Redbooks

ibm.com/redbooks

IBM Reference Architecture for Genomics, Power Systems Edition

Dino Quintero

Luis Bolinches

Marcelo Correia Lima

Katarzyna Pasierb

William dos Santos

Analytics

Power Systems

IBM

Redbooks

International Technical Support Organization

IBM Reference Architecture for Genomics, Power Systems Edition

April 2016

SG24-8279-00

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

First Edition (April 2016)

This edition applies to the IBM Elastic Storage Server (ESS) 3.0.2 (based on IBM Spectrum Scale 4.1.0.7), RHEL 7.2, Ubuntu 14.04, IBM Platform Cluster Manager 4.2.1, IBM Platform Application Center (PAC) 9.1.4.1, IBM Platform LSF 9.1.3, IBM Platform Report Track Monitor (RTM) 9.1.3, IBM Platform Process Manager (PPM) 9.1.3.1.

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, MD-NC119, Armonk, NY 10504-1785, US

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 jurisdictions 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 provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 actual people or business enterprises 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. The sample programs are provided “AS IS”, without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AIX®
BigInsights®
Cognos®
DataStage®
DB2®
FlashCopy®
GPFS™
IBM Platform™
IBM Spectrum Scale™
IBM Watson™
IBM®
InfoSphere®
Linear Tape File System™
LSF®
Power Systems™
POWER8®
PowerHA®
PowerVM®
POWER®
Redbooks®
Redpaper™
Redbooks (logo)  ®
SPSS®
System i®
System Storage®
WebSphere®

The following terms are trademarks of other companies:

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

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

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

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

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

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

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

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

Find and read thousands of IBM Redbooks publications

- Search, bookmark, save and organize favorites
- Get personalized notifications of new content
- Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App

iOS **Download Now** Android

Promote your business in an IBM Redbooks publication

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

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

ibm.com/Redbooks
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redbooks® publication introduces the IBM Reference Architecture for Genomics, IBM Power Systems™ edition on IBM POWER8®. It addresses topics such as why you would implement Life Sciences workloads on IBM POWER8, and shows how to use such solution to run Life Sciences workloads using IBM Platform™ Computing software to help set up the workloads. It also provides technical content to introduce the IBM POWER8 clustered solution for Life Sciences workloads.

This book customizes and tests Life Sciences workloads with a combination of an IBM Platform Computing software solution stack, Open Stack, and third party applications. All of these applications use IBM POWER8, and IBM Spectrum Scale™ for a high performance file system.

This book helps strengthen IBM Life Sciences solutions on IBM POWER8 with a well-defined and documented deployment model within an IBM Platform Computing and an IBM POWER8 clustered environment. This system provides clients in need of a modular, cost-effective, and robust solution with a planned foundation for future growth.

This book highlights IBM POWER8 as a flexible infrastructure for clients looking to deploy life sciences workloads, and at the same time reduce capital expenditures, operational expenditures, and optimization of resources.

This book helps answer clients' workload challenges in particular with Life Sciences applications, and provides expert-level documentation and how-to-skills to worldwide teams that provide Life Sciences solutions and support to give a broad understanding of a new architecture.

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

Luis Bolinches has been working with IBM Power Systems servers for over 10 years and has been with IBM Spectrum Scale (GPFS™) for six years. He is currently at IBM Lab Services, and has been an IBM employee in Spain, Estonia, and now in Finland.

Marcelo Correia Lima is a Business Intelligence Architect at IBM. He has 17 years of experience in development and integration of Enterprise Applications. His current area of expertise is Business Analytics Optimization (BAO) Solutions. He has been planning and managing Solutions lifecycle implementations that involve Multidimensional Modeling, IBM InfoSphere® Data Architect, IBM InfoSphere DataStage®, IBM Cognos® Business Intelligence and IBM DB2®. In addition, Marcelo has added Hadoop, Big Data, IBM InfoSphere BigInsights®, Cloud Computing, and IBM Platform Computing to his background.

Marcelo holds Level 1 IT Specialist Certification Experienced Level in the Services and Business Information Management specializations (by The Actualizing IT Solutions Capability Center - IBM Integrated Technology Delivery Headquarters). Before working as a Business Intelligence Architect, he was involved in the design and implementation of IBM WebSphere® and Java Enterprise Edition Applications for IBM Data Preparation/Data Services.

Katarzyna Pasierb is a learning enthusiast was spent couple of years at Wroclaw University of Technology where she got her first experience with System i® (AS/400). She wrote a few publications on e-health, led classes in Data Warehouse, and contributed to two European projects, which gave her experience with research. Kasia works in Support at IBM, and two years ago joined IBM Kraków's Lab to work with the European DB2 Advanced Support team. You can find her on twitter as @kepasierb, or contact her at katarzyna@pl.ibm.com.

William dos Santos is an IT Specialist at IBM Brazil. He has 20 years of experience in working on cross-UNIX platforms on data center migrations and consolidation projects. His areas of expertise include IBM Power Systems, IBM PowerHA®, IBM PowerVM®, IBM AIX®, Solaris, and Linux. He holds an MBA in Project Management, is certified at the IBM IT Specialist Certification Experienced Level, and holds many other technical certifications in Linux and UNIX systems.

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

Ella Bushlovic
International Technical Support Organization, Poughkeepsie Center

Linda Cham
Tzy-hwa Tzeng
Fernando Pizzano
John Dunham
Karen Wood
Duane Witherspoon
David Shapiro
IBM Poughkeepsie

Jim Woodbury
IBM Rochester

Frank Lee
IBM St. Louis

Richard Wale
IBM United Kingdom

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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

1

Introduction

The biological life sciences have undergone a revolution in studies ranging from single-gene or simplistic disease mechanisms to large scale and complex systems-based molecular studies. This change has been driven by advances in genome sequencing throughput and lowered costs. The broad adoption of high-throughput genome sequencing has accelerated the pace of discovery of the causal mechanisms of diseases. However, the new pace has created very large amounts of data that has overwhelmed the ability of scientists and engineers who are used to hand-written notes to track it effectively.

A new model for large-scale computation happened to also be rapidly evolving at the same time. Life sciences organizations face huge pipeline and productivity challenges. To increase discovery, productivity, innovation in research and development, and compete more effectively, life sciences organizations must establish an optimized, flexible, and resilient infrastructure foundation to improve clinical development processes. With shifting regulatory burdens and the need to compress the timeline from discovery to approval, research teams need comprehensive, high-performance technical computing infrastructure solutions with the flexibility to process massive amounts of data and support increasingly sophisticated analyses.

Genomic medicine promises to revolutionize medical research and clinical care. By investigating the human genome in the context of biological pathways and environmental factors, genomic scientists and clinicians can now identify individuals at risk of disease, provide early diagnoses based on biomarkers, and recommend effective treatments.

IBM Platform Computing solutions can help life sciences customers by delivering high performance resources that use the advantages of cloud.

This chapter introduces IBM Platform Computing for life sciences, and how its portfolio of solutions supports high-performance computing (HPC) management for genomics.

The following topics are presented in this chapter:

- Life sciences in genomics
- IBM high-performance computing solutions

1.1 Life sciences in genomics

It is now believed that the information generated by genomics will, in the long term, have major benefits for the prevention, diagnosis, and management of many diseases that either have been difficult or impossible to control. These include communicable and genetic diseases, together with other common killers and causes of chronic ill health, including cardiovascular disease, cancer, diabetes, the major psychoses, dementia, rheumatic disease, asthma, and many others.[1]

The multi-factorial nature of many complex disorders such as diabetes, obesity, heart disease, Alzheimer's, schizophrenia, bipolar disorder, and autism spectrum disorder (ASD) requires sophisticated computational capabilities. The system must aggregate and analyze large stream of data (genomics, proteomics, imaging) and observation points (clinical, behavioral, environmental, real-world evidence) from a wide range of sources.

For example, it is difficult for doctors to collect data from children to determine whether they can be classified under ASD. ASD is a term for a group of complex disorders of brain development characterized in varying degrees by difficulties in social interaction, verbal and nonverbal communication, and repetitive behaviors. Scientists do not know the exact causes of ASD, but research suggests that both genes and environment play important roles.

Children with ASD do not follow typical patterns when developing social and communication skills. Doctors follow guidelines in the *Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition* (DSM-5) to determine the diagnosis. Although the goal of DSM-5 is to standardize psychiatric diagnostic categories and criteria, it can cause disagreements because the validity and reliability of the diagnostic categories can be subjective, and varies from doctor to doctor.

Parents are usually the first to figure out unusual behaviors in their child. Often, certain behaviors become more noticeable when comparing children of the same age. Children with ASD who have not yet developed meaningful gestures or language can scream or grab or otherwise act out until they are taught better ways to express their needs.

Alzheimer's disease is a progressive illness in which the greatest known risk factor is increasing age. The structure and chemistry of the brain become damaged over time and the person's ability to communicate and reason are affected.

As medical advances continue with the help of technology, the number of new cancer cases can be reduced and many cancer deaths can be prevented. For more information, see Chapter 4, “Medicine of the future with IBM” on page 111.

By combining attributes from the patient’s file with clinical expertise, external research, and data, it is possible to identify potential treatment plans for a patient. This process means that genomic medicine allows doctors to have an even more detailed view of their patients. The development of databases and file repositories that are interconnected based on global data sharing and federated networks, bring the promise of innovative and smarter approaches to access and analyze data in unprecedented scale and dimension.

In the medical and scientific areas, the data sources can be, for example, medical devices, clinical systems, medical research, clinical reports, and medical exams.

[1] [a] *Genomics and World Health: Report of the Advisory Committee on Health research, Geneva, WHO (2002), http://apps.who.int/iris/bitstream/10665/42453/1/a74580.pdf*

By examining the DNA of individuals and family members with ASD and Alzheimer's, researchers can look for genetic variations. Furthermore, it is possible to look for data patterns and risk factors that ASD and Alzheimer's have in common.

The medical benefit of genomics research is that it can collect data, look for patterns, and convert these patterns into medical assumptions that can be used for further research or diagnosis in support of a doctor's treatment paths. As the size of available databases increases, the search of genome sequences requires more time, so you need technology that enables high throughput of data. Data management is critical for genomics because of the volume, velocity, and variety of data. The I/O throughput on a genomic system can be extremely demanding due to data volume and the large number of file and directory objects.

IBM has the combination of deep industry expertise; a comprehensive portfolio of hardware, software, and services; and a proven methodology to help life sciences organizations develop their analytics strategy with an actionable roadmap.

To increase discovery productivity, innovate in research and development, and compete more effectively, organizations can establish an optimized, flexible, and resilient infrastructure foundation to improve clinical development processes.

1.2 IBM high-performance computing solutions

The exponential growth in the amount of genomic data has spurred growing interest in large-scale analysis of genetic information.

Genomics applications, which explore computational methods to allow researchers to sift through the massive data and extract useful information, are becoming increasingly important computer workloads.

1.2.1 Cluster, grids, and clouds

A *cluster* is typically a single application or single group. Because clusters are in multiple applications, multiple groups, and multiple locations, they have become more of a grid. In the past, you needed advanced policy-based scheduling to manage clusters.

In the era of cloud computing, you can use a much more dynamic infrastructure against an infrastructure that allows on-demand self-service. Many grid clients already considered their grids to be clouds. The evolution to cloud computing continues with the ability of the platform to manage the heterogeneous complexities of distributed computing. This management capability has many applications in the cloud.

Figure 1-1 shows the cluster, grid, and HPC cloud evolution.

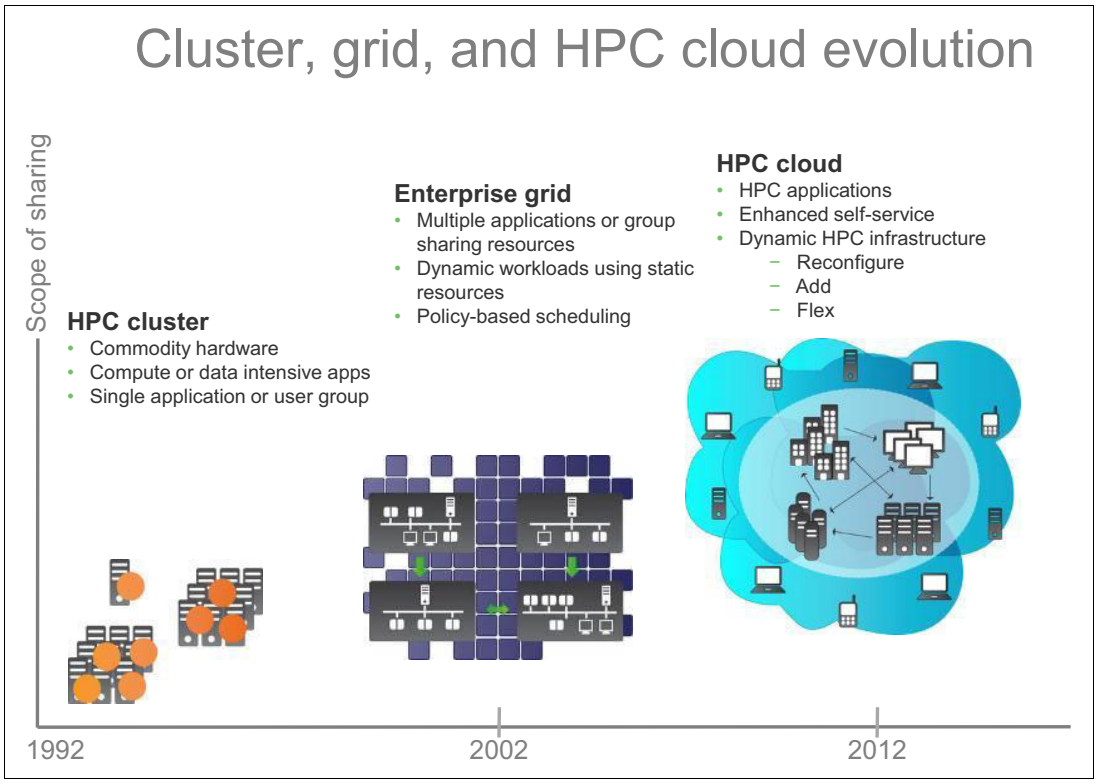


Figure 1-1 Evolution of distributed computing

1.2.2 HPC cluster for public and private cloud

IBM Platform HPC is a complete, high-performance computing management solution in a single product. Its robust cluster and workload management capabilities are accessible by using the latest design in web-based interfaces, making it powerful and simple to use. The solution includes a set of cluster and workload management features that help reduce the complexity of your HPC environment and improve your time-to-results. IBM Platform HPC provides a unified set of management capabilities that make it easy to harness the power and scalability of a technical cluster. These capabilities result in shorter time to system readiness, increasing user productivity and optimizing throughput. IBM Platform HPC includes the following features:

- Cluster management
- Workload management
- Workload monitoring and reporting
- System monitoring and reporting
- IBM Platform MPI
- Integrated application scripts and templates for job submission
- Central web portal
- High availability

1.2.3 Genomic discoveries with IBM Platform Computing solutions

Life sciences organizations face a large pipeline and productivity challenges. To increase discovery, productivity, innovation in research and development, and compete more effectively, organizations must establish an optimized, flexible, and resilient infrastructure

foundation to improve clinical development processes. IBM Platform Computing solutions cluster, grid, and HPC cloud management software can help address these challenges. For more information, see “Streamlining Genomics research with HPC & Big Data” at:

http://www.ibm.com/systems/platformcomputing/industries/lifesciences.html

Accelerate compute and data-intensive workloads

IBM Platform Computing solutions provide cluster, grid, and HPC cloud management software to support all of your technical computing applications. The software is designed to accelerate time-to-results for compute and data-intensive applications, accelerating workloads as diverse as genome sequencing, drug design, simulations, product design analysis, and risk management on distributed computing environments.

IBM Platform Computing product family helps to improve the confidence of life sciences IT professionals that all available resources are fully used from application software licenses to available network bandwidth.

Data management is critical for genomics because of the volume, velocity, and variety of data. Genomic data volume is surging as the cost of sequencing drops precipitously. The I/O throughput on a genomic system can be extremely demanding due to data volume and the large number of file and directory objects. Many data formats also exist, with varying degrees of lifecycle management requirements that range from transient files in scratch space to variant calling files that must always remain online.

Using IBM Spectrum Scale, the Datahub data-management layer defines an enterprise capability to meet these challenges based on its high-performance, scalable, and extensible architecture.

Data, data, and more data

As data volumes grow and technical applications require more compute power, life sciences companies of all sizes need to transform and to integrate their infrastructure to meet the increased demand. IBM enables companies to accelerate time-to-value with intuitive and powerful solutions designed for large data volumes and compute-intensive applications. IBM offers Application Ready Solutions for the life sciences industry that are optimized for technical workloads, simplified for easy adoption and management, and designed to help companies increase their competitive advantage with an agile, high-performance technical computing environment.

Improved cluster utilization

By improving utilization, resources can be kept readily available, helping researchers to get more work done in a shorter amount of time. This benefit can improve collaboration across the clinical development value chain for better insights and superior results. IBM Platform Computing solutions are designed to provide these benefits:

- Optimize resource utilization and reduce costs with robust workload management
- Increase throughput for faster time-to-results with intelligent scheduling
- Improve productivity with simple, intuitive management
- Enable processing of big data in a fast and cost efficient way

IBM Platform Computing enables companies to accelerate time-to-value with intuitive and powerful clusters, grid, and HPC cloud management software solutions designed for large data volumes and compute-intensive applications.

IBM Platform Computing solutions for cluster, grid, and HPC cloud management software can help address these challenges. Application software solutions help manage and accelerate workload processing and help ensure their completion across a distributed, shared IT

environment, while fully using all HPC resources. Using IBM Spectrum Scale, the Datahub data-management layer defines an enterprise capability to meet these challenges based on its high-performance, scalable, and extensible architecture.

In this publication, IBM life sciences engages IBM HPC with a cluster solution and enables it for genomics research.

2

Reference architecture

This chapter describes the IBM Reference Architecture for Genomics solutions in an IBM POWER8 clustered environment. This technical-computing architecture supports life science workflows. Its goal is to contextualize key concepts, components, roles, templates, structures, and rules that are supposed to help guide and manage the solution and its components.

The following topics are discussed in this chapter:

- IBM Reference Architecture for Genomics
- Hardware and components
- Software and components
- Operations
- Network

2.1 IBM Reference Architecture for Genomics

Whether engaged in genome sequencing, drug design, product analysis, or risk management, life sciences research teams need high-performance technical environments with the ability to process massive amounts of data and support increasingly sophisticated simulations and analyses.

Organizations that are looking for root causes and cures for diseases need speed, agility, and control across the clinical development lifecycle to increase productivity, foster innovation, and compete more effectively.

To address the needs for speed, scale, and smarts for genomic medicine, IBM has created an end-to-end reference architecture that defines the most critical capabilities for genomics computing:

- Data management (Datahub)
- Workload orchestration (Orchestrator)
- Enterprise access (AppCenter)

To determine the inclusion and priorities for the building blocks of the reference architecture (capabilities and functions) and mapped solutions (hardware and software), follow these main principles:

- Software-defined: Defining the infrastructure and deployment model based on software-based abstraction layers for computation, storage, and cloud. This configuration helps *future-proof* the genomic infrastructure for growth and expansion as data volume and computing loads continue to increase.
- Data-centric: Meeting the challenge of explosive growth of genomics, imaging, and clinical data with data management capabilities.
- Application-ready: Integrating a multitude of applications into a consistent environment that supports data management, version control, workload management, workflow orchestration, and access for execution and monitoring.

Figure 2-1 illustrates the IBM Reference Architecture for Genomics (v7.6). The purple box depicts the personalized medicine platform. The green box depicts the translational platform. The blue box depicts the genomics platform. These three platforms for genomic medicine can share common enterprise capabilities:

- *Datahub* for data management
- *Orchestrator* for workload management
- *AppCenter* for access management

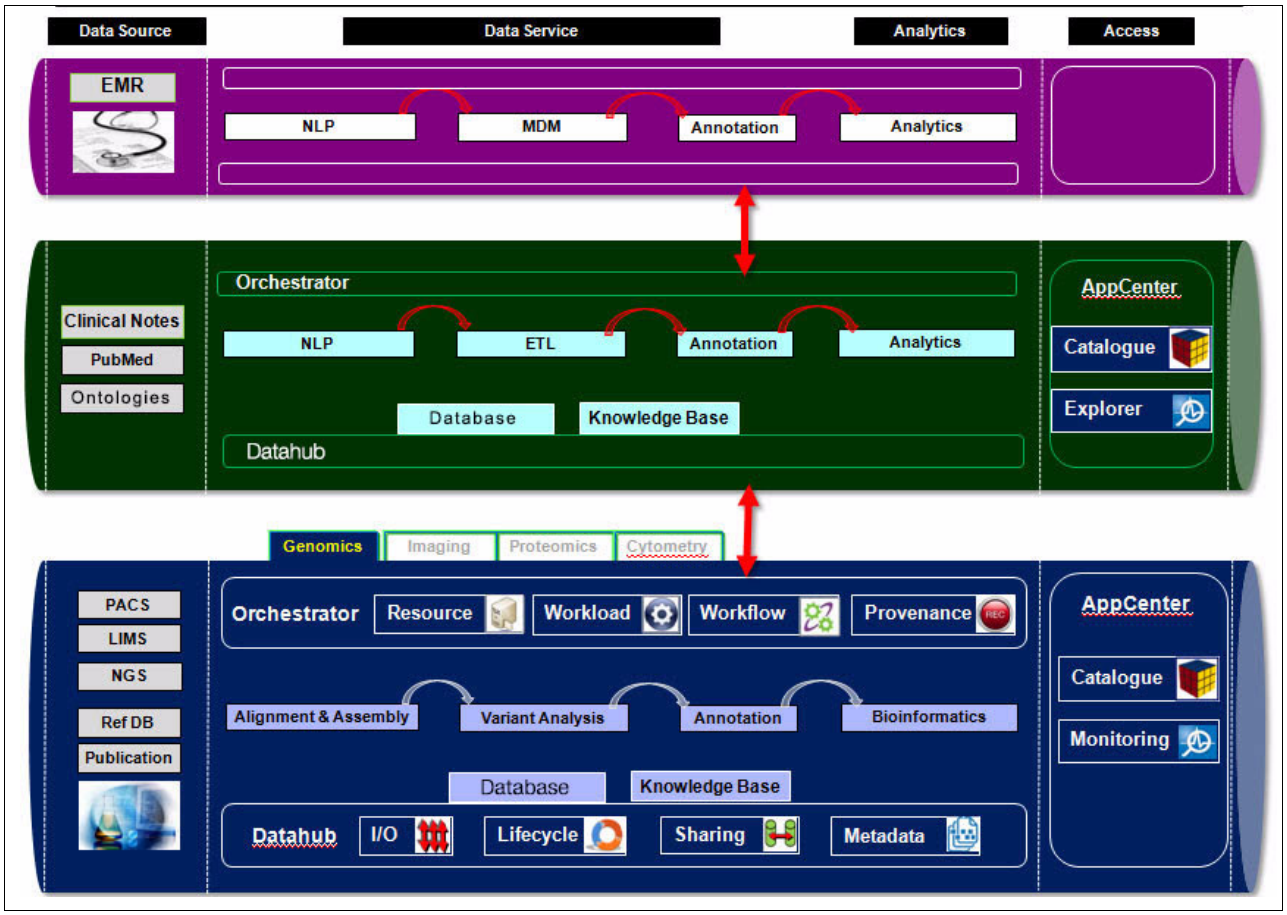


Figure 2-1 IBM Reference Architecture for Genomics

The IBM Reference Architecture for Genomics first serves as a master plan for composing capabilities for data, workload, and access management into system architecture for platforms such as genomics (blue box), translational research (green box), and personalized medicine (purple box). The system architecture then can be implemented as a genomics computing infrastructure.

