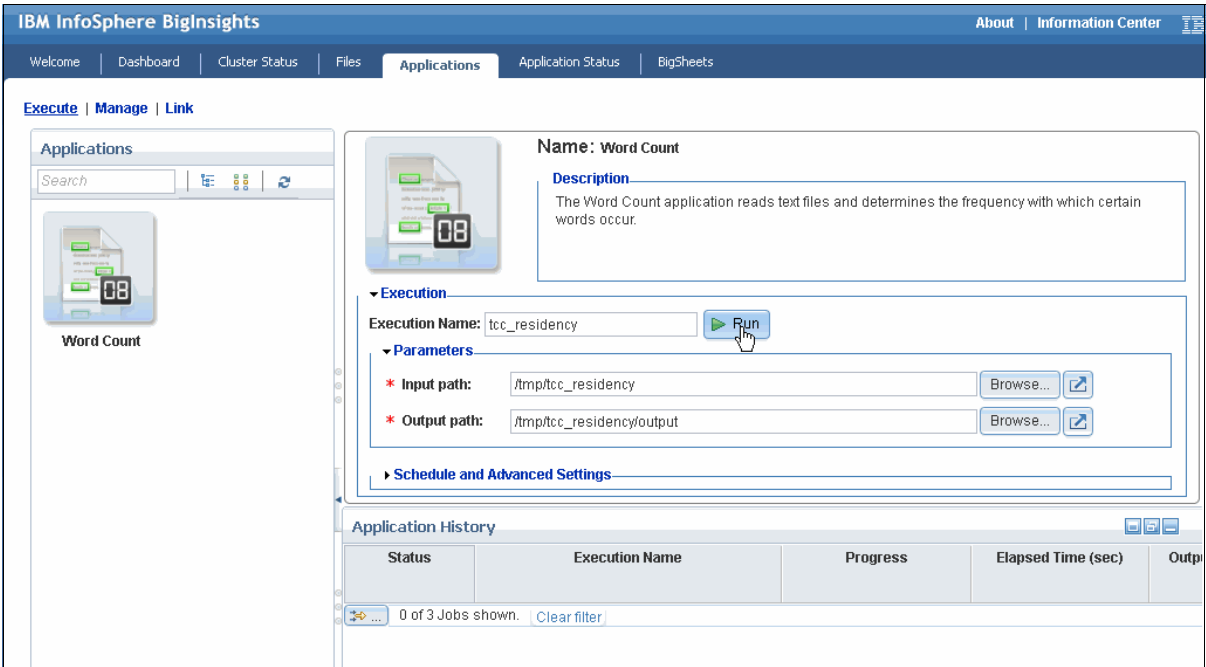


Figure 11-10 Running a word count job in IBM InfoSphere BigInsights

After the job is finished, check its output for the results as depicted in Figure 11-11.