2.1.1 Reference architecture as master plan for deployment

IBM Reference Architecture for Genomics can be deployed with various infrastructure solutions and technologies. IBM has also been working with a growing ecosystem of customers and partners to enrich the portfolio of solutions and products that can be mapped into the architecture.

Figure 2-2 shows the deployment model for the IBM Reference Architecture for Genomics.

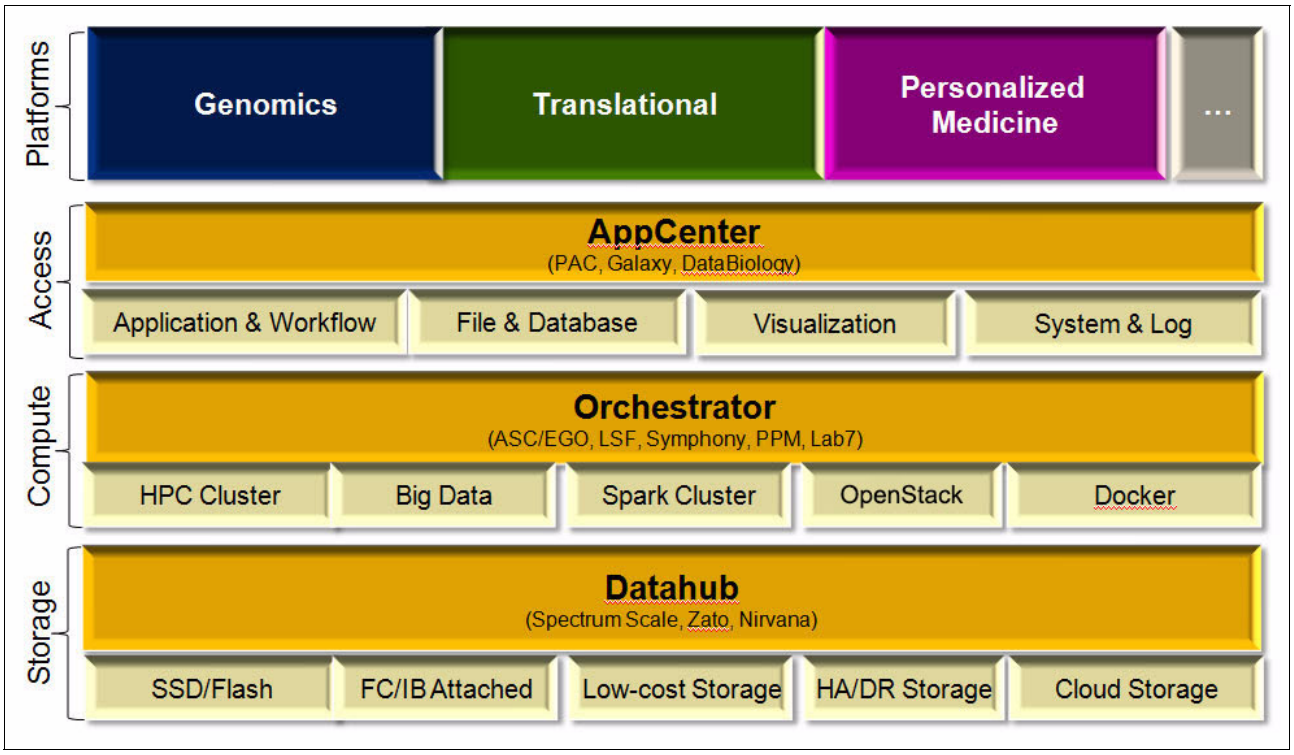


Figure 2-2 Deployment model for the IBM Reference Architecture for Genomics

The infrastructure technologies for storage (SSD, Flash, disk, cloud), compute (HPC, Big Data, Spark, OpenStack, Docker), and informatics technologies for user access (application/workflow, file protocol, database query, visualization, monitoring) are all managed by the three enterprise capabilities of Datahub, Orchestrator, and AppCenter as shown in Figure 2-2). Various commercial productions ranging from IBM Spectrum Scale to open source solutions such as Galaxy are mapped to these capabilities. These solutions and products can then be composed into a deployable platform for genomics, translational, or personalized medicine.

Datahub for data management

Data management is the most fundamental capability for genomics platforms because a huge amount of data needs to be processed at the correct time and placed with feasible costs. The temporal factors can range from hours to analyze the data in a high performance computing (HPC) system to years when data needs to be recalled from a storage archive for reanalysis. The spatial aspect can span a local infrastructure that provides near-line storage capability to a cloud-based remote cold archive.

Data management challenges

The four *Vs* that define big data also describe genomic data management challenges:

- Very large data size and capacity (Volume)
- Demanding I/O speed and throughput (Velocity)
- Fast-evolving data types and analytical methods (Variety)
- The capability to share and explore large volume of data with context and confidence (Veracity)

In this case, the challenges are increased by extra requirements such as regulatory compliance (patient data privacy and protection), provenance management (full versioning and audit trail), and workflow orchestration.

Data volume

Genomic data volume is surging as the cost of sequencing drops precipitously. It is common for an academic medical research center (AMRC) equipped with next-generation sequencing technologies to double its data storage capacity every 6-12 months. Consider a leading AMRC in the New York City (NYC) that started 2013 with 300 TB of data storage. By the end of 2013, the storage volume surged passing 1 PB (1000 TB), more than tripling the amount from 12 months before. What made it even more astonishing was that the speed of growth has been accelerating and continues to today. For some of world's leading genomic medicine projects such as Genome England (UK), Genome Arabia (Qatar), Million Veteran Project (US), and China National GeneBank, the starting points or baseline for data volume are no longer measured in terabytes (TB) but tens and hundreds of petabytes (PB).

Data velocity

The data velocity in a genomic platform can be extremely demanding due to these divergent requirements:

- Very large files: These files are used to store genomic information from study subject, which can be a single patient or group of patients. There are two main types of such files: Genomic sequence alignment (BAM or Binary Alignment/Map) and genetic variants (VCF or Variant Call File). They are often larger than 1 TB and can take up half of total storage volume for a typical genomic data repository. Additionally, these files are quickly growing larger, often the result of condensing more genomic information from higher resolution coverage (for example, from 30X to 100X for full genome) or a larger study size. As the genomic research evolves from Rare Variant study (variant calling from a single patient) to the Common Variant study, there is an emerging need to make joint variant calling from thousands of patient samples. Consider a hypothetical case provided by the Broad Institute. For 57,000 samples to be jointly called, the input BAM file is 1.4 PB and the output VCF file is 2.35 TB, which are both extremely large by today's standards, but soon will become commonly used.
- Many small files: These files are used to store raw or temporary genomic information such as output from sequencers (for example, BCL file format from Illumina). They are often smaller than 64 KB, and can take up half of total file objects for a typical genomic data repository. Because each file I/O requires two operations for data and metadata, workload that generates or requires access to large number of files creates a distinct challenge that is different from that of large files. In this case, the velocity can be measured in I/O operations per second (IOPS) and they typically reach millions of IOPS for the underlying storage system. Consider a storage infrastructure at a San Diego-based AMRC that was not optimized for massive small file operation. A workload such as BCL conversion (for example, CASAVA, from Illumina) stalled because compute servers were constrained with limited I/O capability, especially IOPS. A benchmark shows that the CPU efficiency drops to single digits because the computing power is being wasted waiting for data to be served. To alleviate this computational bottleneck, IBM researchers developed a data caching method and tool to move I/O operation from disk into memory.
- Parallel and workflow operation: To scale performance and speed up time-to-results, genomics computing is often run as an orchestrated workflow in batch mode. This parallel operation is essential to deliver fast turnaround as workloads evolve from small-scale targeted sequencing to large-scale full-genome sequencing. With hundreds to thousands of diverse workloads running concurrently in such a parallel computational environment, the requirement for storage velocity as measured in I/O bandwidth and IOPS is aggregated and will increase explosively. Consider a bioinformatics application from the

NYC AMRC. This application can run in parallel on 2500 compute cores, each writing the output to the disk at a rate of 1 file per second and collectively creating millions of data objects, either 2500 folders each with 2500 files or 14 million files in one directory. This workload is one of many that contributed to a data repository with 600 million objects, including 9 million directories that each contain only one file. Due to the massive amount of metadata, the IOPS load was constraining the overall performance so much that even a file system command to list files (`ls` in Linux) took several minutes to complete. A parallel application such as the GATK Queue also suffered from poor performance. In early 2014, the file system was overhauled with an emphasis on improving the metadata infrastructure. As a result, both bandwidth and IOPS performance were significantly improved and the benchmark showed a 10X speedup of a gene-disease application without any application tuning.

Data variety

Many types of data formats need to be handled in terms of storage and access. The data formats range from intermediary files that are created during a multi-step workflow to output files that contain vital genomic information to reference data sets that need the version to be carefully tracked. The common approach today is to store all this data in online or near-line disk in one storage tier despite the expense of this approach. One practical constraint is the lack of lifecycle management capability for the massive amount of data. If it takes the genomic data repository a long time to scan the file system for candidate files for migration or backup, it becomes impossible to complete this task in a timely fashion. Consider a large US genome center that is struggling to manage its fast-growing data as it adopts Illumina X10 sequencer for full genome sequencing. To complete a scan of the entire file system, it currently takes up to four days, making daily or even longer backup windows impossible. As a result, data is piling up quickly in the single-tier storage and slowing down the metadata scan and performance even further, causing a vicious cycle for data management.

Another emerging challenge for data management is created by the spatial varieties of data. As inter-institutional collaboration becomes more common and a large amount of data needs to be shared or federated, the locality becomes an indispensable character of data. The same data set, especially reference data or output data, can exist in multiple copies in different locations, or in duplicates in the same location due to regulatory compliance requirement (for example, physically isolating a clinical sequencing platform from one for research). In this case, managing the metadata efficiently to reduce data movement or copying will not only reduce cost due to extra storage, but also minimize problems due to tracked the version and synchronization.

Data veracity

The multi-factorial nature of many complex disorders such as diabetes, obesity, heart disease, Alzheimer's, and autism spectrum disorder (ASD) requires sophisticated computational capabilities to aggregate and analyze large stream of data (genomics, proteomics, imaging) and observation points (clinical, behavioral, environmental, real-world evidence) from a wide range of sources. The development of databases and file repositories that are interconnected based on global data sharing and federated networks bring the promise of innovative and smarter approaches to access and analyze data in unprecedented scale and dimensions.

The veracity (trustworthiness) of data is an essential element in this context. For example, clinical data (genomics and imaging) needs to be properly de-identified to protect the confidentiality of the study subject. Genomic data needs to have end-to-end provenance tracking from bench to bedside to provide full audit trail and reproducibility. The authorship and ownership of data needs to be properly represented in a multi-tenancy and collaborative infrastructure. With built-in capability to handle data veracity, a genomic computing

infrastructure can enable the researchers and data scientists to share and explore a large volume of data with context and confidence.

Datahub functions

To address the challenges of data management for genomics, define an enterprise capability that functions as a scalable and extensible layer for serving data and metadata to all workloads. Name this layer `Datahub` to reflect its critical role as a central hub for data, which is storing, moving, sharing, and indexing massive amount of genomic raw and processed data.

The Datahub also manages the underlying heterogeneous storage infrastructure from SSD/Flash to disk to tape to Cloud (Figure 2-3).

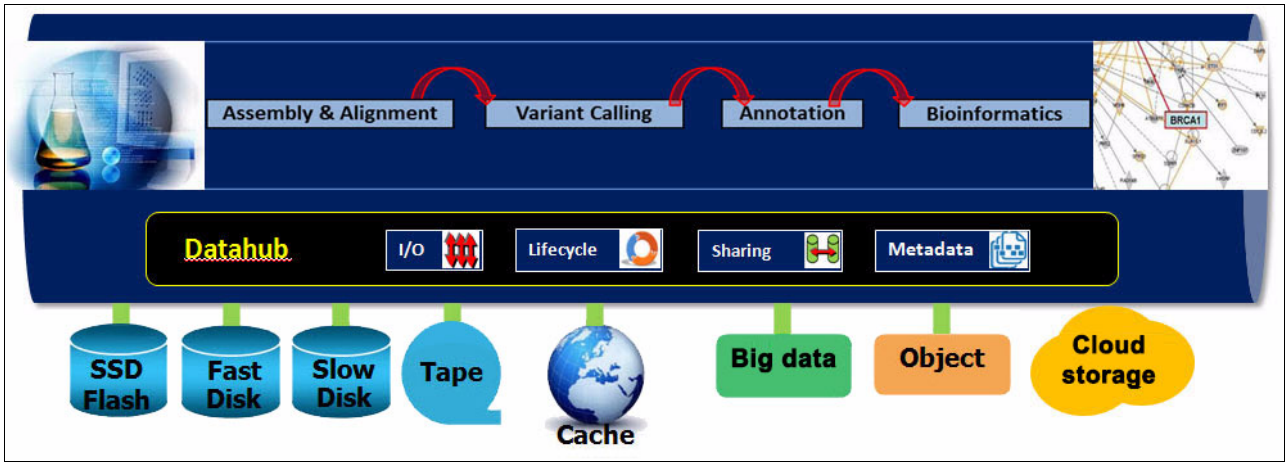


Figure 2-3 Overview of Datahub

The Datahub is the enterprise capability for serving data and metadata to all the workloads (Figure 2-3). It defines a scalable and extensible layer that virtualizes and globalizes all storage resources under a global name space. Datahub is designed to provide four key functions:

- High-performance data input/output (I/O)
- Policy-driven information lifecycle management (ILM)
- Efficient data sharing through caching and necessary replication
- Large-scale metadata management

For physical deployment, the Datahub can support an increasing number of storage technologies as modular building blocks, including these:

- Solid-state disk (SSD) and Flash storage system
- High-performance fast disks
- Large-capacity slow disks (4 TB per drive)
- High-density and low-cost tape library
- External storage cache that can be locally or globally distributed
- Big data storage based on Hadoop
- Cloud-based external storage

Orchestrator functions

Through the Orchestrator, the IBM Reference Architecture for Genomics defines the capability to orchestrate resources, workload, and workflow as shown in Figure 2-4. A unique combination of the workload manager and workflow engine links and coordinates a spectrum of computational and analytical jobs into fully automated pipelines that can be easily built, customized, shared, and run on a common platform. This configuration provides the necessary abstraction of applications from the underlying infrastructure such as a high-performance computing cluster with a graphics processor unit (GPU) or a big data cluster in the cloud.

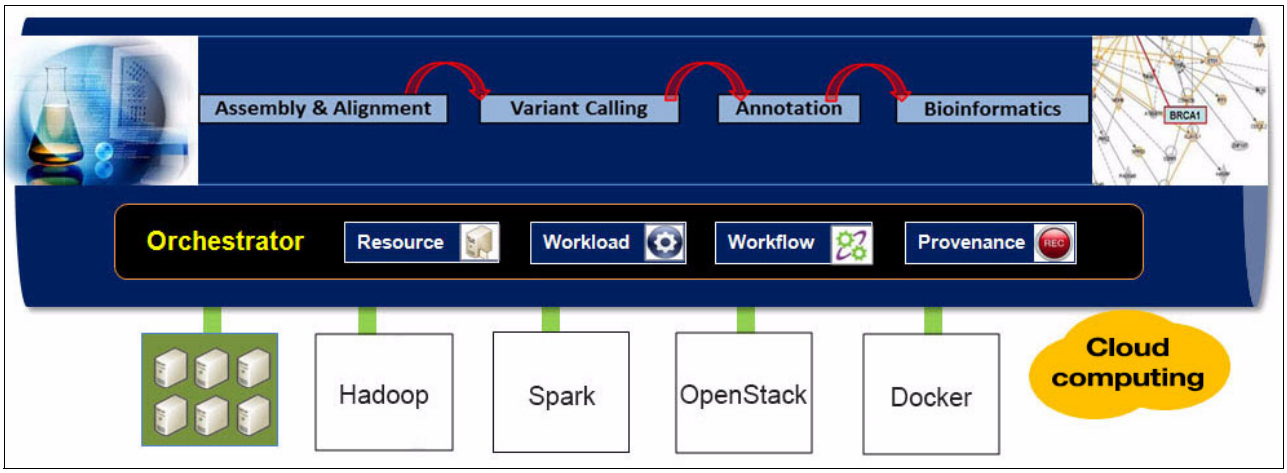


Figure 2-4 Orchestrator overview

The Orchestrator is the enterprise capability for orchestrating resource, workloads, and managing provenance as shown in Figure 2-4. It is designed to provide these key functions:

- Resource management by allocating infrastructure towards computational requirement dynamically and elastically
- Workload management by efficient placement of jobs onto various computational resources such as local or remote clusters
- Workflow management by linking applications into logical and automated pipelines
- Provenance management by recording and saving metadata associated with the workload and workflow

By mapping and distributing workloads to elastic and heterogeneous resources (HPC, Hadoop, Spark, OpenStack/Docker, Cloud) based on workflow logic and application requirements (for example, architecture, CPU, memory, I/O), the Orchestrator defines the abstraction layer between the diverse computing infrastructure and the fast-growing array of genomic workloads.

Resource Manager

The Resource Manager in the Orchestrator functions allocate computational resources in a policy-driven way to meet the computational needs of genomic workloads. The most commonly used resource is a high-performance computing bare-metal cluster (HPC). The Resource Manager can either one-time provision the resource, or dynamically shift and allocate resources. Just as Datahub I/O Management provides a layer of storage services, the Resource Manager provides a fabric of computing services. Additionally, a new type of infrastructure can be added into the resource pool. These include big data Hadoop cluster, Spark cluster, OpenStack virtual machine cluster, and Docker cluster.

The ability to manage and shift resources based on the load information from the workload is a requirement for the Resource Manager. As an example, for a genomic infrastructure that is shared between a batch alignment job and Spark machine learning job, as the workload fluctuates during the run time, the Resource Manager is able to sense the utilization level and shift resources, in the form of computing slots or containers, from supporting one job to the other.

Workload Manager

The Workload Manager function in the Orchestrator enables the genomic computing resources, as made available by the Resource Manager, to efficiently share, use, and deliver optimal performance to the genomic applications. The Workload Manager is able to handle demanding, distributed, and mission-critical applications such as Illumina ISSAC CASAVA, bcltofastq, BWA, SAMtools, SOAP, and GATK. The Workload Manager also needs to be highly scalable and reliable to manage a large number of jobs submitted in batches, a common requirement for mid-to-large genomic computing infrastructure. For ever-growing numbers of genomics applications with different level of maturity and architectural requirements (for example, CPU, GPU, large-memory, MPI), Workload Manager provides the necessary resource abstraction so that jobs can be transparently handled for submission, placement, monitoring, and logging.

Workflow Engine

This Orchestrator function addresses the need for genomic workflow management. The Workflow Engine works to connect jobs into a logical network. This network can be a linear progression of computational pipeline in multiple steps such as sequence alignment, assembly, and variant calling. It can also be a much more complex network with conditional branches or loops based on user-defined criteria and requirements for completion.

The Orchestrator Workflow Engine distinguishes itself from traditional workflow tools with its ability to handle complex workflows dynamically and quickly. Individual workloads or jobs can be defined through an easy to use interface, incorporating variables, parameters, and data definition into a standard workflow template. Many workload types can be easily integrated into the workflow engine: Parallel HPC applications, big data applications, or R scripts for analytics workload. After it is defined and validated, the template can be used by users to start the workflow directly from their workstations, or it can be published in an enterprise portal (AppCenter) to be used by other users.

Orchestrator Workflow Engine (Figure 2-5 on page 16) can deliver the following additional values:

- Job arrays: To maximize the throughput of the workflow for genomics sequencing analysis, a special type of workload can be defined by using job arrays so data can be split and processed by many jobs in parallel.
- Subflow: In another innovative use case for genomics processing, multiple subflows can be defined as a parallel pipeline for variant-analysis following the alignment of the genome. The results from each subflow can then be merged into a single output and provide analysts with a comparative view of multiple tools or settings.
- Reusable module: The workflow can also be designed as a module and embedded into larger workflows as a dynamic building block. Besides enabling efficient building and reuse of the pipelines, this approach also encourages collaborative sharing of genomic pipelines among a group of users or within larger scientific communities.

Figure 2-5 shows the following components, starting from the left in the pipeline:

- Box 1: The arrival of data such as BCL files automatically triggers CASAVA as the first step of the workflow.

- Box 2: A dynamic subflow uses BWA for sequence alignment.
- Box 3: SAMtools performs post-processing in a job array.
- Box 4: Different variant analysis subflows can be triggered in parallel.

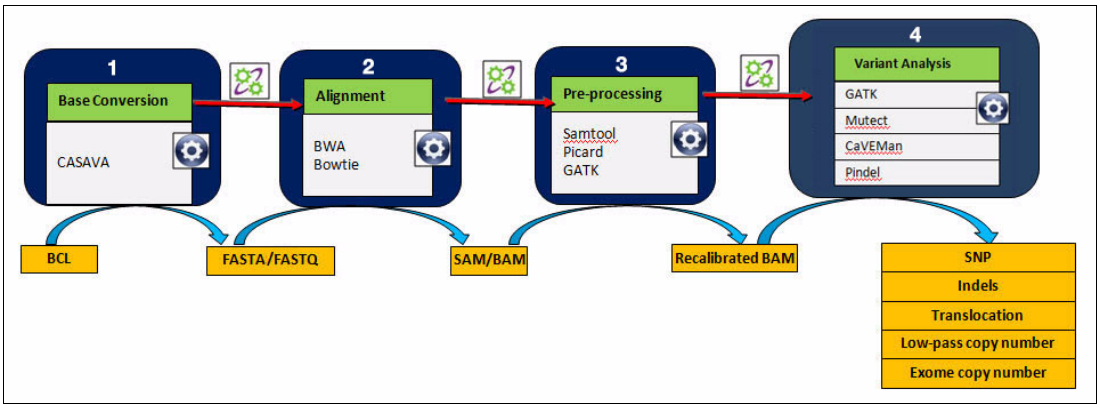


Figure 2-5 Genomic pipeline implemented by using Orchestrator

Figure 2-5 shows a conceptual view of the IBM Genomic Pipeline in which an end-to-end workflow was created to process raw sequence data (BCL) into variants (VCF) using a combination of applications and tools. Each box represents a functional module of the workflow that consists of genomic applications that map to functions such as base conversion, sequence alignment, pre-processing, and variant calling/analysis. These modules can be implemented as stand-alone workflows, and can be connected logically and conditionally into a larger flow such as the one displayed in Figure 2-5.

As more institutions are deploying hybrid cloud solutions with distributed resources, the Orchestrator can coordinate the distribution of workloads based on data localities, predefined policies, thresholds, and real-time input of resource availibilities. For example, a workflow can be designed for processing genomic raw data closer to sequencers, and followed by sequence alignment and assembly using the MapReduce framework on a remote big data cluster. In another use case, a workflow can be designed to start a proxy event of moving data from a satellite system to the central HPC cluster when the genomic processing reaches 50% completion rate. The computation and data movement can happen concurrently to save time and costs.

Another emerging use case for the Orchestrator is the publishing and sharing of genomic pipelines by one research institution with others. Because the software tools enable the saving and distribution of workflow templates, some leading cancer and medical research institutions exchange genomic pipelines to facilitate collaboration.

Provenance Manager

Many computational methods and applications can be applied to assemble, analyze, and annotate the genomic sequences. The choice of applications, reference data, and runtime variables are critical provenance information that can have a significant impact on interpretation and preservation of genomic analysis. Currently, there is little to no public standards or practice to capture the provenance information, which potentially amounts to missing or losing critical data for computational analysis. This problem is potentially compounded by other factors such as complexity of data, workflow, or pipeline as a high-level analytical process, and frequent releases and updates of applications.

Therefore, provenance management is called out as an important function within Orchestrator, which is analogous to the importance of metadata management to Datahub.

One can also think of provenance data as the *metadata of workloads*. The functional requirement for the Provenance Manager is to capture, store, and index user-defined provenance data in a transparent and nondisruptive way to any existing computational workloads or workflow.

Multiple technologies and solutions are being developed or have already been made available. Some of them are commercial solution such as ESP Platform from Lab7 and Nirvana from General Atomics. IBM has also been developing a technology for large-scale and near real-time metadata management system that can work in concert with Datahub and Orchestrator.

2.1.2 Overview of AppCenter

In the IBM Reference Architecture for Genomics, the third enterprise capability called AppCenter is the user interface for accessing both Datahub and Orchestrator. The AppCenter provides an enterprise portal with role-based access and security control while enabling researchers, data scientists, and clinicians to easily access data, tool, applications, and workflows. Its goal is to make complex genomics platforms accessible to research and data scientists who do not have computer programming experience.

AppCenter can be added as part of genomics, translational, and personalized medicine platforms, taking advantage of its reusability.

Figure 2-6 shows an overview of the AppCenter.

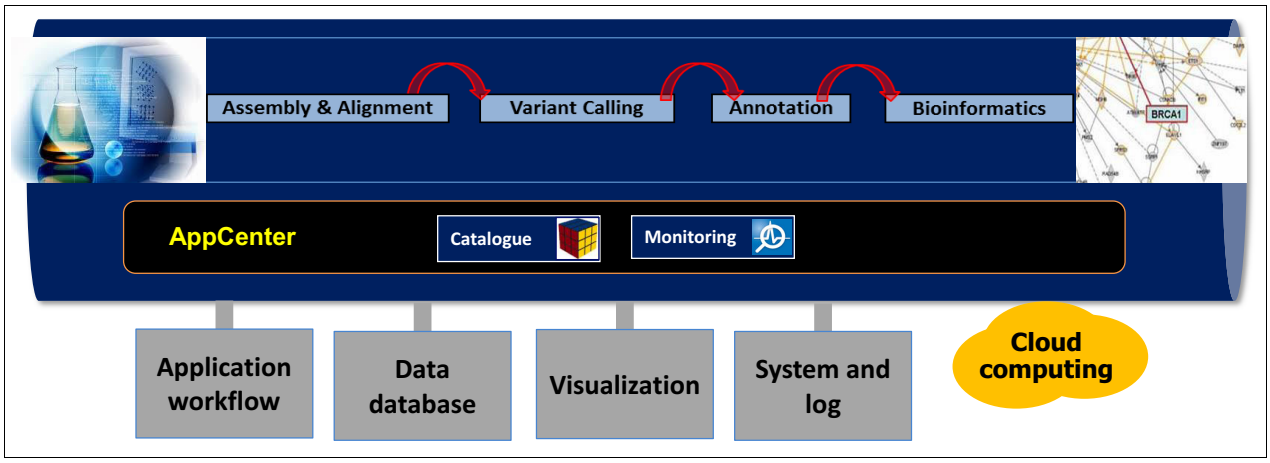


Figure 2-6 Overview of the AppCenter

The AppCenter, as shown in Figure 2-6, is the enterprise capability for accessing the genomics platform to start and monitor workloads, query and explore data, visualize analytical output, and track system log and utilization information. It defines the abstraction layer between users (researchers, clinicians, and analysts) and the Datahub/Orchestrator.

AppCenter has two functions:

- A portal-based catalog that provides access to applications, workflow, data set, and capability for visualization.
- A monitoring capability that allows application-specific information to be monitored, tracked, reported, and managed.

Catalog

To minimize or eliminate the barriers between complex genomic analysis and data scientists who want intuitive access to genomic workflow and data set, use a functional component called *AppCenter catalog*. This component serves as a catalog of pre-built and pre-validated application template and workflow definitions so that users can easily start the job or pipelines directly from the portal.

Figure 2-7 shows the genomic pipeline in the AppCenter.

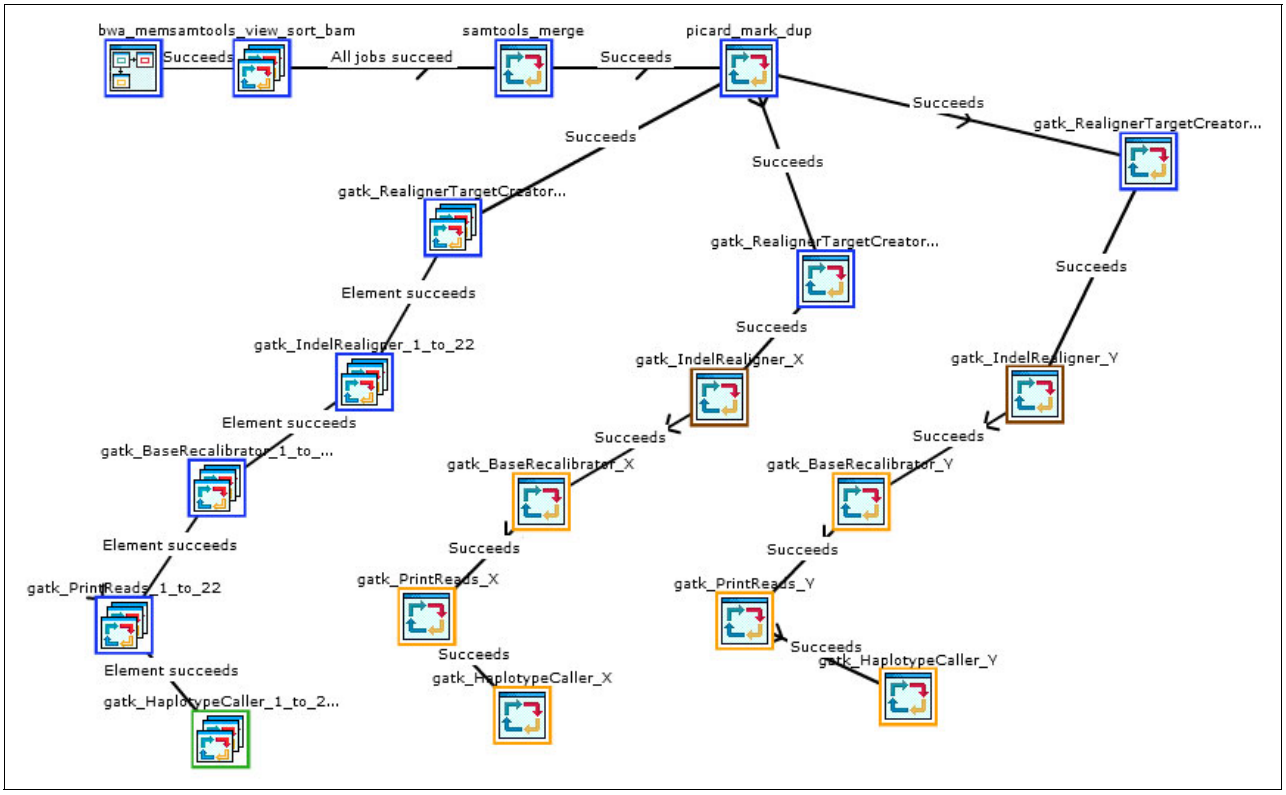


Figure 2-7 Genomic pipeline in the AppCenter

An end-to-end genomic pipeline can be started and visualized from the AppCenter catalog portal. Figure 2-7 shows these components in the flow:

- Arrival of data automatically triggers the pipeline into action.
- A dynamic subflow uses BWA for sequence alignment.
- SAMtools performs post-processing in a job array.
- BAM file recalibration.
- GATK performs variant calling.

The AppCenter catalog can be configured with a cloud data browser for user to manage the data needed for genomic computing. Within this portal-based browser, users can locate genomic data by browsing or searching all files/directories on remote or local storage servers (Datahub). Wherever the files are located, only two more clicks are needed to append the files for starting the job. With the data browser, users can mark/unmark a file directory as favorite, so it can be found easily and quickly. A useful directory to mark as favorite for a genomics computing user can be the one storing a commonly accessed reference data set.

Finally, the data browser can also facilitate data transfer because users can upload multiple files at the same time by dragging them from the browser desktop to the current remote directory.

Monitoring

The AppCenter Monitoring provides a portal-based dashboard that provides comprehensive workload monitoring, reporting, and management. Unlike other monitoring tools that focus on just one facet of system monitoring, this AppCenter function provides a complete, integrated monitoring facility for workloads. With the diverse profiles of genomic applications (for example, large memory, parallel, or single-threaded), the ability to track and report usage information at a job/application level can help improve application efficiency in terms of using computer CPU, memory, and storage I/O.

2.2 Hardware and components

The servers that are described in this architecture are IBM Power Systems S812L (8247-21L) and the IBM Power Systems S822L (8247-22L) open technology servers. These servers use the POWER8 processor technology designed to deliver performance, scalability, reliability, and manageability for demanding workloads.

These Power Systems servers are optimized for Linux to deliver new solutions and faster services, with higher quality. These servers are also optimized for big data and analytics as well to provide the ideal foundation for scale-out data and cloud environments. You can gain faster insights with the IBM POWER8 processor and smart acceleration that is enabled by the Coherent Accelerator Processor Interface (CAPI) technology.

Designed to empower the ecosystem of open source development, these servers also support the Linux operating system.

Note: For more technical details about these servers, see the *IBM Power Systems S812L and S822L Technical Overview and Introduction*, REDP-5098 at the following website:

`http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/redp5098.html`

2.2.1 IBM scale-out servers

This section provides details about the IBM scale-out servers.

IBM Power Systems S812L

The Power S812L server is a powerful one socket server. It includes one POWER8 dual chip module (DCM) processor that offers 3.42 GHz and 3.02 GHz performance with 10 or 12 fully activated cores, and up to 512 GB of memory.

Figure 2-8 shows the front view of a Power S812L.

Figure 2-8 Front view of a Power S812L

Figure 2-9 shows the rear view of a Power S812L.

Figure 2-9 Rear view of a Power S812L

IBM Power Systems S822L

The Power S822L (8247-22L) server is a powerful two socket server. It has POWER8 DCM processor modules that offer 4.15 GHz, 3.42 GHz, and 3.02 GHz performance with 16, 20, or 24 fully activated cores and up to 1 TB of memory.

Figure 2-10 shows the front view of a Power S822L.

Figure 2-10 Front view of a Power S822L

Figure 2-11 shows the rear view of a Power S822L.

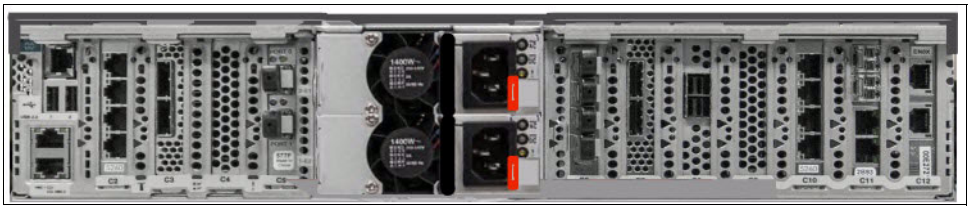


Figure 2-11 Rear view of a Power S822L

Processor modules

A maximum of two processors with eight processor cores or two processors with 10 processor cores or 12 processor cores is allowed. All processor cores must be activated. The following quantities of processing activation entitlements are allowed:

- Two 8-core 4.1 GHz processors require that 16 processor activation codes be ordered. A maximum of 16 processor activations is allowed.
- Two 10-core, 3.42 GHz processors require that 20 processor activation codes be ordered. A maximum of 20 processor activations is allowed.
- Two 12-core, 3.02 GHz processors require that 24 processor activation codes be ordered. A maximum of 24 processor activations is allowed.

System memory

A minimum of 32 GB of memory is required on the Power S822L system. Memory upgrades require memory pairs. Base memory is two 16 GB, 1600 MHz DDR3 memory modules.

Recommend configurations

Table 2-1 shows the recommended configuration for the S812L.

Table 2-1 Recommended configuration for S812L

Component	Value
Processors	10
Memory	32 GB
Power supplies	2
Network I/O adapters	2x 10 GbE RoCE SFP+ Adapter and 1 Gigabit Ethernet (UTP) 4 Port Adapter PCIE-4x/Short

Table 2-2 shows the recommended configuration for the S822L.

Table 2-2 Recommended configuration for S822L

Component	Value
Processors	20
Memory	128 GB
Power supplies	2
Network I/O adapters	2x 10 GbE RoCE SFP+ Adapter and 1 Gigabit Ethernet (UTP) 4 Port Adapter PCIE-4x/Short

Adapter slots

The adapter slots include one x8 PCIe slot, which must contain a 4-port 1 Gb Ethernet LAN available for client use:

- S812L: Six PCIe Gen3 slots with concurrent maintenance: Two x16 plus four PCIe Gen3 x 8
- S822L: Nine PCIe Gen3 slots with concurrent maintenance: Four x16 plus five PCIe Gen3 x 8 Two CAPI adapters per processor module

Network adapters and connections

This section describes the network adapters and connections.

PCIe2 2-Port 10 GbE RoCE SFP+ Adapter

The PCIe2 LP 2-Port 10 GbE RoCE SFP+ adapter is a low-profile, PCIe generation-2 (PCIe2), dual port, 10-Gigabit Ethernet (GbE) adapter with a PCIe 2.0 host bus interface. The adapter delivers high-bandwidth and low-latency 10 GbE connectivity. The adapter is optimized for high-transaction databases, cloud computing, virtualization, storage, and other data center applications. The adapter improves network performance by increasing available bandwidth to the central processing unit (CPU) and providing enhanced performance.

The two 10 Gb small form-factor pluggable (SFP+) transceiver ports are used for connectivity with other servers or switches in the network. Each SFP+ port provides Ethernet connectivity with a nominal data rate of 10 gigabits per second (Gbps), and uses the SFP+ copper twinaxial cables up to 5 m (16.4 ft.) in length. The link aggregation and failover features of the adapter are ideal for critical network applications that require redundancy and high availability.

Figure 2-12 shows the PCIe2 2-Port 10 GbE RoCE SFP+ Adapter.

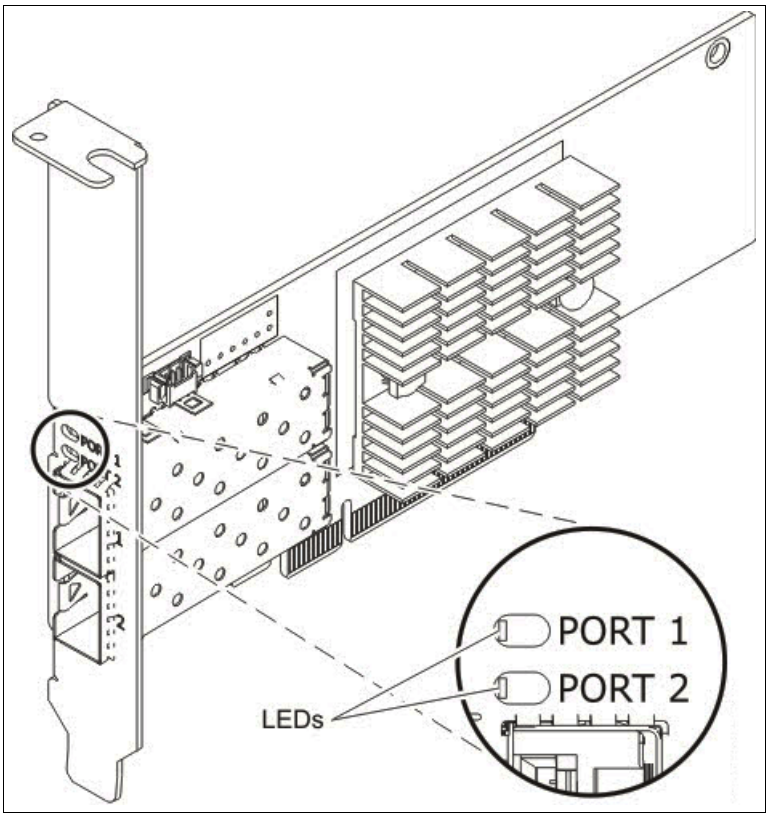


Figure 2-12 PCIe2 2-Port 10 GbE RoCE SFP+ adapter

2-Port 56 Gb FDR PCIe x16

The PCIe3 low profile 2-port 56 Gb FDR InfiniBand adapter x16 provides high-speed connectivity with other servers or InfiniBand switches. Each port maximum of 56 Gb assumes that no other system or switch bottlenecks are present. The x16 adapter allows full bandwidth in a PCIe Gen3 slot.

Figure 2-13 shows the 2-Port 56 Gb FDR PCIe x16.

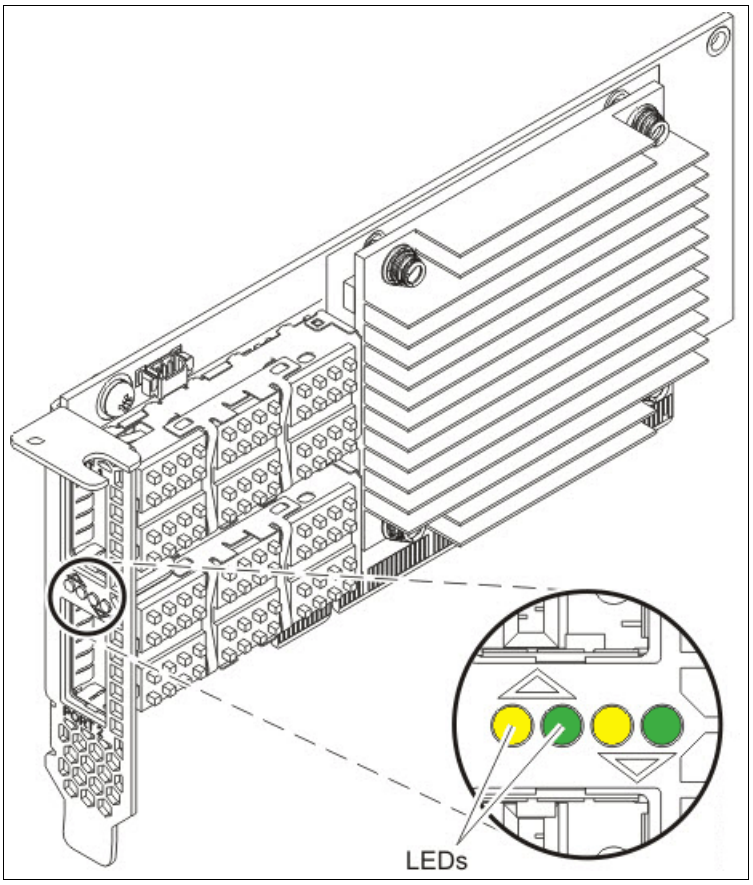


Figure 2-13 Port 56 Gb FDR PCIe x16

The LEDs indicate these statuses:

- A lit solid amber LED (left LED) indicates a good physical link.
- A flashing amber LED (left LED) indicates a problem with the physical link.
- If neither the amber LED (left LED) nor the green LED (right LED) is lit, then the physical link has not been established.
- The amber LED (left LED) turns off when the logical link is established.
- A lit solid green LED (right LED) indicates a valid logical (data activity) link without data transfer.
- A flashing green LED (right LED) indicates a valid logical link with data transfer.

If only the amber LED (left LED) is lit and the green LED (right LED) is off, then the logical link has not been established.

PCIe2 4-port 1 GbE adapter

The PCIe2 LP 4-port 1 GbE adapter is a full-height, generation-2 PCIe adapter. This adapter provides four 1-Gb Ethernet ports that can be configured to run at 1000 megabits per second (Mbps) (or 1 gigabit per second (Gbps)), 100 Mbps, or 10 Mbps. The adapter connects to a network that uses unshielded twisted pair (UTP) cable for distances of up to 100 meters (328.08 feet). The adapter supports jumbo frames when running at the 1000 Mbps speed.

Figure 2-14 shows the PCIe2 4-port 1 GbE adapter.