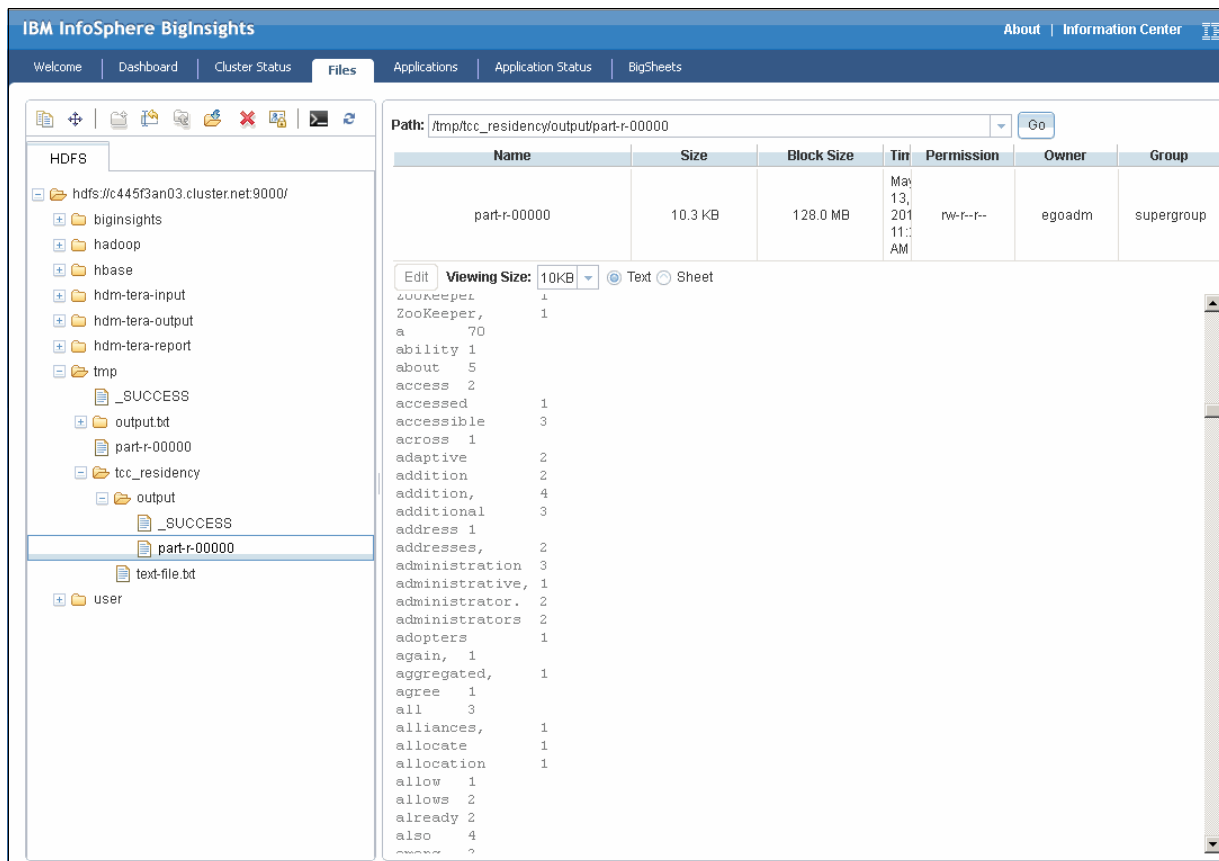


Figure 11-11 BigInsights results of counting the words of a text file

Essentially, this use case demonstrates the simplicity of running an analytics workload inside of a cloud environment. Notice that no programming skills are required of the user, and the deployment of the computing cloud is straightforward after a working cluster definition exists within PCM-AE.

11.2 NoSQL

The current trend is for more people gaining access to the internet and using services such as blog posting, personal web pages, and social media. This has created an explosion of data that needs to be handled. As a consequence, companies that provide these services need to be able to scale in terms of data handling.

Now, imagine that all of the above happen continuously. It is not uncommon to read statements that mention that most of today's data have been created in the past two years or so. How can the service providers keep up with this pace, and gain business advantages over their competitors? Part of the answer to that lies in the mix of Cloud Computing and BigData.

Computer grids have been commonly used to solve BigData problems. Distributed computing (grids) are the standard infrastructure that is used to solve these problems. They use cheap hardware, apply massive virtualization, and use modern technologies that are able to scale and process at the rate that new data is generated. Clouds are an excellent choice to host these grid environments because cloud characteristics makes them a good fit for this

scenario. This is because clouds offer flexible scalability (grow, shrink), self-service, automated and quick provisioning, and multi-tenancy.

To address today's needs for being able to analyze more data, including unstructured data, researchers have proposed models that work in a different manner than a standard relational database management system (RDBMS).

Relational databases are based on ensuring that transactions are processed reliably. They rely on the atomicity, consistency, isolation, and durability (ACID) properties of a single transaction. However, these databases can face great challenges when it comes to analyzing huge amounts of, for example, unstructured BigData data.

As a solution to this, and following the trend of distributed computing that the cloud provides, a new paradigm of databases is proposed: NoSQL, recently referred to as *Not only SQL* databases. NoSQL is based on the concepts of low cost and performance. The following is a list of its characteristics:

- ▶ Horizontal data scaling
- ▶ Support for weaker consistency models
- ▶ Support for flexible schemas and data models
- ▶ Able to use simple, low-level query interfaces

As opposed to RDMS databases that rely on the ACID concepts of a transaction, NoSQL databases rely on the basic availability, soft-state, eventual consistency (BASE) paradigm.

BASE databases do not provide the full fault-tolerance that an ACID database does. However, it is being proven to be suitable for use within large grids of computers that are analyzing data. If a node fails, the whole grid system does not come down. Just the amount of data that was accessible through that node becomes unavailable.

The eventual consistency characteristic means that changes are propagated to all nodes if enough time has passed. In an environment where data is not updated often, this is an acceptable approach. This weaker consistency model results in higher performance of data processing. Some businesses, such as e-commerce, prefer to prioritize high performance to be able to attend thousands or millions of customers with less processing delay and eventually deal with an inconsistency than to ensure a full data consistency that requires delays in merchandise purchase processes.

The soft-state characteristic is related to the eventual consistency of a data. Because eventual consistency relies on the statement that data is probably consistent when enough time has passed, inconsistencies might occur during that time. This is then handled by the application rather than by the database. In a real world scenario, this means that an e-commerce site that sold the last inventory item of a product to two customers might need to cancel one of them and offer that customer some kind of trade-off in return. They might determine that this as a more profitable approach over slowing down sales to enforce data consistency.