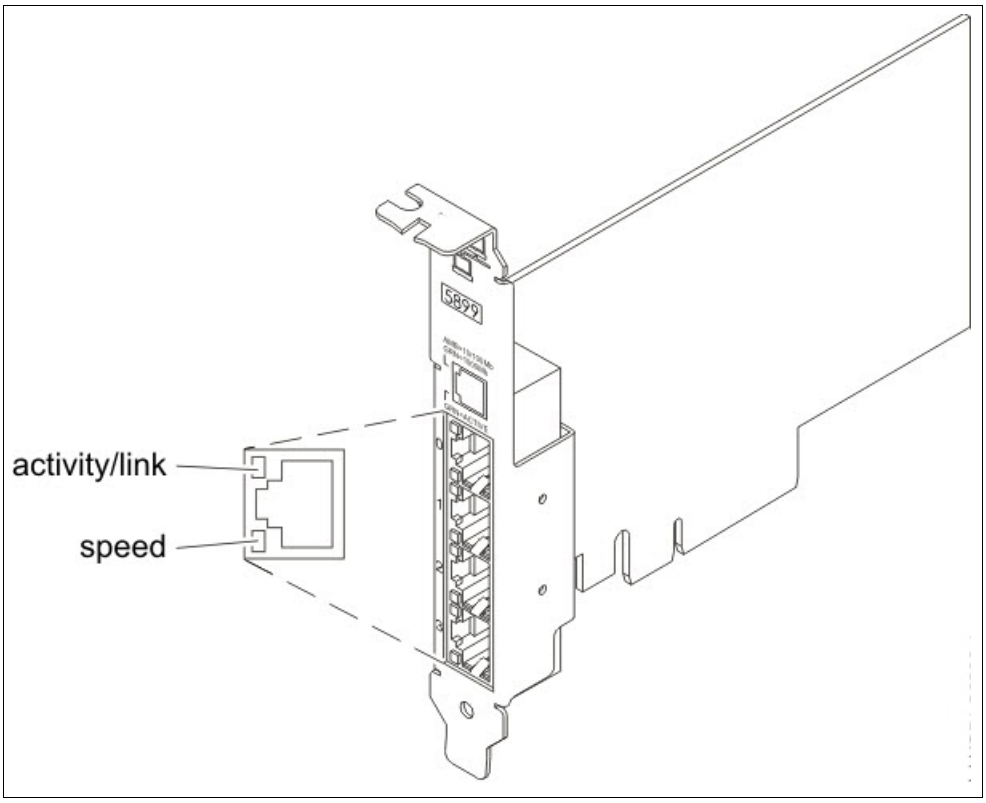


Figure 2-14 PCIe2 4-port 1 GbE adapter

Table 2-3 shows the recommended adapters.

Table 2-3 Recommended adapter configurations

Adapters	Network
Ethernet adapter 1 GB	Campus network
Ethernet adapter 1 GB	Management network
Ethernet adapter 1 GB	Provisioning network
Ethernet adapter 1 GB	Service network
InfiniBand adapter or Ethernet adapter 10 GB	Data network

Note: For more information, see the following website:

http://www.ibm.com/support/knowledgecenter/8247-22L/p8hcd/p8hcd_83x_8rx_kickoff.htm?lang=en

Coherent Accelerator Processor Interface for POWER8 systems

Coherent Accelerator Processor Interface (CAPI) is available only on POWER8 systems. Through a unique hardware investment and operating-system additions, CAPI on POWER8 processors provides clients with a customizable acceleration engine that runs as a peer to the POWER8 cores. It accesses memory by using the same programming methods and virtual address space as the application that calls it.

The CAPI solution provides much faster data preparation time when compared to a traditional, I/O attached field-programmable gate array (FPGA) by removing the need for a device driver and its code stack. CAPI-attached devices can perform functions such as pointer chasing that an I/O attached device cannot do.

Flexible service processor

The flexible service processor (FSP) is an advanced service processor that provides function to start the system chip sets, boot to a hypervisor runtime, interface to the management module, and provide runtime support for advanced reliability, availability, and serviceability (RAS) functions.

FSP is an extra layer in the hardware that closely monitors the other hardware components. During an incident, it notifies the system administrator or vendor to work proactively before the failure to avoid disruption in the service.

Connecting the FSP to the HMC

Figure 2-15 shows the Ethernet connection from the IBM Power 812L and 822L to the Hardware Management Console (HMC).

Note: The hardware allows two HMC connections (P1-T1 and P1-T2) for redundancy.

Figure 2-15 shows the HMC ports 1 (P1-T1) and 2 (P1-T2).

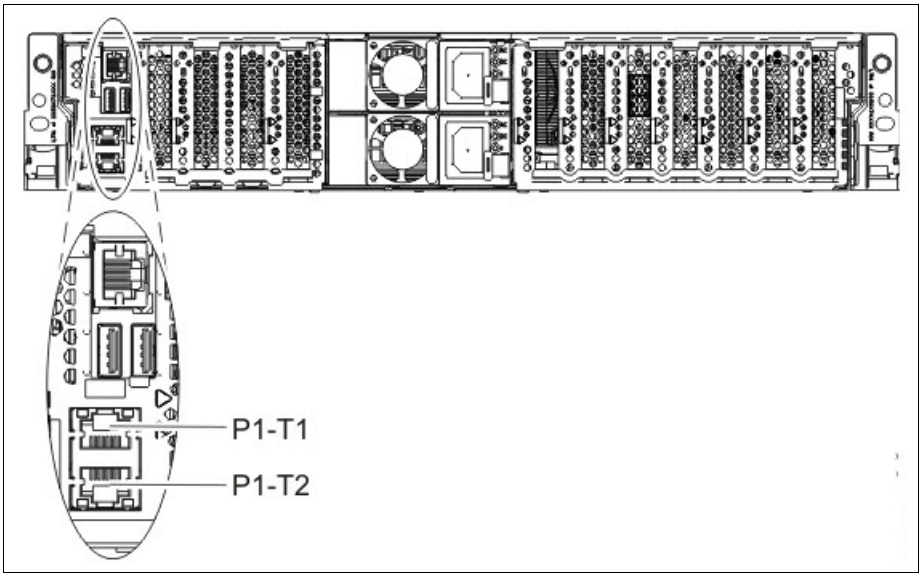


Figure 2-15 HMC ports 1 and 2 of the Power S812L and S822L

2.2.2 Hardware Management Console

The Hardware Management Console (HMC) allows you to configure and manage servers. One HMC can manage multiple servers, and a dual HMC can provide redundant support by managing the same system. To ensure consistent functionality, each HMC is preinstalled with the HMC Licensed Machine Code Version 8, Release 1.

Figure 2-16 shows the front view of the HMC 7042-CR8.

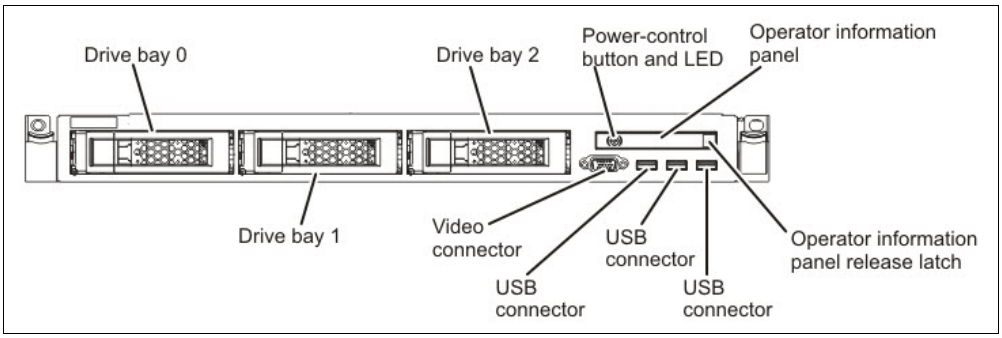


Figure 2-16 Front view of the HMC

Figure 2-17 shows the rear view of the HMC.

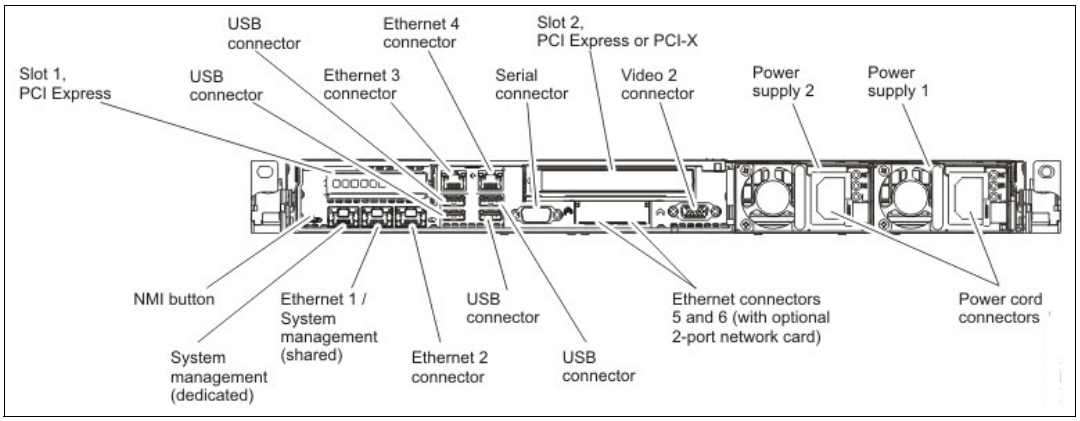


Figure 2-17 Rear view of the HMC

Figure 2-18 shows the HMC Web GUI.

Figure 2-18 HMC Web GUI

HMC firmware level recommendation

IBM support requires that the HMC firmware level must be equal to or later than the firmware level that the system is managing.

Note: For this architecture, the HMC version must be *V8R8.2.0*. If you need to update the firmware level, see the following IBM Support Portal:

`http://www.ibm.com/support/fixcentral/`

2.2.3 Storage enclosure and components

To support I/O-intensive applications, this architecture uses a two-node storage server (gss01 and gss02), which is used to provide a shared file system (specifically IBM Spectrum Scale) to the rest of the systems.

The storage server provides secure storage to IBM Platform LSF® users to maintain user profiles, and to share compute model data files. This configuration allows applications that are running on the cluster within high speed and low latency interconnect to access any required data, which significantly improves application performance.

To create a high available Platform Cluster Manager environment, shared storage is required to share user home directories and system working directories. All shared file systems must be accessible by the provisioning network for Platform Cluster Manager management nodes and all compute nodes. The Platform LSF working directory must be available through a

shared file system in the master and master candidate hosts. Platform Application Center DB data files are stored in the shared file system, so the database on the secondary master candidate node uses the same data on the shared file system when it is started. The Platform Report Track Monitor (RTM) DB data files are also stored in the shared drive.

Each storage node is an IBM Power S822L server with a Linux and a file system server. The IBM Elastic Storage Server (ESS) as the storage subsystem is compatible with this architecture.

Figure 2-19 shows the storage nodes gss01 and gss02.

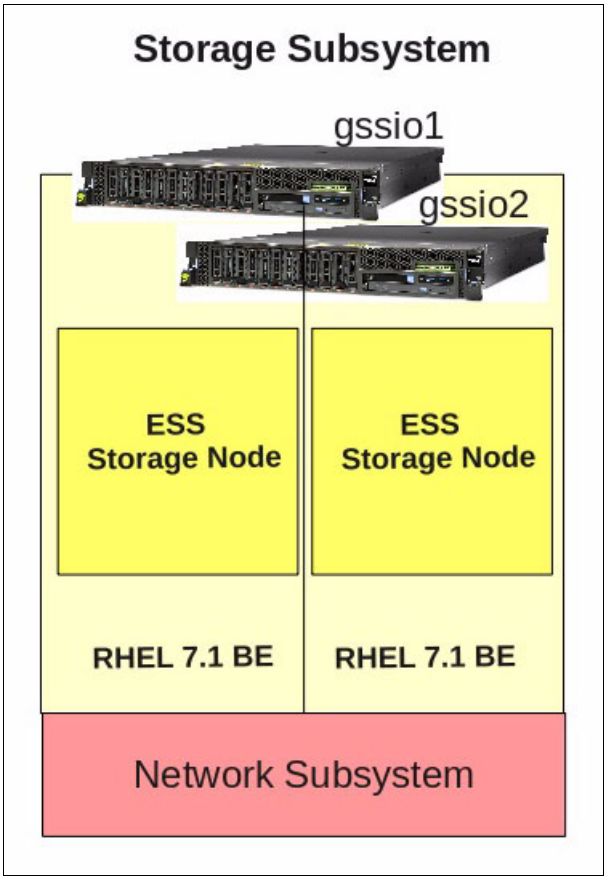


Figure 2-19 Storage nodes gss01 and gss02

IBM Elastic Storage Server 5146-GL2

The IBM Elastic Storage Server (ESS) is a modern implementation of software-defined storage built on the IBM Spectrum Scale. This technology combines the CPU and I/O capability of the IBM POWER8 architecture and matches it with 2U and 4U storage enclosures. Table 2-4 shows the storage nodes and management console.

Table 2-4 Storage nodes and management console

Hostname	Description	Hardware
ems1	Platform and storage management console	S812L (8247-21L)
gss01	Elastic Storage Server node	S822L (8247-22L)
gss02	Elastic Storage Server node	S822L (8247-22L)

This architecture allows the IBM Spectrum Scale RAID software capability to actively manage all RAID functions that were formerly accomplished by a hardware disk controller.

Newly developed RAID techniques from IBM use this CPU and I/O power to help overcome the limitations of current disk drive technology and simplify your transition to a multitier storage architecture that employs solid-state flash technology and robotic tape libraries.

In addition to the technological advancements available with the Elastic Storage Server, it can also address other data issues found in many businesses. For example, each department or division in your organization can have different storage needs, which can result in a costly duplication of hardware resources. The resulting islands of information can hold valuable insights that cannot be accessed by the other departments. By consolidating storage requirements across your organization onto the Elastic Storage Server, you can reduce inefficiency and acquisition costs while simplifying management and improving data protection.

Elastic Storage Server has these capabilities, among others:

- Software RAID: Spectrum Scale RAID runs IBM disks in a dual-ported storage enclosure that does not require external RAID storage controllers or other custom hardware RAID acceleration.
- Declustering: Spectrum Scale RAID distributes client data, redundancy information, and spare space uniformly across all disks of a JBOD. This distribution reduces the rebuild or disk failure recovery process effort compared to conventional RAID. Critical rebuilds of failed full multi-terabyte drives can be accomplished in minutes rather than the hours or even days needed by traditional RAID technology.
- Data redundancy: Spectrum Scale RAID supports highly reliable 2-fault-tolerant and 3-fault-tolerant Reed-Solomon-based parity codes and 3-way and 4-way replication.
- Large cache: Using a combination of internal and external flash devices along with large memory cache in the Power Systems server, the Elastic Storage Server is better able to mask the inefficiencies and long latency times of nearline SAS drives. However, it still uses the high density of the drives themselves.
- Graphical user interface (GUI): The intuitive GUI allows management and monitoring of the system, both locally and remotely. For more information, see 2.3.2, “IBM Elastic Storage Server GUI” on page 35.
- Scalability: As server configurations are added to an installed configuration, the capacity, bandwidth, performance, and the single name space all grow. This capability means that installations can start small, and grow as data needs expand. Each storage node has an installation of Spectrum Scale to accomplish its role.

Figure 2-20 shows the storage nodes and the DCS3700 expansion drawers.

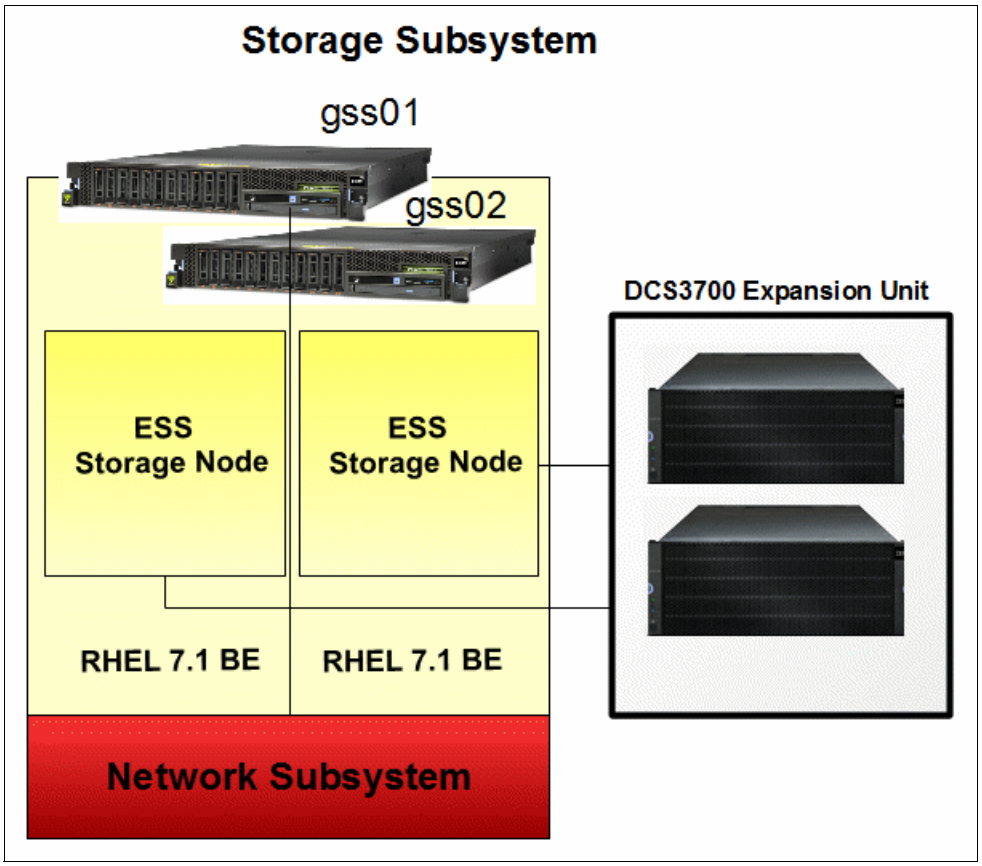


Figure 2-20 The storage nodes and DCS3700 expansion drawers

DCS3700 Expansion Unit 1818-80E

The IBM 1818 Model 80E is a DCS3700 Expansion Unit (Dual ESM) with 60 drive bays.

- The IBM System Storage® DCS3700 Storage System has these features:
  - Dual-active intelligent array controllers enabled with Turbo Performance
  - 6 Gb SAS host ports standard with optional 8 Gb Fibre Channel (FC) host ports
  - 60 SAS drive bays, with support for up to 180 drives with the attachment of two DCS3700 Expansion Units
  - Data replication options with IBM FlashCopy®, Volume Copy, and Remote Mirroring over FC
  - IBM DS Storage Manager for administrative and management activities
- The performance modules have the characteristics above and also feature:
  - Eight 8 Gb FC ports with optional 6 GB SAS, 10 Gb iSCSI, or additional 8 Gb FC host interface cards (HICs)
  - Support for up to 5 DCS3700 Expansion Units for up to 360 drives
  - Use of drives formatted for T10 PI enhanced data security
- The IBM System Storage DCS3700 Expansion Unit has these features:
  - Dual-active Environmental Services Modules (ESMs)
  - Sixty SAS drive bays

- 6 Gb SAS attachment to the DCS3700 Storage System or to another DCS3700 Expansion Unit

- The unit has these model features:
  - Slim 4U, 19-inch rack mount enclosure
  - Dual port, hot-swappable 2 TB and 3 TB SAS nearline disk drives
  - Dual port, hot-swappable 6 Gb SAS SSDs in 200 and 400 GB capacities
  - Dual port, hot-swappable 6 Gb SAS 10,000 rpm disk drives in 600 and 900 GB capacities
  - Redundant, hot-swappable hardware components
  - IBM installation and a one year warranty with 24 x 7 onsite repair

Figure 2-21 shows the front view of the DCS3700 Expansion Unit 1818-80E.

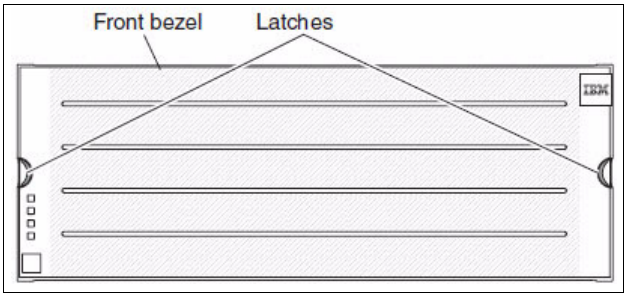


Figure 2-21 Front view of DCS3700 Expansion Unit 1818-80E

Figure 2-22 shows the front view of DCS3700 Expansion Unit drive drawers.

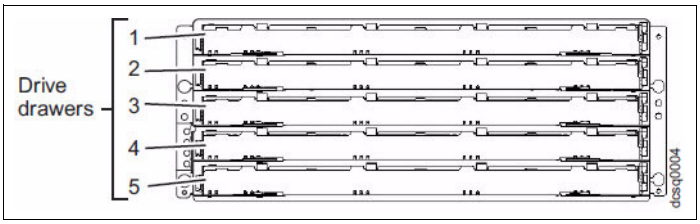


Figure 2-22 Front view of DCS3700 Expansion Unit drive drawers

Figure 2-23 shows the rear view of DCS3700 Expansion Unit 1818-80E.

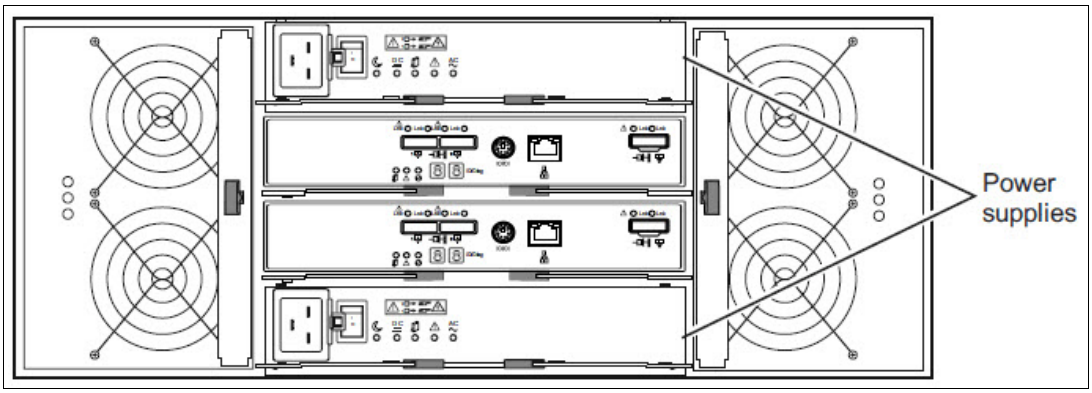


Figure 2-23 Rear view of DCS3700 Expansion Unit 1818-80E

2.3 Software and components

This section describes the software and components included in this reference architecture. Concerning applications, the intent is to support a wide variety of genomic application suites, including those that are widely available (for example, GATK, SAMtools, and BWA).

2.3.1 Operating systems and distributions

This solution was implemented by using the Linux distributions Red Hat Enterprise Linux 7.1 Big Endian (BE) and Ubuntu 14.04.2 Little Endian (LE) with PowerNV (baremetal).

IBM PowerVM

PowerVM for Linux provides virtualization technologies for Linux running on Power Systems servers by virtualizing processor, memory, storage, and I/O resources for client partitions that enable increased asset utilization, enhanced infrastructure flexibility, and reduced cost.

Figure 2-24 shows the stack diagram for PowerVM.

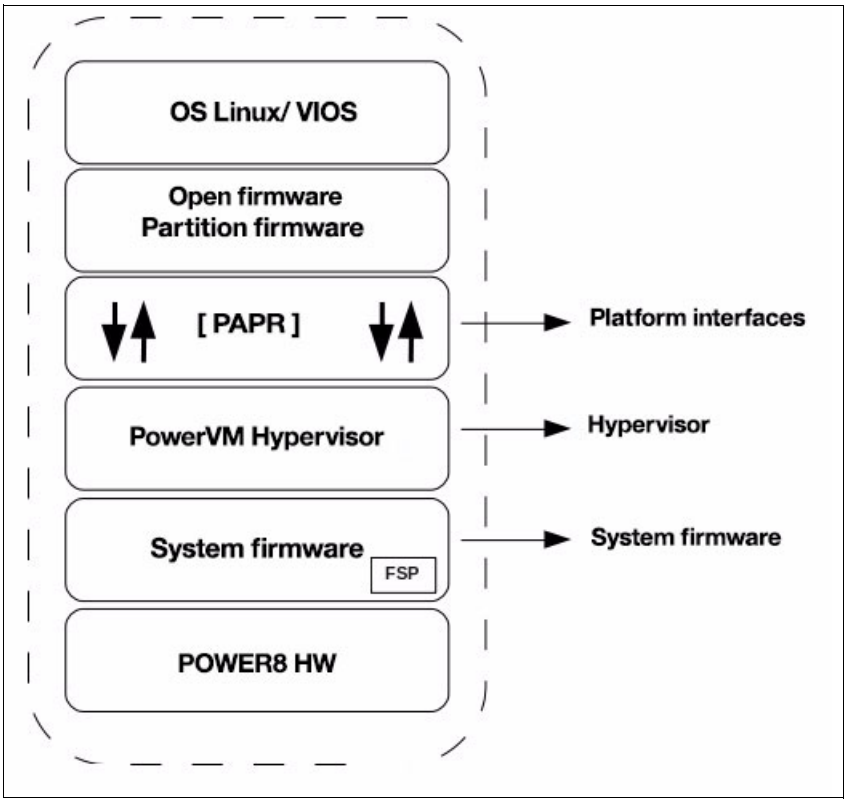


Figure 2-24 Stack diagram for PowerVM

IBM PowerKVM

PowerKVM is the Linux host kernel that runs in the hypervisor mode. PowerKVM with the help of an open source KVM system, allows you to run multiple Linux guests seamlessly in the supervisor mode. Host kernel has access to all memory and all system resources. All the guests running on top of PowerKVM are paravirtualized using the PAPR interface. The PowerKVM host runs in the hypervisor mode while the guest kernel runs in the supervisor mode. The special firmware, Open Power Abstraction Layer (OPAL), provides full access to

the hypervisor mode, and guests can run at full hardware speed without emulating any privileged instructions.

Figure 2-25 shows the stack diagram for PowerKVM.

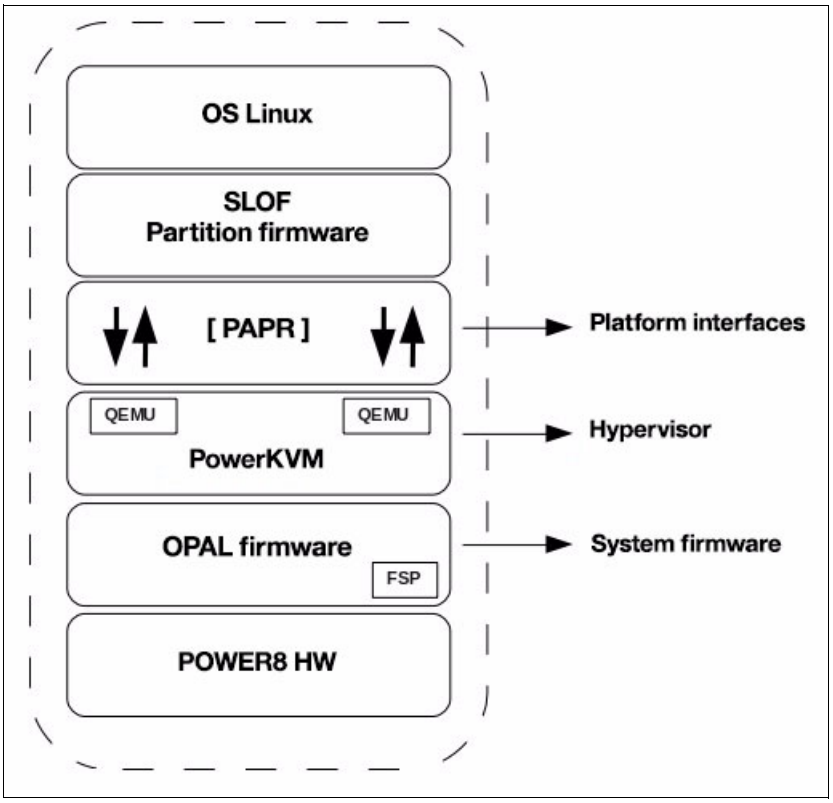


Figure 2-25 Stack diagram for PowerKVM

Open Power Abstraction Layer

Open Power Abstraction Layer (OPAL) is a small layer of firmware designed to allow a Linux operating system to run directly on the POWER8 Systems server. OPAL is part of the firmware that interacts with the hardware and exposes it to the PowerKVM hypervisor. Because it runs directly on the system, the operating system can be a KVM host. Table 2-5 shows the hosts, hardware, and subsystems.

Table 2-5 Show hosts, hardware, and subsystem

Hostname	Operating system level	Subsystem	Hardware
ems1	Red Hat Enterprise Linux 7.1 BE	Management Subsystem	8247-21L
mn01	Ubuntu 14.04.2 LE	Management Subsystem	8247-22L
mn02	Ubuntu 14.04.2 LE	Management Subsystem	8247-22L
cn01	Ubuntu 14.04.2 LE	Compute Subsystem	8247-22L
cn02	Ubuntu 14.04.2 LE	Compute Subsystem	8247-22L
cn03	Ubuntu 14.04.2 LE	Compute Subsystem	8247-22L
cn04	Ubuntu 14.04.2 LE	Compute Subsystem	8247-22L

Hostname	Operating system level	Subsystem	Hardware
gss01	Red Hat Enterprise Linux 7.1 BE	Storage Subsystem	8247-22L
gss02	Red Hat Enterprise Linux 7.1 BE	Storage Subsystem	8247-22L

Note: The recommended Linux distribution for a particular server is always the latest level distribution that is optimized for the server. The listed distributions are the operating system versions that are supported for the specific hardware. For information about the product lifecycles for the Linux distributions, see the distribution's support site:

- Red Hat Enterprise Linux: Red Hat Enterprise Linux Lifecycle

  https://access.redhat.com/site/support/policy/updates/errata/
- Ubuntu: Ubuntu Release Lifecycle

  http://www.ubuntu.com/info/release-end-of-life

Bare metal or PowerNV (Power non-virtualized)

Bare metal, or non-virtualized, servers support various adapters. Bare metal supported operating systems include Ubuntu 14.04.2, and Ubuntu 14.10 and later.

See Table 2-6, Table 2-7, Table 2-8 on page 35, and Table 2-9 on page 35 to determine whether your I/O adapter is supported on systems running in bare metal mode.

Table 2-6 Supported InfiniBand adapters

InfiniBand adapter name	Feature code	Customer card identification number (CCIN)	Operating system levels
PCIe3 LP 2-port 56 Gb FDR IB Adapter x16	EL3D	2CE7	Ubuntu 14.04.2 with MLNX_OFED Ubuntu 15.04, or later
PCIe3 2-port 56 Gb FDR IB Adapter x16	EL50	2CE7	Ubuntu 14.04.2 with MLNX_OFED Ubuntu 15.04, or later

Table 2-7 shows the supported local area network (LAN) adapters.

Table 2-7 Supported local area network (LAN) adapters

LAN adapter name	Feature code	Customer card identification number (CCIN)	Operating system levels
PCIe2 LP 4-Port (10 Gb+1 GbE) SR+RJ45 Adapter	EN0T	2CC3	Ubuntu 14.04.1, or later Ubuntu 14.10, or later
PCIe2 LP 4-Port (10 Gb+1 GbE) SR+RJ45 Adapter	EN0S	2CC3	Ubuntu 14.04.1, or later Ubuntu 14.10, or later
PCIe2 LP 4-port (10 Gb+1 GbE) Copper SFP+RJ45 Adapter	EN0V	2CC3	Ubuntu 14.04.1, or later Ubuntu 14.10, or later

LAN adapter name	Feature code	Customer card identification number (CCIN)	Operating system levels
PCIe2 LP 2-port 10 GbE BaseT RJ45 Adapter	EL3Z	2CC4	Ubuntu 14.04.1, or later Ubuntu 14.10, or later
PCIe2 LP 2-port 10 GbE BaseT RJ45 Adapter	EL55	2CC4	Ubuntu 14.04.1, or later Ubuntu 14.10, or later

Table 2-8 shows the supported storage backplanes for serial attached SCSI (SAS).

Table 2-8 Supported storage backplanes for serial attached SCSI (SAS)

Storage controller name (serial attached SCSI)	Feature code	Customer card identification number (CCIN)	Operating system levels
PCIe3 x8 SAS RAID internal adapter 6Gb	EL3T	57D7	Ubuntu 14.04.1, or later Ubuntu 14.10, or later
PCIe3 x8 SAS RAID internal adapter 6Gb	EJ0N	57D7	Ubuntu 14.04.1, or later Ubuntu 14.10, or later

Table 2-9 shows the supported accelerator adapters.

Table 2-9 Supported accelerator adapters

Accelerator adapters	Feature code	Customer card identification number (CCIN)	Operating system levels
PCIe3 LP CAPI Accelerator Adapter	EJ16	N/A	Ubuntu 14.10, or later
PCIe gen3 x16 GPU Adapter	EC47	2CE8	Ubuntu 14.04.2, with CUDA 7.0, available from NVIDIA Ubuntu 14.10, with CUDA 7.0, available from NVIDIA
PCIe gen3 x16 GPU Adapter	EC4B	2CE9	Ubuntu 14.04.2, with CUDA 7.0, available from NVIDIA

2.3.2 IBM Elastic Storage Server GUI

The IBM Elastic Storage Server (ESS) graphical user interface (GUI) is an administrative interface that can be used in addition to the command-line interface (CLI), allowing users to manage and monitor the ESS. This GUI is a web portal that is installed on ems1 server, and can be accessed by using a web browser.

Figure 2-26 shows the ESS GUI access from a web browser.

Figure 2-26 ESS GUI access from a browser

Figure 2-27 shows the dashboard from the ESS GUI.

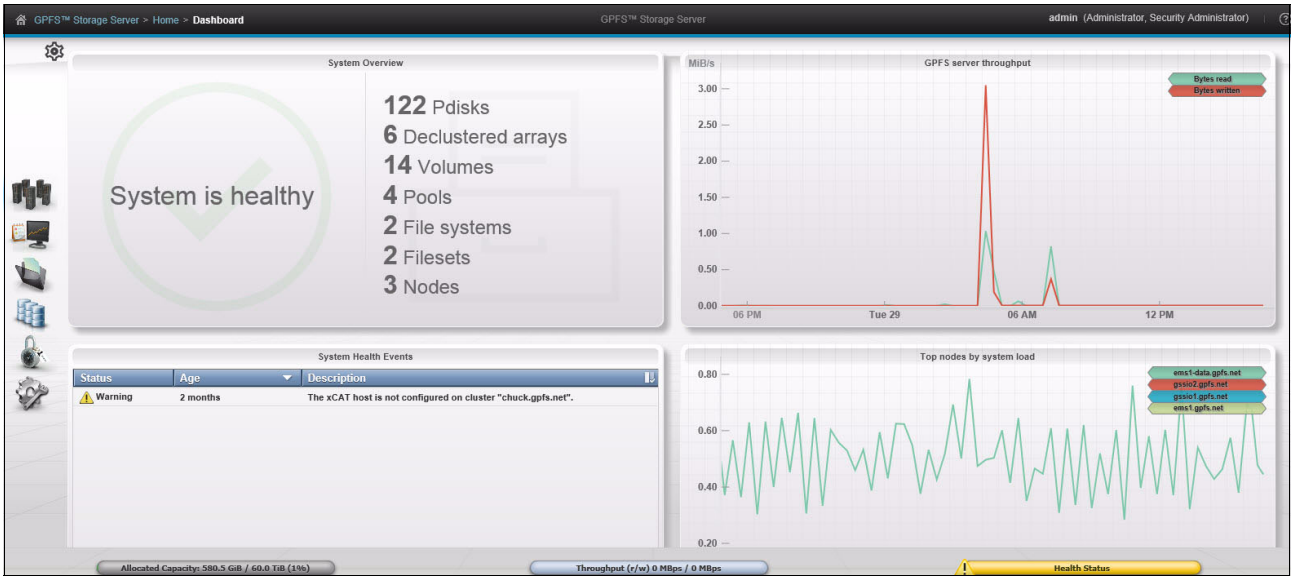


Figure 2-27 Dashboard from the ESS GUI

Figure 2-28 shows a file system created from the ESS GUI.

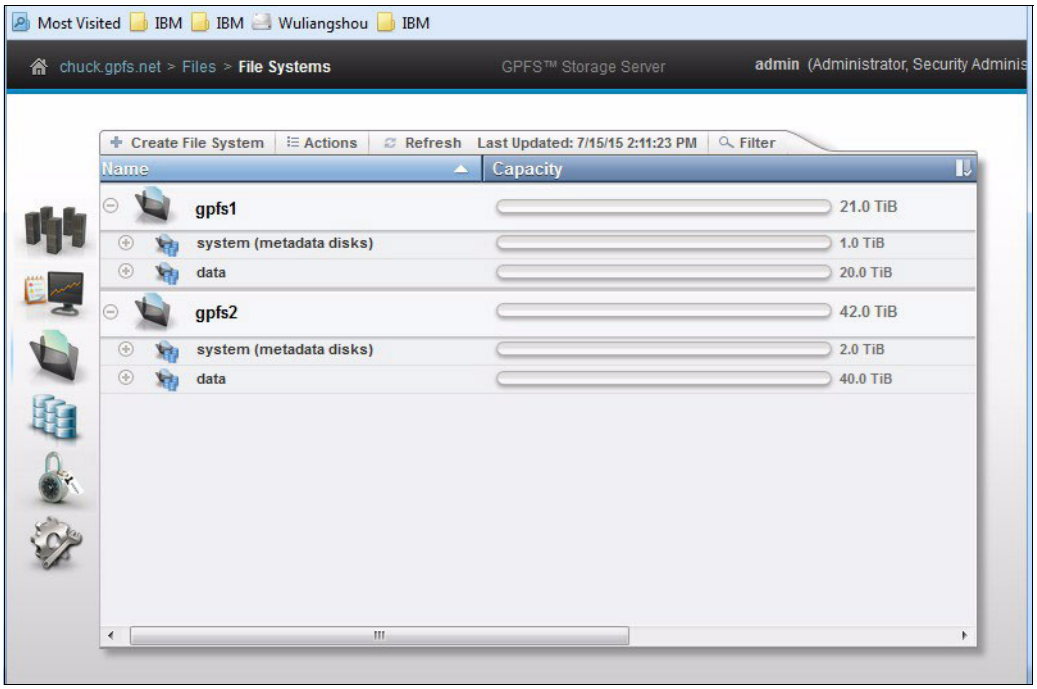


Figure 2-28 File system created from the ESS GUI

2.3.3 Applications

Applications consist of the software elements that accomplish the genomic processing directly. Within this architecture, these can include any applications or application suites that are relevant to the particular installation (this architecture does not prescribe any specific applications or application suites). These can be individual applications or application suites (for example, GATK) which accomplish operations that can be combined to form more complex flows.

This architecture is composed of the following software:

- IBM Platform Cluster Manager provisions the storage nodes and the management nodes. This is typically a one-time operation. Provisioning of the compute nodes usually is not a frequent operation because the compute nodes can normally be provisioned in a static way that supports a wide variety of applications (jobs and flows).
- IBM Platform Application Center as the specific application center to be used, but alternative components (for example, Galaxy and Lab7) can be used for this function.
- IBM Platform LSF as the specific workload manager to be used for this function.
- IBM Platform Process Manager as the specific workflow engine to be used, but alternative components (for example, Galaxy and Lab7) can be used for this function.
- IBM Platform Report Track Monitor (RTM) as the specific job and flow monitor to be used for this function.

2.3.4 Open source tools

This section describes the open source tools available to help life sciences (genomics) workloads.

BWA

The Burrows-Wheeler Alignment Tool (BWA) is 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 at no extra cost, 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

The Sequence Alignment and Map Tool (SAMtools) provides various utilities for manipulating alignments in the SAM format, including sorting, merging, indexing, and generating alignments in a per-position format.

Picard

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

The Genome Analysis Toolkit open source version (GATK) software package 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.

2.3.5 Platform Cluster Manager

IBM Platform Cluster Manager provides an easy way to use provisioning software that allows management or head nodes to install, configure, and monitor compute nodes. It is highly configurable and allows the administrator to provision site unique software with the use of *kits*. Kits are installation packages that contain customized software. This solution provides scripts to create kits that contain bioinformatic software.

The Platform Cluster Manager environment includes the following components:

- User self-service and administration portal.
- The management server, which is responsible for running the system services and managing provisioned clusters.
- The xCAT provisioning engine, which provisions clusters with physical machines. The provisioning engine is responsible for managing the physical machines that make up provisioned clusters.
- A database to store operational data. You can specify Platform Cluster Manager to install the MariaDB, or you can use an existing MySQL or Oracle database.
- Physical machines, which are the compute nodes within a cluster.

Figure 2-29 shows the Platform Cluster Manager Web GUI.

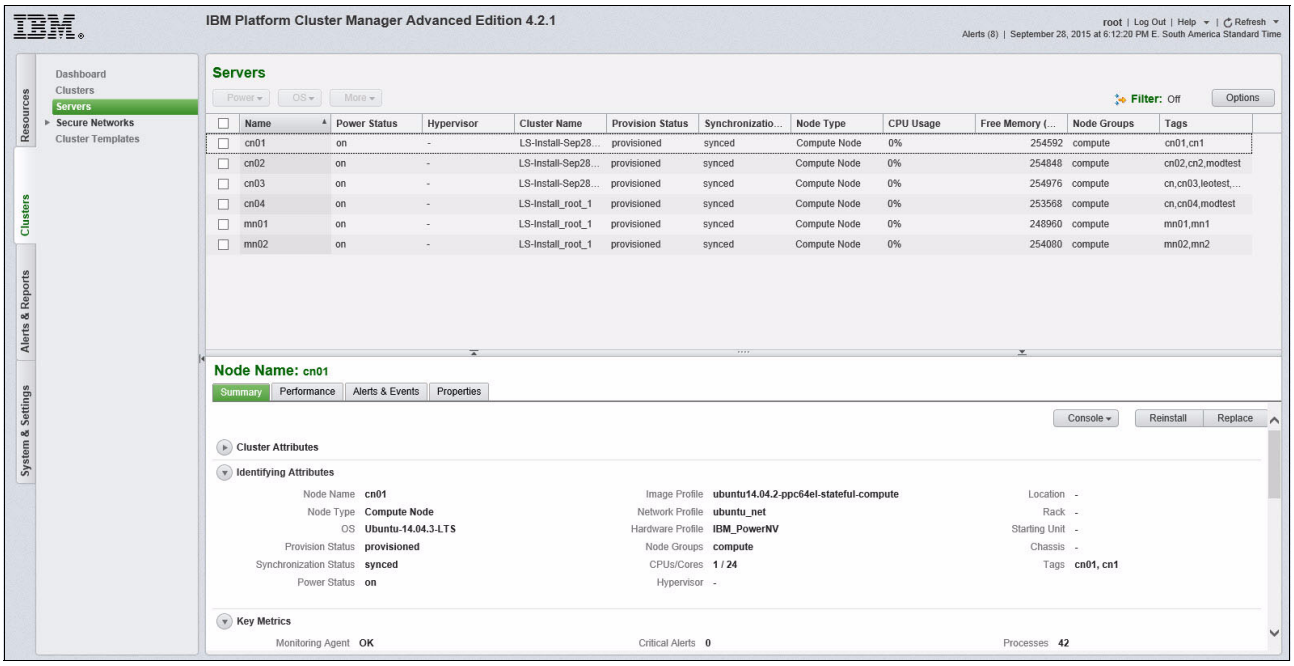


Figure 2-29 Platform Cluster Manager Web GUI

A Platform Cluster Manager template is provided for installing and configuring Platform LSF, Platform RTM, Platform Process Manager (PPM), and Platform Application Center onto a server. If the setup is done by using this Platform Cluster Manager template, failover for LSF and PPM is also configured. All software dependencies are resolved during the process.

Figure 2-30 shows the Platform Cluster Manager cluster template to provision the platform products.

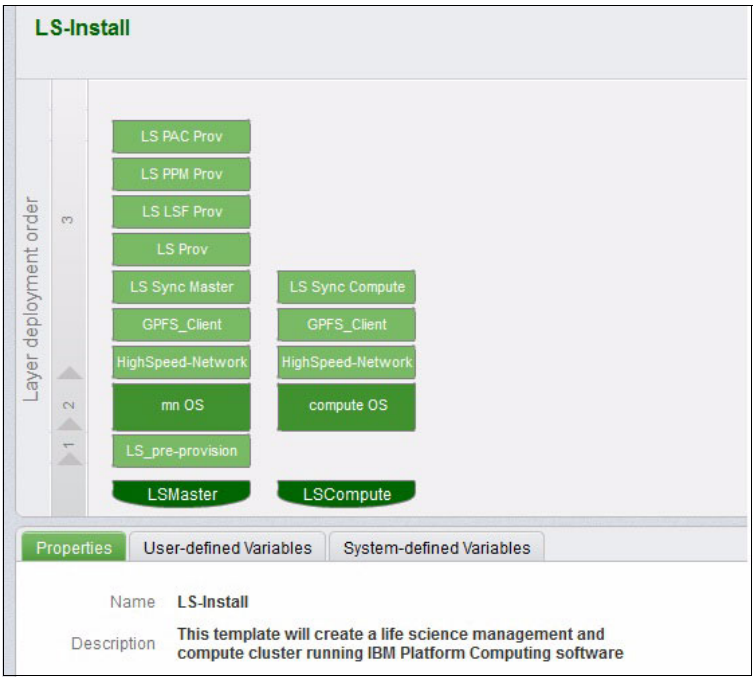


Figure 2-30 Platform Cluster Manager cluster template to provision the platform products

2.3.6 Platform Application Center

The Platform Application Center is a web-based GUI that integrates job submission and management as well as cluster management and reporting features. It simplifies and standardizes application submission. It provides fine-grain control over who has access to what information. Multiple users can run and manage flows and flow-related data through the portal. Running workflows can be monitored and managed by any job.