NoSQL databases can be based on different data models to manage data:

- ▶ Key-value pairs
- ▶ Row storage
- ▶ Graph oriented
- ▶ Document oriented

Multiple solutions today are based on the concepts presented here. Hadoop and HBase are open source examples. IBM offers support for NoSQL within DB2 as well.

11.2.1 HBase

HBase is an example of a database implementation that follows the NoSQL concepts. It is included in Apache's Hadoop and its development is supported by IBM.

HBase is a column-oriented database that runs on top of data that are stored on HDFS. As such, the complexity of handling with distributed computing is abstracted from the database itself. As the data is organized in columns, a group of columns forms a row. Next, a set of rows form a table. Data is indexed by a row key, column key, and time stamp. The keys map to a value, which is an uninterpreted array of bytes.

To provide more performance and allow the manipulation of tons of data, HBase data is not updated in place. Updating occurs by adding a data entry with a different time stamp. This follows the "Eventual consistency" characteristic of the BASE paradigm mentioned for NoSQL databases.

The following is a list of characteristics that make HBase useful for business analytics:

- ▶ Supported in IBM InfoSphere BigInsights
 - Enables users to use MapReduce algorithms of BigInsights
- ▶ Lower cost compared to other RDBMS databases
- ▶ Is able to scale up to the processing of very large data sets (terabytes, petabytes)
- ▶ Supports flexible data models of sparse records
- ▶ Supports random access and read/write support for Hadoop applications
- ▶ Automatic sharing without the corresponding penalties of an RDBMS database

Table 11-1 compares some aspects of HBase with other RDBMS databases.

Table 11-1 Comparison of HBase and RDBMS databases

Characteristic	HBase	RDBMS databases
Data layout	Column Family-oriented	Row or column-oriented
Transactions	Single row only	Yes
Query language	get/put/scan	SQL
Security	Authentication/ACL	Authentication/Authorization
Indexes	Row/Column/Timestamp only	Yes
Maximum data size	Petabytes and up	Terabytes
Read/write throughput limits	Millions of queries per second	Thousands of queries per second

An HBase implementation is characterized by a layout of a few interconnected components as shown in Figure 11-12.

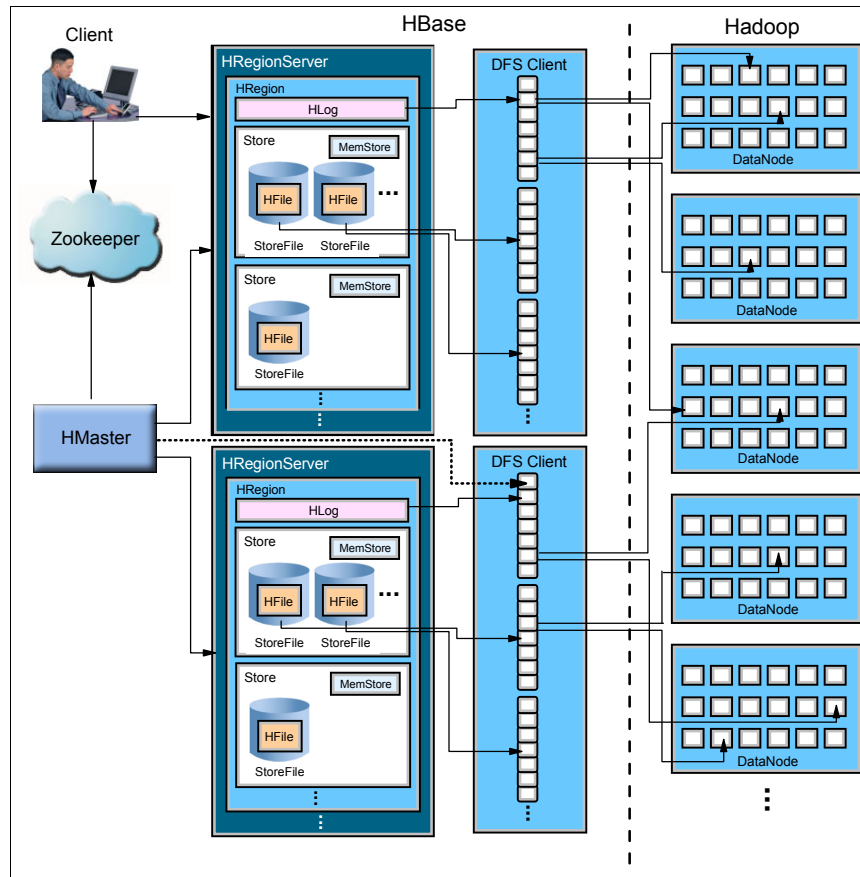


Figure 11-12 HBase component architecture

The architecture shown in Figure 11-12 is composed of these components:

Region	A subset of table rows. Automatically shared upon growth.
Region servers	Host tables. Run read operations and buffered write operations. Clients talk to region servers to have access to data.
Master	Coordinates the region servers, detects their status, and runs load balance among them. It also assigns regions to region servers. Multiple master servers are supported starting with IBM InfoSphere BigInsights 1.4 (active master, one or more passive master backups).
Zookeeper	Part of the Hadoop system. Ensures that the master server is running, provides bootstrap locations for regions, registers region servers, handles region and master server failures, and provides fault tolerance to the architecture.
Hadoop data nodes	Nodes that store data using the Hadoop file system. Communication with region servers happens through a distributed file system (DFS) client.

For more information about HBase, see the following publications:

- ▶ <http://hbase.apache.org>
- ▶ <http://wiki.apache.org/hadoop/Hbase>
- ▶ George, Lars. *HBase The Definitive Guide* (O' Reilly 2011)

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *IBM Platform Computing Integration Solutions*, SG24-8081
- ▶ *IBM Platform Computing Solutions*, SG24-8073
- ▶ *Implementing the IBM General Parallel File System (GPFS) in a Cross-Platform Environment*, SG24-7844
- ▶ *Implementing IBM InfoSphere BigInsights on System x*, SG24-8077
- ▶ *Integration of IBM Platform Symphony and IBM InfoSphere BigInsights*, REDP-5006
- ▶ *Platform Process Manager Version 9 Release 1*, go to <http://www.ibm.com/shop/publications/order> and search for the document
- ▶ *Workload Optimized Systems: Tuning POWER7 for Analytics*, SG24-8057

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- ▶ *Administering Platform Application Center*, SC22-5396-01
- ▶ *Cluster and Application Management Guide*, SC22-5368-00
- ▶ *Connector for Microsoft Excel User Guide*, SC27-5064-01
- ▶ *IBM General Parallel File System Version 3 Release 5.0.7: Advanced Administration Guide*, SC23-5182-07
- ▶ *IBM General Parallel File System Version 3 Release 5.0.7: Concepts, Planning, and Installation Guide*, GA76-0413-07
- ▶ *IBM Platform Cluster Manager Advanced Edition Administering Guide*, SC27-4760-01
- ▶ *IBM Platform MPI User's Guide*, SC27-4758-00
- ▶ *IBM Platform Symphony Version 6 Release 1.0.1 Application Development Guide*, SC27-5078-01
- ▶ *Platform Symphony Version 6 Release 1.0.1 Cluster and Application Management Guide*, SC27-5070-01

- ▶ *Platform Symphony Version 6 Release 1.0.1 Platform Symphony Reference*, SC27-5073-01
- ▶ *Platform Symphony Foundations - Platform Symphony Version 6 Release 1.0.1*, SC27-5065-01
- ▶ *Platform Symphony Reference*, SC22-5371-00
- ▶ *Platform Symphony Version 6 Release 1.0.1 Integration Guide for MapReduce Applications*, SC27-5071-01
- ▶ *User Guide for the MapReduce Framework in IBM Platform Symphony - Advanced Edition*, GC22-5370-00

Online resources

These websites are also relevant as further information sources:

- ▶ GPFS Frequently Asked Questions and Answers
http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=%2Fcom.ibm.cluster.gpfs.doc%2Fgpfs_faqs%2Fgpfsclustersfaq.html
- ▶ IBM InfoSphere BigInsights 2.1
<http://www.ibm.com/software/data/infosphere/biginsights/>
- ▶ IBM Platform Computing
<http://www-03.ibm.com/systems/technicalcomputing/platformcomputing/>
- ▶ IBM Technical Computing
<http://www-03.ibm.com/systems/technicalcomputing/>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



IBM Technical Computing Clouds

(0.5" spine)
0.475" <-> 0.873"
250 <-> 459 pages



IBM Technical Computing Clouds



Provides cloud solutions for technical computing

Helps reduce capital, operations, and energy costs

Documents sample scenarios

This IBM Redbooks publication highlights IBM Technical Computing as a flexible infrastructure for clients looking to reduce capital and operational expenditures, optimize energy usage, or re-use the infrastructure.

This book strengthens IBM SmartCloud solutions, in particular IBM Technical Computing clouds, with a well-defined and documented deployment model within an IBM System x or an IBM Flex System. This provides clients with a cost-effective, highly scalable, robust solution with a planned foundation for scaling, capacity, resilience, optimization, automation, and monitoring.

This book is targeted toward technical professionals (consultants, technical support staff, IT Architects, and IT Specialists) responsible for providing cloud-computing solutions and support.

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

SG24-8144-00

ISBN 0738438782