Color-coded workflow charts provide real-time visibility into wthe workflow progress. Users can drill into workflows to see details about subflows and individual job steps. Workflows can be published selectively and shared with other users.

As workflows evolve, version histories are maintained in IBM Platform Process Manager, making workflows easier to maintain. All data associated with workflows and individual job steps is conveniently accessible through the Platform Application Center.

Figure 2-31 shows Platform Application Center on mn01 and mn02.

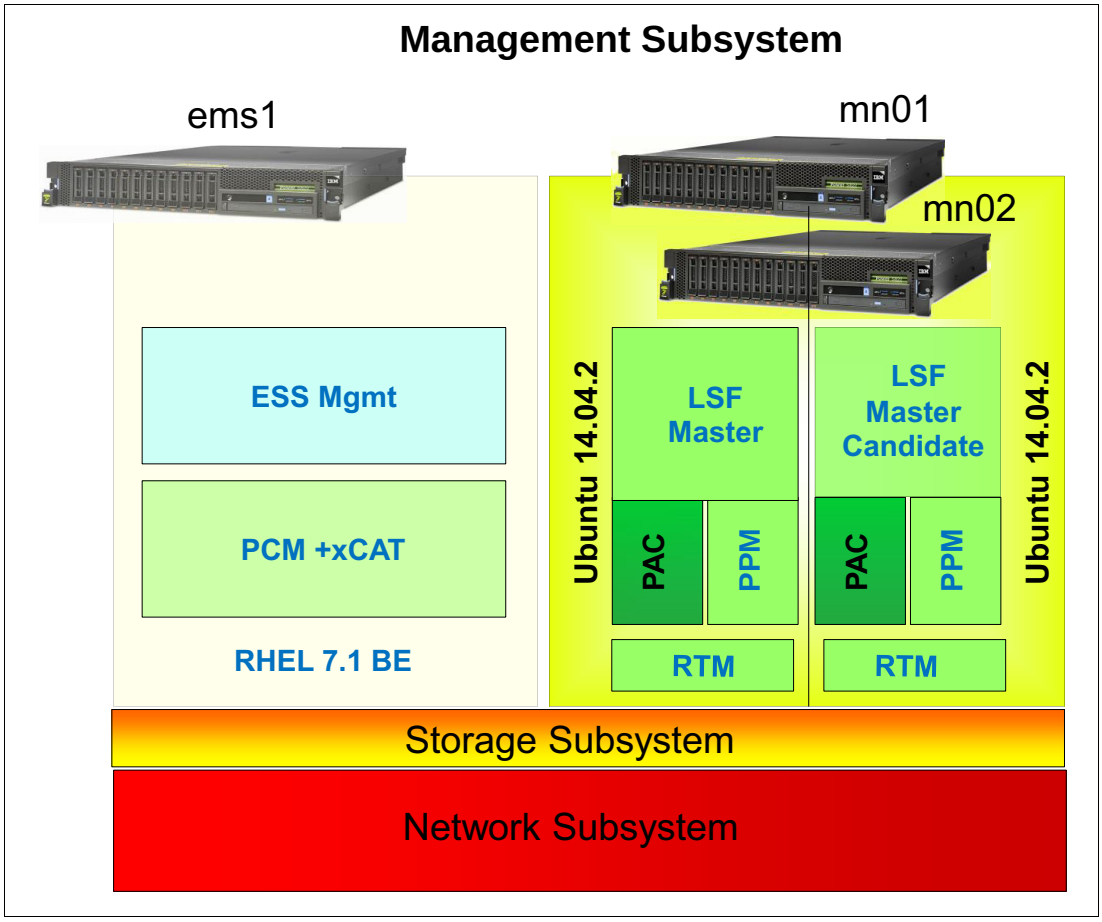


Figure 2-31 Platform Application Center solution architecture reference deployment

Figure 2-32 shows the Platform Application Center Web GUI.

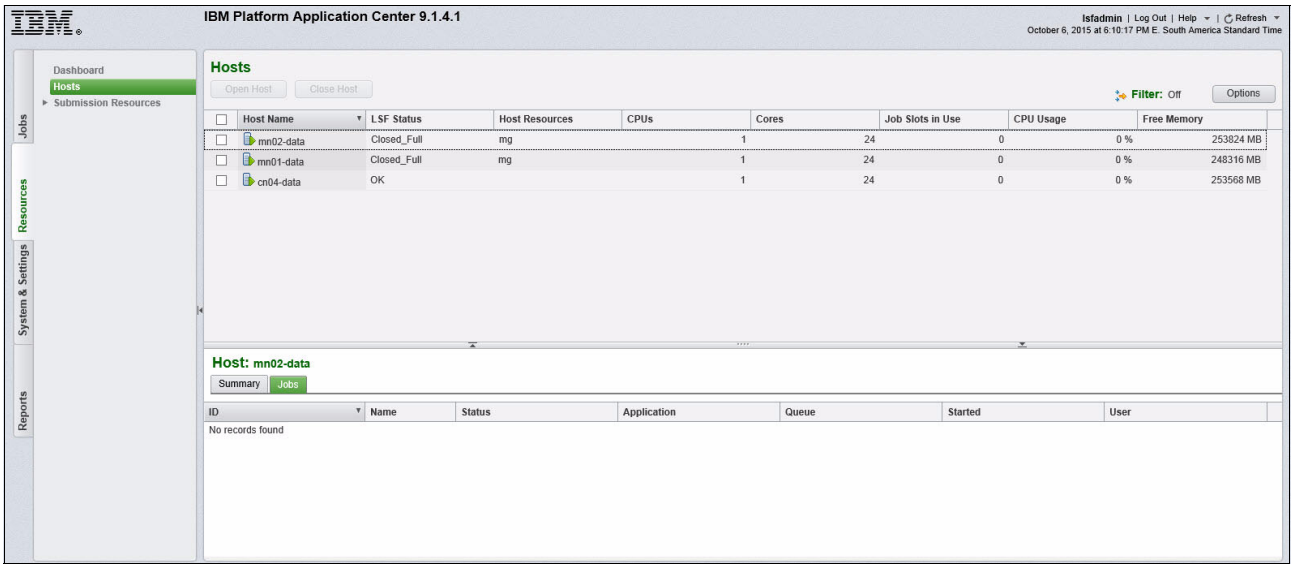


Figure 2-32 Platform Application Center Web GUI

2.3.7 IBM Platform Load Sharing Facility

IBM Platform Load Sharing Facility (LSF) is a powerful workload management platform for demanding, distributed HPC environments. It provides a comprehensive set of intelligent, policy-driven scheduling features that enable you to use all of the compute infrastructure resources and ensure optimal application performance.

LSF provides a resource management framework that takes your job requirements, finds the best resources to run the job, and monitors progress. Jobs always run according to host load and site policies.

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 difficult to maintain without automation. To reach the wanted format in a variant call workflow, you must follow some important steps to transform the data properly.

The sequence of the tasks and workflows must be carefully observed because specific input and output formats are required by the open source tools that are used in the process.

Note: For more information about the workflow process, see 3.5.1, “Installation of sample life science workflows” on page 76.

Fault tolerance

LSF has a robust architecture that was designed with fault tolerance in mind. Every component in the system has a recovery operation so that vital components are monitored by another component and can automatically recover from a failure.

LSF is designed to continue operating even if some of the hosts in the cluster are unavailable. One host in the cluster acts as the master, but if the master host becomes unavailable, another master host candidate takes over. LSF is available as long as one master host candidate is available in the cluster.

LSF can tolerate the failure of any host or group of hosts in the cluster. When a host becomes unavailable, all jobs that are running on that host are either requeued or lost, depending on whether the job was marked as rerunnable. No other pending or running jobs are affected.

LSF master node

To achieve the highest degree of performance and scalability, use a powerful master host. There is no minimum CPU requirement, and so any host with sufficient physical memory can run LSF as master host or master candidate host. Active jobs use most of the memory that LSF requires. The LSF master node connects to the campus (public) and provisioning networks.

LSF master candidate nodes

Two LSF master candidate nodes are recommended for failover of the Platform Application Center, which provides a graphic user interface to Platform LSF. Ideally, these nodes can be placed on separate racks for resiliency. Both LSF master candidate nodes connect to campus (public) and provisioning networks.

Running jobs are managed by **sbatchd** on each server host. When the new **mbatchd** starts, it polls the **sbatchd** on each host and finds the status of its jobs. If **sbatchd** fails but the host is still running, jobs running on the host are not lost. When **sbatchd** is restarted, it regains control of all jobs that are running on the host.

Figure 2-33 shows the LSF master nodes on mn01 and mn02.

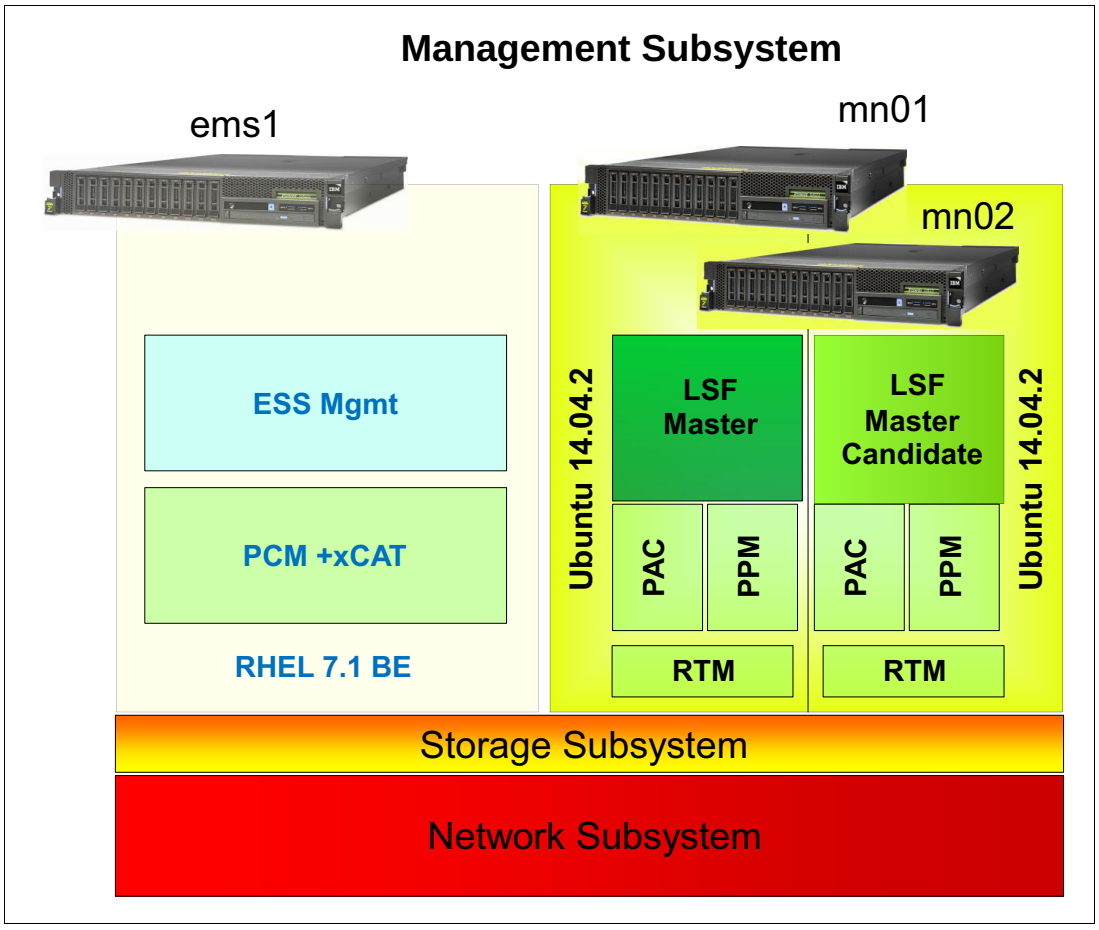


Figure 2-33 LSF master nodes on mn01 and mn02

How failover works

Fault tolerance in LSF depends on the event log file, `lsb.events`, which is kept on the primary file server. Every event in the system is logged in this file, including all job submissions and job and host status changes. If the master host becomes unavailable, a new master is chosen from the master candidate list, and **sbatchd** on the new master starts a new **mbatchd**. The new **mbatchd** reads the `lsb.events` file to recover the state of the system.

Host failover

The LSF master host is chosen dynamically. If the current master host becomes unavailable, another host takes over automatically. The failover master host is selected from the list defined in LSF_MASTER_LIST in the `lsf.conf` (specified in the `install.config` at installation). The first available host in the list acts as the master.

Job failover

Jobs can be submitted as rerunnable, so that they automatically run again from the beginning or as checkpointable, so that they start again from a checkpoint on another host if they are lost because of a host failure.

If all of the hosts in a cluster go down, all running jobs are lost. When a master candidate host comes back up and takes over as master, it reads the `lsb.events` file to get the state of all batch jobs. Jobs that were running when the systems went down are assumed to have exited unless they were marked as rerunnable, and email is sent to the submitting user. Pending jobs remain in their queues, and are scheduled as hosts become available.

Job exception handling

You can configure hosts and queues so that LSF detects exceptional conditions while jobs are running, and takes appropriate action automatically. You can customize what exceptions are detected and the corresponding actions. For example, you can set LSF to restart a job automatically if it exits with a specific error code.

2.3.8 IBM Platform Report Track Monitor

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. Running better tools, administrators can find efficiencies, reduce costs, and improve service levels by identifying and resolving resource management challenges quickly.

IBM Platform Report Track Monitor (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 interface enables Platform LSF administrators to manage multiple clusters easily while providing a better quality of service to cluster users.

Administrators can monitor both workloads and resources for all clusters in their environment by using a single monitoring tool.

Platform RTM displays resource-related information, such as the number of jobs that are submitted. The details of individual jobs (load average, CPU usage, and job owner) can be used to help improves service levels.

Figure 2-34 shows Platform RTM on mn01 and mn02.

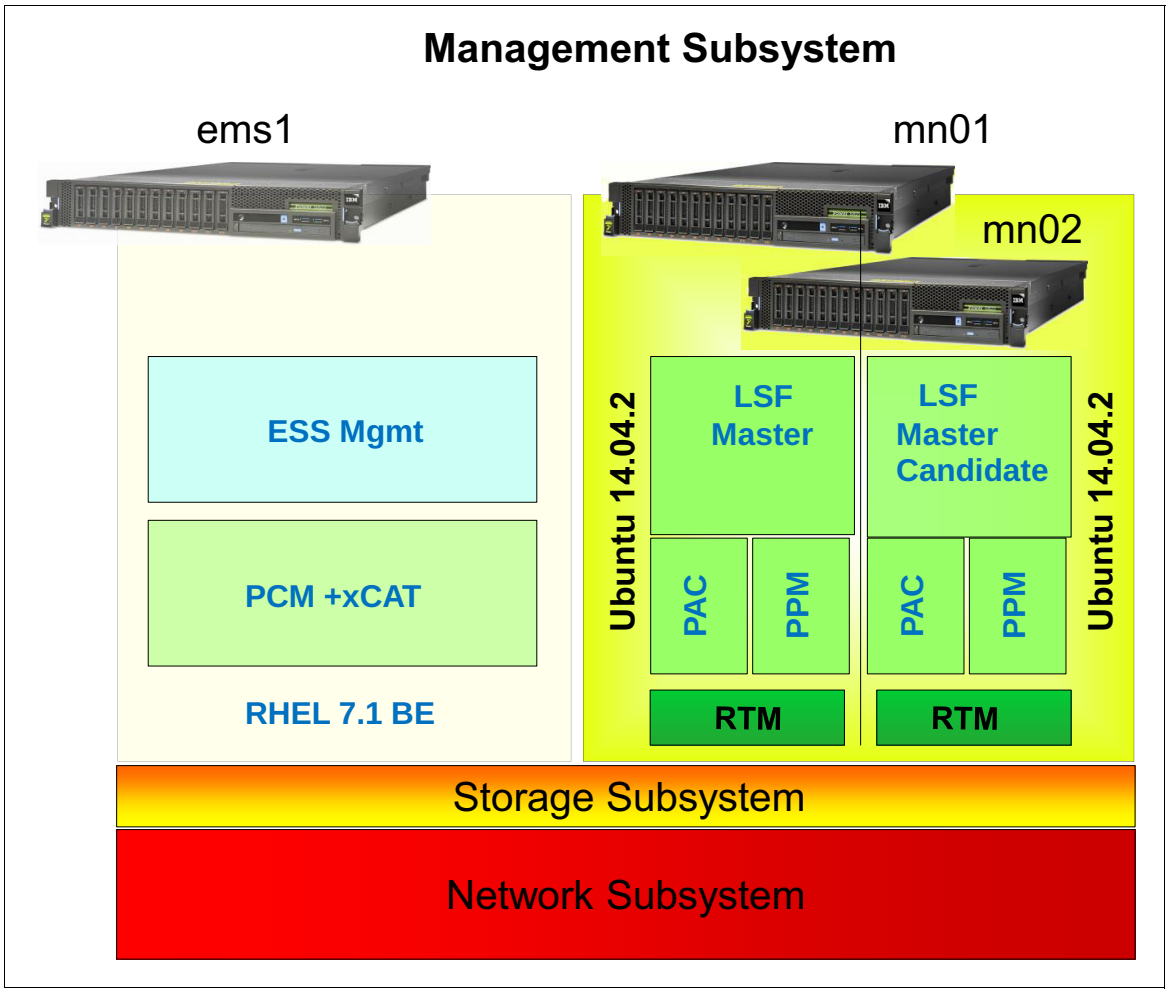


Figure 2-34 Platform RTM solution reference architecture deployment

2.3.9 IBM Platform Process Manager

The Platform Process Manager simplifies the design and automation of complex computational and analytic processes. With it, you can capture and protect repeatable best practices, and manage the complex workflows that are required by the genomic sequencing pipeline to create and manage a variant detection workflow for a genomic sequencing experiment.

An intuitive user interface makes it quick and easy to design workflows. And, when combined with Platform Application Center, the complexity of these processes can be hidden behind an easy-to-use, process-specific interface.

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 an LSF cluster. With the flow editor, you can create jobs and their relationships, and define dependencies based on files or time.

Figure 2-35 shows Platform Process Manager on mn01 and mn02.

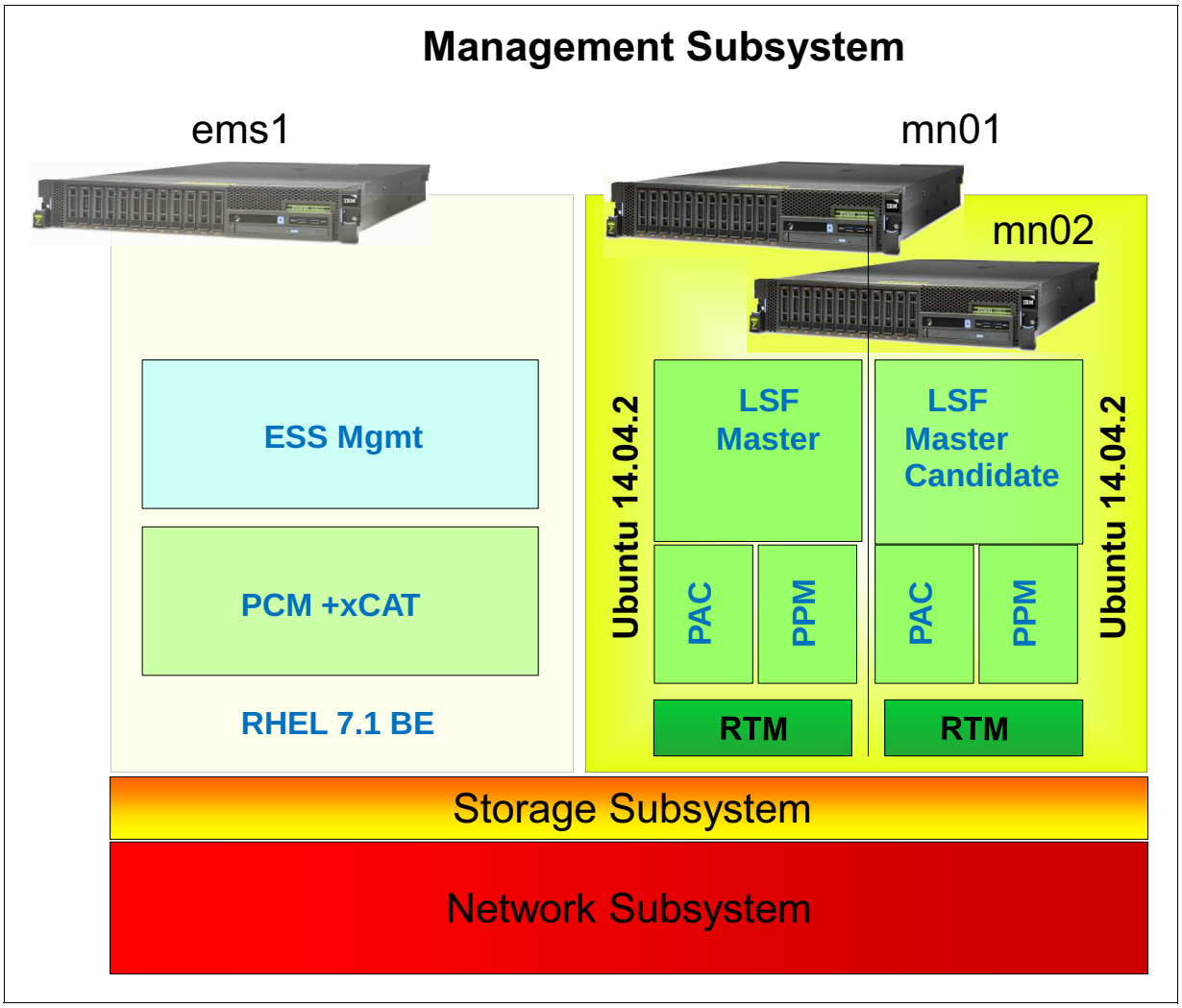


Figure 2-35 Platform Process Manager solution reference architecture deployment

2.4 Operations

This section describes two primary operations that are accomplished by a system within this architecture. The sequences and messages that are outlined below are conceptual and are not prescriptive. In other words, the actual specifics of the interactions between the elements within this architecture are left as design details for the specific elements that are chosen for a particular design. These operations are described here to illustrate how the elements within this architecture typically communicate and cooperate to accomplish a complete operation.

2.4.1 Submit a simple job

A common operation is to submit a simple job for execution by the system. A sequence diagram for this operation is depicted in Figure 2-36 on page 46. A user initiates this operation through the Application Center (for example, Platform Application Center).

The Application Center passes the request to the Workload Manager (LSF), which selects appropriate compute resources (for example, job slots on the nodes within a cluster) and schedules the job for execution on the selected resources.

To run, the job typically starts an application that is available to it. The application typically reads data from and writes data to the Data Hub. When execution completes, the elements report completion status back to their callers, and the Application Center represents the completion status to the user. The user can then access the results of the job, which are typically files stored within the data hub.

Figure 2-36 shows a sequence diagram for submitting a simple job.

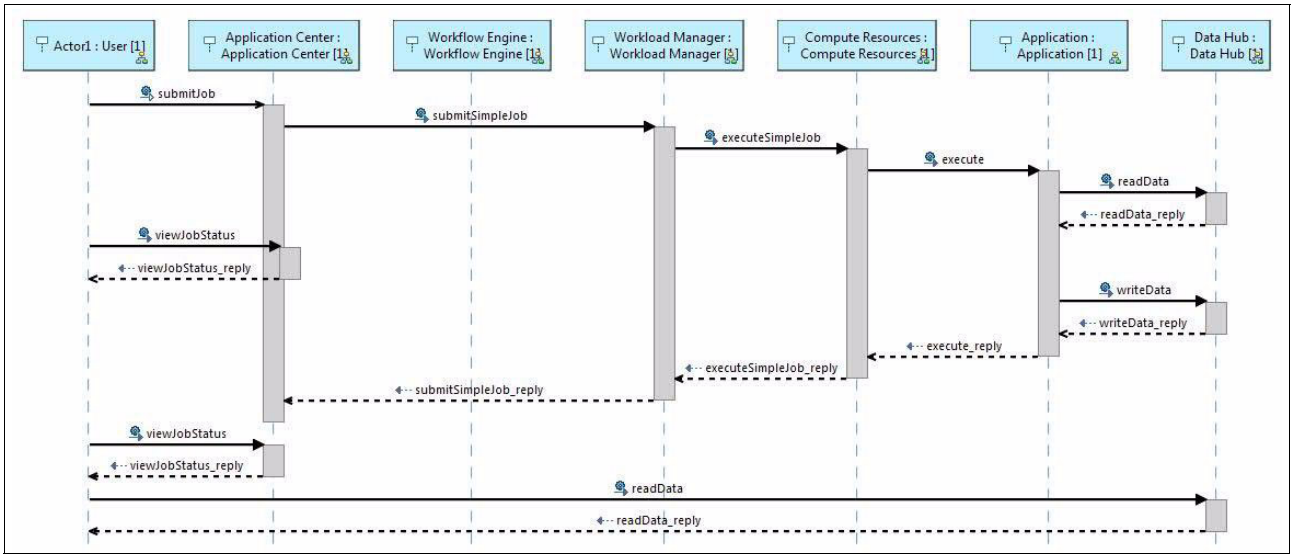


Figure 2-36 Sequence diagram - Submit simple job

2.4.2 Submit flow

Another common operation is to submit a flow for execution by the system. A sequence diagram for this operation is depicted in Figure 2-37 on page 47**.** A user initiates this operation through the application center (for example, Platform Application Center). The Application Center passes the request to the workflow engine, which submits the steps in the flow for execution.

Each step in the flow is essentially a simple job that is submitted to the workload manager (LSF) for processing, as described in 2.4.1, “Submit a simple job” on page 45. When each step in the flow completes, the workflow engine determines whether another step can be run, and, if so, it runs that next step.

When all of the steps of the flow are completed (based on the completion logic), the workflow engine reports the completion status to the application center, which represents the completion status to the user. The user can then access the results of the flow, which are typically files stored within the data hub.

Figure 2-37 shows the sequence diagram for the submit flow.

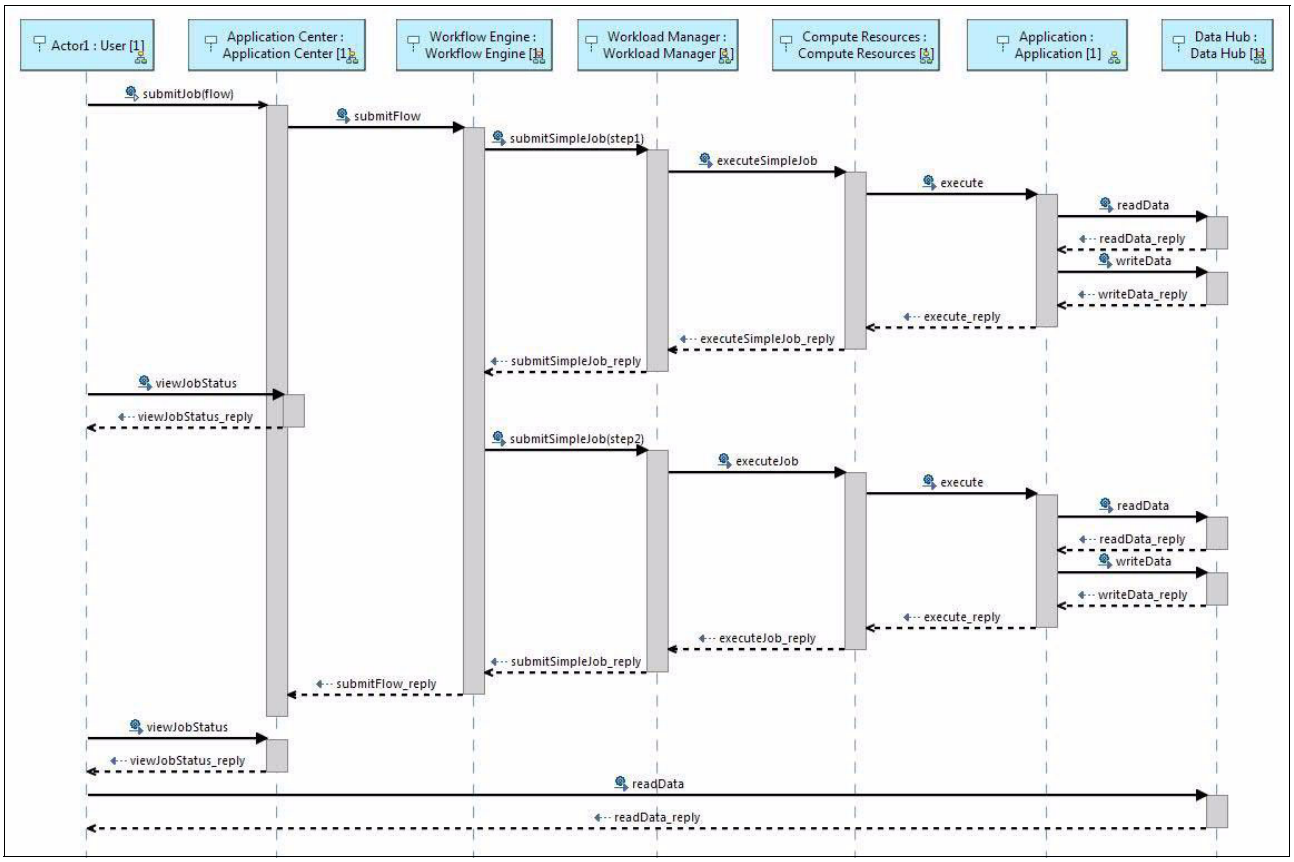


Figure 2-37 Sequence diagram - Submit flow

2.5 Network

Before any installation and configuration, the network must be designed and defined. Table 2-10 shows a typical network definition for a life science cluster. The network definition includes five networks in total. Three of these are private (internal to the cluster), and two are public (externally visible and accessible to elements outside of the cluster).

Table 2-10 Network definition

Network name	Public	Private
Campus network	Yes	-
Management network	Yes	Yes
Provisioning network	-	Yes
Service network	-	Yes
Data network	-	Yes

2.5.1 Logical mappings

This section describes the logical mappings of the example networks:

- Campus network: The campus network is usually a public (externally visible from the cluster) network, and it is the primary path for users to access the system. Users access the Platform Application Center over the campus network. The campus network is also the default path for movement of data into and out of the system.
- Management network: The management network is used by administrators or other privileged persons to access the infrastructure or other elements that are not intended to be accessible to users. The management network cannot be distinct from the campus network in some environments (the campus network can also be used for management).
- Provisioning network: The provisioning network is a private network that is used by the Resource Manager of Platform Cluster Manager to provision of the nodes within the system and subsequently monitor those nodes.
- Service network: This network is also known as the Flexible Service Processor network or *FSP network*. The service network is a private Ethernet network that is used to access the management processors of the servers within the system. A management processor can be an FSP (typical for Power Systems servers) or a BMC (typical for x86 servers). For FSP-based servers that are managed by an HMC, this is the private network that is used by the HMC to connect to the FSPs of those servers. For servers that are not managed by an HMC, this is the network over which persons or programs (for example, Platform Cluster Manager) access the management processors to accomplish operations like power control (for example, using IPMI). Systems with two HMCs have two service networks.

  The Platform Cluster Manager node must have a connection to this network at the operating system level that can used for hardware control through xCAT for operations such as power off and power on.
- Data network: This network is also known as the *application network*. The application network is typically a private network that is used to provide higher speed, higher bandwidth, and lower latency communication between the compute nodes and the storage nodes, and also between the compute nodes themselves. The application network can be Ethernet or InfiniBand.

Although the application network is typically private to facilitate the transfer of large amounts of data into and out of the system, the application network can also be bridged directly to other external (to the system) client networks to provide a more optimal data transmission path.

2.5.2 Network connections

The network subsystem is the collection of logical networks and the physical elements (for example, switches and cables) that host and realize them. The logical networks are specified by this architecture and consist of the networks listed as follows. The collection of switches (and their configurations) that realize these networks is largely left as a design consideration, with the following considerations:

- The application network can be Ethernet or InfiniBand.
- The other networks (campus network, management network, provisioning network, and service network) must be Ethernet.

The application network is typically a higher speed network that is used to support application processing on the compute nodes, and the related data movement to and from the storage

subsystem. Because of this configuration, the switches that support the application network are typically distinct and separate from those that support the other networks. The application network switches typically support 10 Gb or higher data rates, whereas the balance of the network switches typically require only 1 Gb.

Several networks are defined by this architecture to support communication and movement of data between its elements, usually virtual local area networks (VLANs), are defined. Table 2-11 shows the network VLANs.

Table 2-11 Network VLANs

Network name	VLAN
Campus network	2
Management network	1
Provisioning network	88
Service network	110
Data network	77

Figure 2-38 shows the logical network connections.

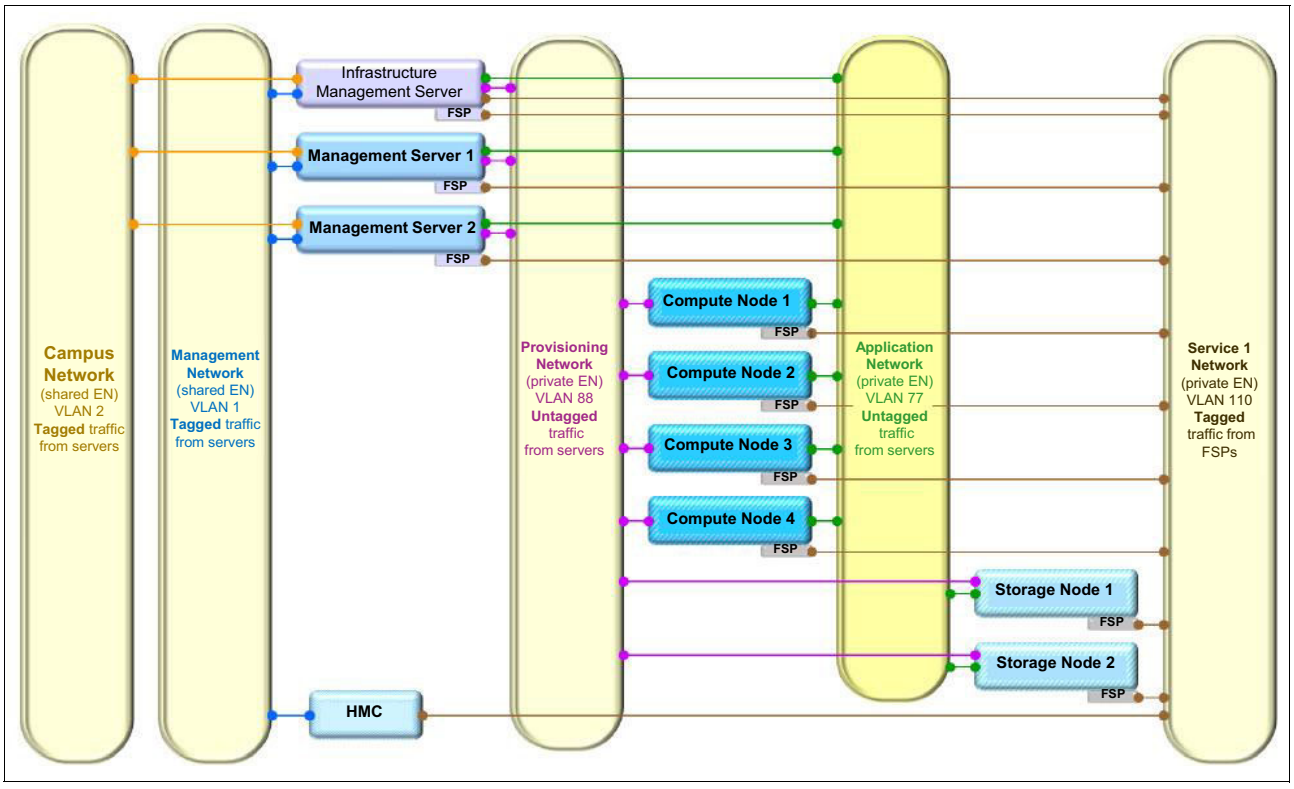


Figure 2-38 Logical network connections - All networks

Link Aggregation Control Protocol

To provide better performance, these hosts use Link Aggregation Control Protocol (LACP), which is an IEEE 802.3ad standard for grouping several physical ports into one logical port (known as a dynamic trunk group or link aggregation group) with any device that supports the standard.

The 802.3ad standard allows standard Ethernet links to form a single Layer 2 link by using LACP. Link aggregation is a method of grouping physical link segments of the same media type and speed in full duplex, and treating them as though they were part of a single, logical link segment. If a link in a LACP trunk group fails, traffic is reassigned dynamically to the remaining links of the dynamic trunk group.

Note: See IEEE 802.3ad-2002 for a full description of the standard at the following website:

http://www.ieee802.org/3/ad/

2.5.3 Cabling

The physical cabling for each server in the system is identical. Likewise, the switch configurations for the ports for each node are identical. This configuration provides consistency and reduces the opportunity for error. It also provides flexibility for special situations that can arise. Using this consistent physical cabling, each server is configured (within its operating system) to connect to the appropriate network in a manner that is consistent with the logical view that is described in Figure 2-38 on page 49.

Figure 2-39 show the infrastructure in block diagram.

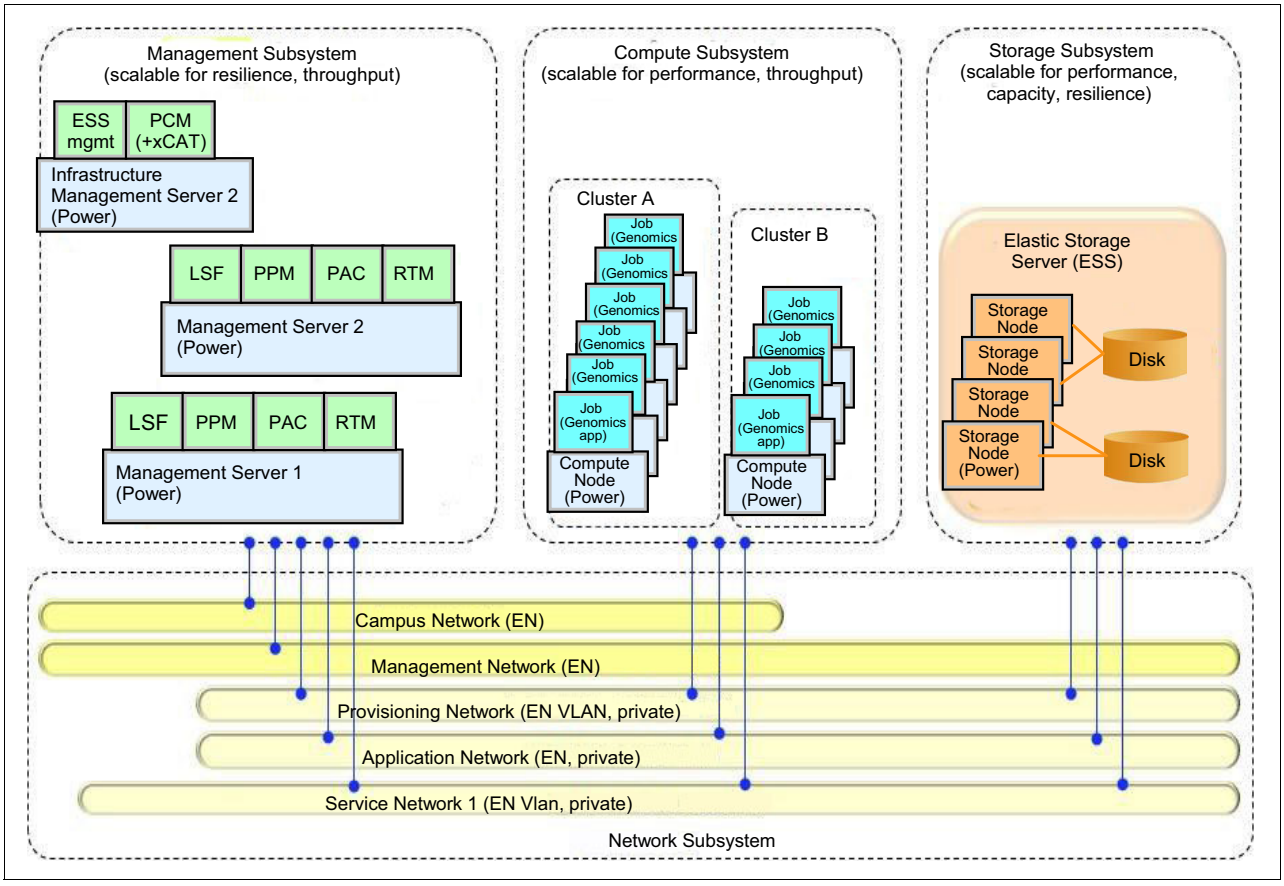


Figure 2-39 Infrastructure in block diagram

1 Gb networks

The connection between each server and the switches for the campus network, management network, provisioning network, and the operating system connection for the service network (infrastructure management server only) is carried over two physical links (cables) to the 1 Gb switches. This configuration provides a redundant path that is used to provide resilience for these networks.

The logical networks that are listed in Figure 2-39 on page 50 are trunked over this pair of links, minimizing the need for dedicated links for these networks. This pair of links is configured for link aggregation by using LACP on the server and on the switch. Further, the 1 Gb switch pair is configured (and cabled with an ISL) for Virtual Link Aggregation (VLAG), which allows the links to be aggregated across the pair of switches. On the server side, the two network interfaces for these two cabled ports are bonded together (LACP mode). IP address configuration is applied to the bond interface for the native VLAN (88), and VLAN-based interfaces with IP addresses are added for 1 Gb traffic that requires tagging (VLANs 1, 2, and 110).

The 1 Gb switches also make up the service network. The service network is different than the other 1 Gb networks in that each server has a single dedicated link between one of its FSP interfaces and one of the switches. With a single HMC and a single service network, all of these FSP interfaces are connected to just one of the switches (management switch A). The infrastructure management server also requires an operating system level connection to the service to accomplish power operations to the other servers in the system that it can provision.

Figure 2-40 shows the diagram of the cabling design for the 1 Gb Networks.

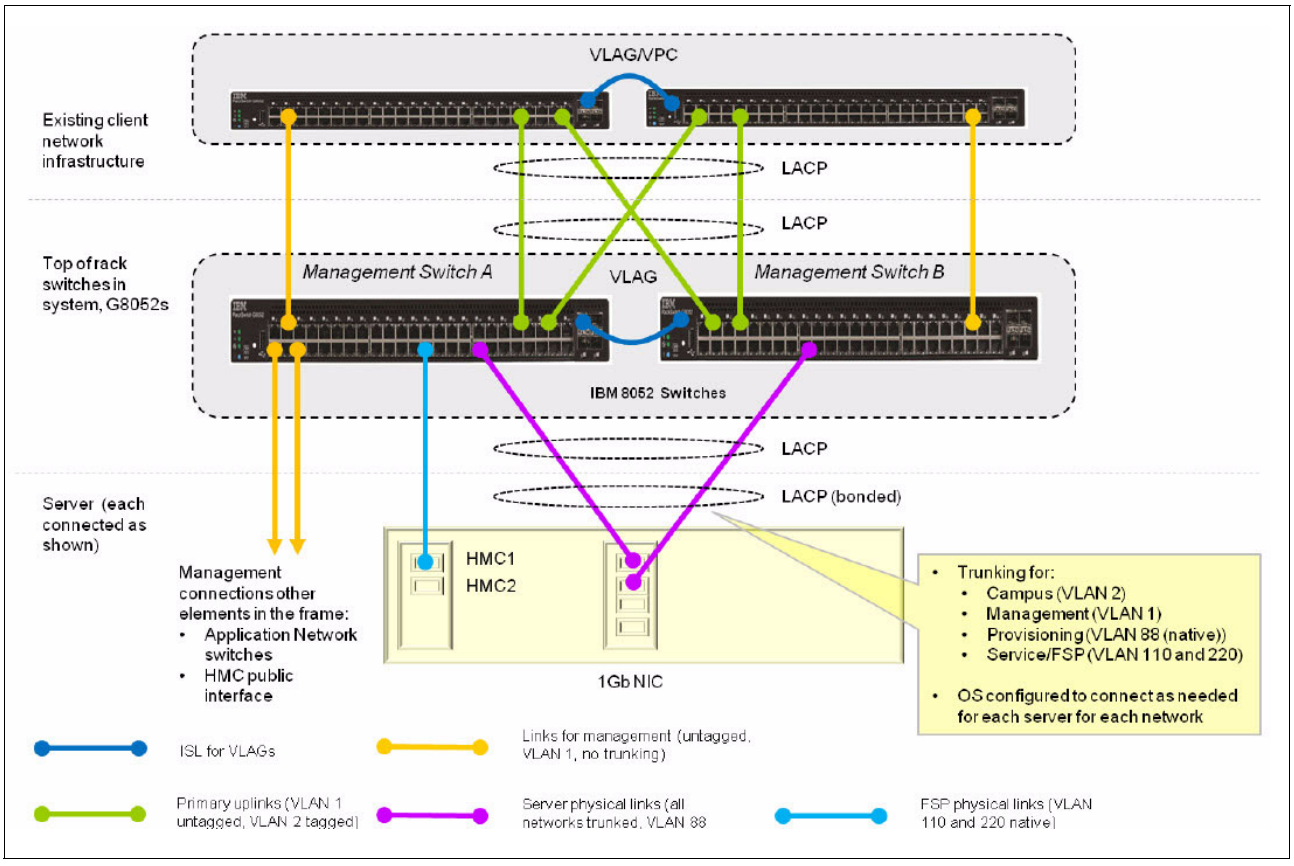


Figure 2-40 Diagram of the cabling design for the 1 Gb networks

Table 2-12 shows examples of connections on the switch.

Table 2-12 Example of connections on top of the rack switch G8052s

Host/Element	Hardware	Interface	Switch	Port
ems1	8247-21L	Un-P1-T1	Switch A	VLAN 110
ems1	8247-21L	Un-P1-C2-T1	Switch A	Trunk
ems1	8247-21L	Un-P1-C2-T2	Switch B	Trunk
mn01	8247-22L	Un-P1-T1	Switch A	VLAN 110
mn01	8247-22L	Un-P1-C2-T1	Switch A	Trunk
mn01	8247-22L	Un-P1-C2-T2	Switch B	Trunk
mn02	8247-22L	Un-P1-T1	Switch A	VLAN 110
mn02	8247-22L	Un-P1-C2-T1	Switch A	Trunk
mn02	8247-22L	Un-P1-C2-T2	Switch B	Trunk
hmc1	7042-CR8	Ethernet1	Switch A	VLAN 1
hmc1	7042-CR8	Ethernet4	Switch A	VLAN 110
cn01	8247-22L	Un-P1-T1	Switch A	VLAN 110
cn01	8247-22L	Un-P1-C2-T1	Switch A	Trunk
cn01	8247-22L	Un-P1-C2-T2	Switch B	Trunk
cn02	8247-22L	Un-P1-T1	Switch A	VLAN 110
cn02	8247-22L	Un-P1-C2-T1	Switch A	Trunk
cn02	8247-22L	Un-P1-C2-T2	Switch B	Trunk
cn03	8247-22L	Un-P1-T1	Switch A	VLAN 110
cn03	8247-22L	Un-P1-C2-T1	Switch A	Trunk
cn03	8247-22L	Un-P1-C2-T2	Switch B	Trunk
cn04	8247-22L	Un-P1-T1	Switch A	VLAN 110
cn04	8247-22L	Un-P1-C2-T1	Switch A	Trunk
cn04	8247-22L	Un-P1-C2-T2	Switch B	Trunk
gss01	8247-22L	Un-P1-T1	Switch A	VLAN 110
gss01	8247-22L	Un-P1-C2-T1	Switch A	Trunk
gss01	8247-22L	Un-P1-C2-T2	Switch B	Trunk
gss02	8247-22L	Un-P1-T1	Switch A	VLAN 110
gss02	8247-22L	Un-P1-C2-T1	Switch A	Trunk
gss02	8247-22L	Un-P1-C2-T2	Switch B	Trunk

10 Gb network

The connection between each server and the switches for the application network is carried over four physical links (cables) to the 10 Gb switches. This configuration provides a redundant path that is used to provide resilience for these networks, and increased bandwidth (up to 40 Gb) between the compute nodes and the storage subsystem.

With only a single logical network, no trunking or tagging is required, and the switch ports are configured to place the traffic from the servers on VLAN 77 as the native VLAN. Similar to the 1 Gb links, this set of four links is configured for link aggregation by using LACP on the server and on the switch. The 10 Gb switch pair is similarly configured (and cabled with an ISL) for VLAG, which allows the links to be aggregated across the pair of switches. See Table 2-13.

Table 2-13 Example of connections on top of the rack switches G8064s

Host/Element	Hardware	Interface	Switch	Port
ems1	8247-21L	Un-P1-C7-T1	Switch A	VLAN 77
ems1	8247-21L	Un-P1-C7-T2	Switch B	VLAN 77
ems1	8247-21L	Un-P1-C9-T1	Switch A	VLAN 77
ems1	8247-21L	Un-P1-C9-T2	Switch B	VLAN 77
mn01	8247-22L	Un-P1-C7-T1	Switch A	VLAN 77
mn02	8247-22L	Un-P1-C7-T2	Switch B	VLAN 77
cn01	8247-22L	Un-P1-C9-T1	Switch A	VLAN 77
cn01	8247-22L	Un-P1-C9-T2	Switch B	VLAN 77
cn01	8247-22L	Un-P1-C7-T1	Switch A	VLAN 77
cn01	8247-22L	Un-P1-C7-T2	Switch B	VLAN 77
cn02	8247-22L	Un-P1-C9-T1	Switch A	VLAN 77
cn02	8247-22L	Un-P1-C9-T2	Switch B	VLAN 77
cn02	8247-22L	Un-P1-C7-T1	Switch A	VLAN 77
cn02	8247-22L	Un-P1-C7-T2	Switch B	VLAN 77
cn03	8247-22L	Un-P1-C9-T1	Switch A	VLAN 77
cn03	8247-22L	Un-P1-C9-T2	Switch B	VLAN 77
cn03	8247-22L	Un-P1-C7-T1	Switch A	VLAN 77
cn03	8247-22L	Un-P1-C7-T2	Switch B	VLAN 77
cn04	8247-22L	Un-P1-C9-T1	Switch A	VLAN 77
cn04	8247-22L	Un-P1-C9-T2	Switch B	VLAN 77
cn04	8247-22L	Un-P1-C7-T1	Switch A	VLAN 77
cn04	8247-22L	Un-P1-C7-T2	Switch B	VLAN 77
gss01	8247-22L	Un-P1-C9-T1	Switch A	VLAN 77
gss01	8247-22L	Un-P1-C9-T2	Switch B	VLAN 77
gssi01	8247-22L	Un-P1-C7-T1	Switch A	VLAN 77

Host/Element	Hardware	Interface	Switch	Port
gss01	8247-22L	Un-P1-C7-T2	Switch B	VLAN 77
gss02	8247-22L	Un-P1-C9-T1	Switch A	VLAN 77
gss02	8247-22L	Un-P1-C9-T2	Switch B	VLAN 77
gss02	8247-22L	Un-P1-C7-T1	Switch A	VLAN 77
gss02	8247-22L	Un-P1-C7-T2	Switch B	VLAN 77

Figure 2-41 shows the diagram of the cabling design for the 10 Gb networks.

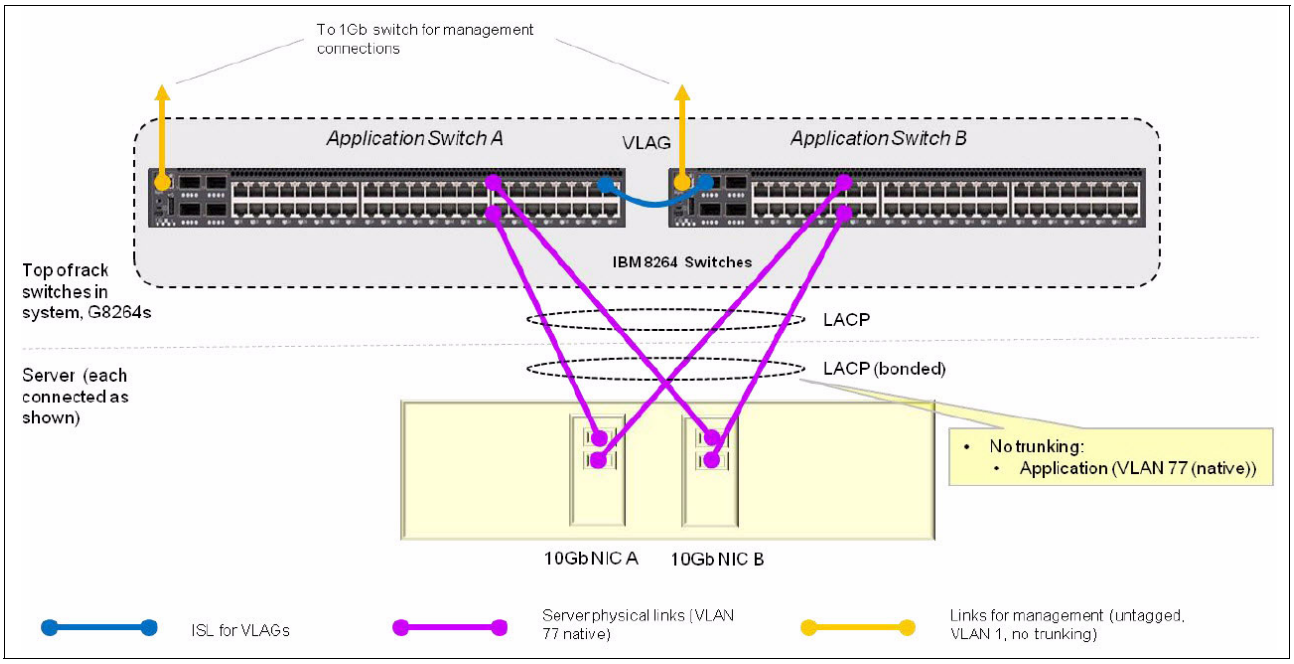


Figure 2-41 Diagram of the cabling design for the 10 Gb Networks

Dynamic Host Configuration Protocol

This architecture provides Dynamic Host Configuration Protocol (DHCP) for two of the networks in the system. The HMC is configured to provide DHCP for the service network, and the infrastructure management node is configured to provide DHCP for the provisioning network.

Note: For more information about how to configure the HMC as a DHCP server, see the IBM Knowledge Center at the following website:

`http://www.ibm.com/support/knowledgecenter/?lang=en`

3

Scenarios using the reference architecture with workflow examples

This chapter shows a scenario with workflow examples for genomics that can run on the reference architecture introduced in Chapter 2, “Reference architecture” on page 7. This basic solution uses the BioBuilds and GATK software.

This chapter includes the following sections:

- What is needed to start scenarios for genomics workloads
- Getting familiar with IBM Platform Computing software
- Introduction to genomic workflow
- Preparing the environment for Life Sciences solution for genomics workloads
- IBM Life Science Platform Provisioning package
- Storage options using IBM Spectrum Scale
- Adding compute nodes to a running cluster
- Additional information
- Other vendors packages for IBM Power Systems

3.1 What is needed to start scenarios for genomics workloads

Before working through the scenarios in this chapter, you need to complete these steps:

1. Create your environment by following the reference architecture in Chapter 2, “Reference architecture” on page 7, and in this chapter. IBM help deliver the best possible solution for your clustered environment.
2. Know how IBM Platform Computing solutions work.
3. Prepare the IBM Life Sciences solution for Genomics.
4. Know your genomic software.
5. Understand the genomic workflow.
6. Create or modify the templates delivered with the IBM solution.
7. Run the computing jobs.
8. Visualize the results.

3.2 Getting familiar with IBM Platform Computing software

This section provides details to help familiarize you with the IBM Platform Computing solutions software.

3.2.1 Terminology

The following IBM Platform LSF terminology is used through this chapter:

Job A command that is submitted to IBM Platform LSF batch. A job can take up more than one job slot.

Task An interactive command that is submitted to IBM Platform LSF base.

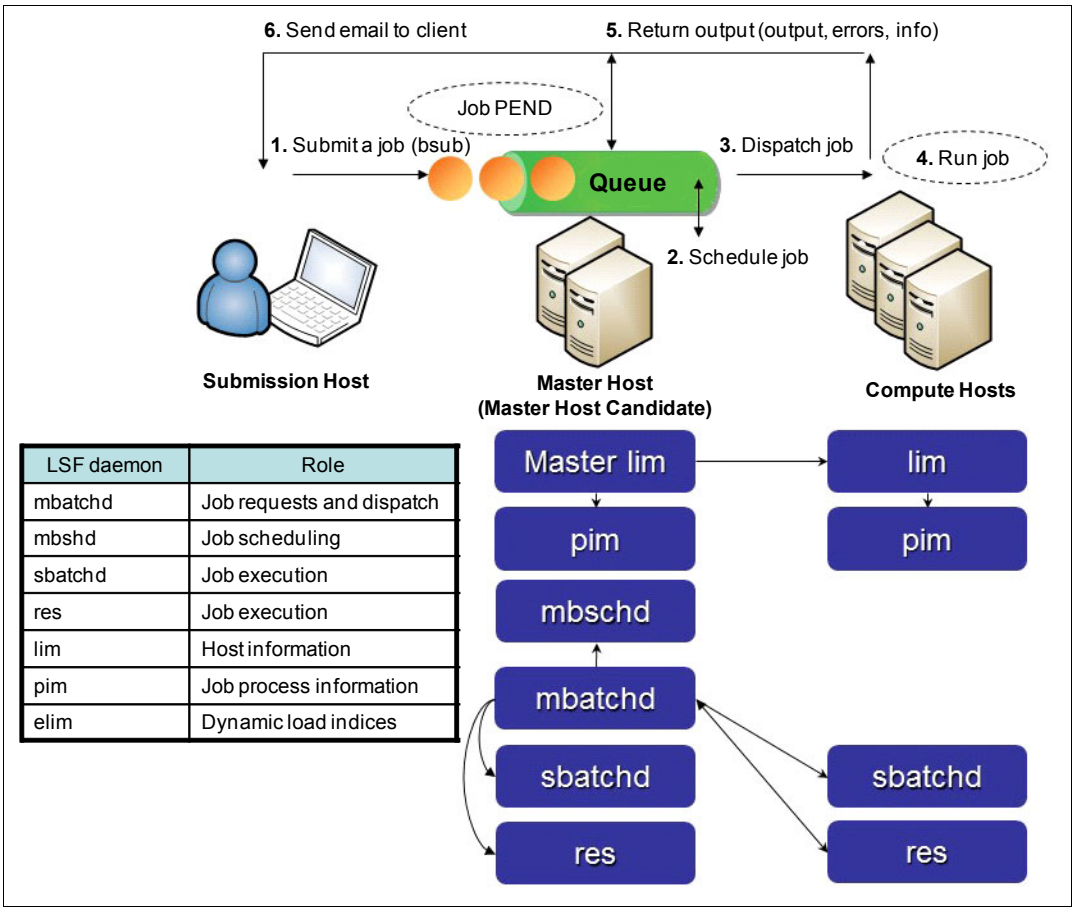
Queue A network-wide holding place for jobs that implement different job scheduling and control policies.

Job slot The basic unit of processor allocation in IBM Platform LSF. A job slot can be more than one per processor.

3.2.2 IBM Platform Load Sharing Facility job lifecycle

IBM Platform Load Sharing Facility (LSF) is a workload management platform for high-performance computing (HPC) environments. An IBM Platform LSF cluster can be divided into two groups of hosts: Management hosts and compute hosts. Management hosts provide specialized services to the cluster, and compute hosts run the user workload.

Figure 3-1 shows the basic structure of IBM Platform LSF with the job lifecycle and the communication paths between the daemons in the cluster.



LSF daemon	Role
mbatchd	Job requests and dispatch
mbshd	Job scheduling
sbatchd	Job execution
res	Job execution
lim	Host information
pim	Job process information
elim	Dynamic load indices

Figure 3-1 Platform LSF job lifecycle

Figure 3-1 shows the following steps:

1. Submit a job. You submit a job from an LSF client or server with the **bsub** command. If you do not specify a queue when you submit the job, the job is submitted to the default queue. Jobs are held in a queue and wait to be scheduled. These jobs are in the PEND state.
2. Schedule job. The master batch daemon (**mbatchd**) looks at jobs in the queue and sends these jobs for scheduling to the master batch scheduler (**mbschd**) at a preset time interval.

   mbschd evaluates jobs and makes scheduling decisions that are based on job priority, scheduling policies, and available resources. **mbschd** selects the best hosts where the job can run and sends its decisions back to **mbatchd**.

   Resource information is collected at preset time intervals by the master load information manager (LIM) daemon from LIMs on server hosts. The master LIM communicates this information to **mbatchd**, which in turn communicates it to **mbschd** to support scheduling decisions.
3. Dispatch the job. As soon as **mbatchd** receives scheduling decisions, it immediately dispatches the jobs to hosts.
4. Run job. The slave batch daemon (**sbatchd**) performs these tasks:
   a. Receives the request from **mbatchd**.
   b. Creates a child **sbatchd** for the job.

c. Creates the execution environment.

d. Starts the job by using a remote execution server (*res*).

5. Return output. When the job is completed, it is assigned the DONE status if the job completed without any problems. The job is assigned the EXIT status if errors prevented the job from completing. **sbatchd** communicates job information, including errors and output, to **mbatchd**.

6. Send email to client. **mbatchd** returns the job output, job error, and job information to the submission host through email.

3.2.3 Access the Platform Computing graphical interface

IBM Platform LSF does not provide a GUI interface, but the full workload and cluster management functionality is available through the command line. Some of the functions that are offered by IBM Platform LSF, such as job submission and resources management, are available through a GUI by using optional add-ons.

IBM Platform Application Center

The web-based interface of IBM Platform Application Center (PAC) simplifies workload management with remote job monitoring, easy access to job-related data, and the capability to manage jobs, such as stopping, suspending, resuming, or requeuing jobs.

To access the PAC GUI, use the URL `http://<PAC management node>:9090/platform` where 9090 is the default port value.

Log in with the LSF admin ID and password. The admin ID and password are set to the value specified for the user defined variable LSF_ADMIN_USER in the cluster template. You can also find URL for the PAC GUI in the installation logs under the `/opt/ibm/lifescience/prov/pac/pacinstall.log` directory.

IBM Platform Report Track Monitor

IBM Platform Report Track Monitor (RTM) is an operational dashboard for IBM Platform LSF environments that provides comprehensive workload monitoring, reporting, and management.

Note: IBM Platform RTM uses Cacti as a rich graphical user interface framework.

To access the RTM GUI, use the following URL `http://<RTM management node>:80/cacti` where `80` is the default port value.

Log in with the admin ID and password. The default administrator ID is *admin*, and the default password for this account is *admin*. Platform RTM supports Firefox 3.6.11 and later, and Internet Explorer (IE) 8 and 9.

IBM Platform Process Manager

While technical users employ the XML-based file formats, a rich command set allows time-dependent jobs or flows to be triggered and managed through scripts, which is also an alternative to the graphical user interface (GUI).

Users or designers can use the graphical interface to describe workflow steps and dependencies so that lengthy, repetitive tasks that are prone to human error can be automated. User-defined and system-defined calendars can be combined so that workflows and individual jobs can be run automatically at predefined times.

The following sections describe how to access the IBM Platform Process Manager (PPM) GUI and commands.

On a Linux system

Source the PPM profile to access the PPM commands in the command line as shown in Example 3-1.

Example 3-1 PPM profile

```
$ source  <PPM install directory>/conf/profile.js

Start the PPM floweditor GUI to create or modify workflows
$ floweditor &

Start the PPM flowmanager GUI to manage the workflows
$ flowmanager &

Start the PPM caleditor GUI to manage calendar events
$ caleditor &
```

Note: The Linux server must have the X11 display installed to access the PPM GUI.

On a Windows system

Install the PPM Windows client, *ppm9.1.3.0_pinstall_client.msi* to access the floweditor, flowmanager, and caleditor GUI.

For more information, see the PPM Administration Guide under the Add a Windows client section at the following website:

https://support.sas.com/rnd/scalability/platform/PSS7.1/pm8.20.pdf

3.2.4 Platform Computing software administration tips

This section provides a guide about how to administer Platform Computing applications.

IBM Platform LSF

To start Platform LSF, source the `profile.lsf` file if you have not done so, and then use the **`lsfstartup`** command as shown in Example 3-2.

Example 3-2 Starting Platform LSF

```
root@mn01:/gpfs/lsf/conf# source profile.lsf
root@mn01:/gpfs/lsf/conf# lsfstartup
Starting up all LIMs ...
Do you really want to start up LIM on all hosts ? [y/n]y
Start up LIM on <mn01> ...... done
Start up LIM on <cn01> ...... done
Start up LIM on <cn02> ...... done
Start up LIM on <cn03> ...... done

Waiting for Master LIM to start up ...  Master LIM is ok
Starting up all RESes ...
Do you really want to start up RES on all hosts ? [y/n]y
Start up RES on <mn01> ...... done
Start up RES on <cn01> ...... done
```

```
Start up RES on <cn02> ...... done
Start up RES on <cn03> ...... done

Starting all slave daemons on LSBATCH hosts ...
Do you really want to start up slave batch daemon on all hosts ? [y/n] y
Start up slave batch daemon on <mn01> ...... done
Start up slave batch daemon on <cn01> ...... done
Start up slave batch daemon on <cn02> ...... done
Start up slave batch daemon on <cn03> ...... done

Done starting up LSF daemons on the local LSF cluster ...
```

To stop Platform LSF, source the `profile.lsf` file if you have not done so, and use the **`lsfshutdown`** command as shown in Example 3-3.

Example 3-3 Stopping Platform LSF

```
root@mn01:/gpfs/lsf/conf# source profile.lsf
root@mn01:/gpfs/lsf/conf# lsfshutdown
Shutting down all slave batch daemons ...

Shut down slave batch daemon on all the hosts? [y/n] y
Shut down slave batch daemon on <cn01> ...... done
Shut down slave batch daemon on <cn02> ...... done
Shut down slave batch daemon on <cn03> ...... done
Shut down slave batch daemon on <mn01> ...... done
Shutting down all RESes ...
Do you really want to shut down RES on all hosts? [y/n] y
Shut down RES on <mn01> ...... done
Shut down RES on <cn01> ...... done
Shut down RES on <cn02> ...... done
Shut down RES on <cn03> ...... done
Shutting down all LIMs ...
Do you really want to shut down LIMs on all hosts? [y/n] y
Shut down LIM on <mn01> ...... done
Shut down LIM on <cn01> ...... done
Shut down LIM on <cn02> ...... done
Shut down LIM on <cn03> ...... done
```

IBM PAC

To start PAC, source the `profile.pmc` file and use the command **`perfadmin start all`** as shown in Example 3-4.

Example 3-4 Starting PAC

```
root@mn01:/gpfs# source /opt/pac/gui/conf/profile.pmc

root@mn01:/opt/pac/gui/conf# perfadmin start all
Starting service <jobdt>. Run <perfadmin list> to check startup.
Starting service <plc>. Run <perfadmin list> to check startup.
Starting service <purger>. Run <perfadmin list> to check startup.

root@mn01:/opt/pac/gui/conf# pmcadmin start
Starting service <WEBGUI>. Run <pmcadmin list> to check startup.
```

To stop PAC, source the `profile.pmc` file if you have not done so, and use the command **`perfadmin stop all`** as shown in Example 3-5.

Example 3-5 To stop PAC

```
root@mn01:/gpfs# source /opt/pac/gui/conf/profile.pmc
root@mn01:/gpfs# perfadmin stop all
Service <jobdt> stopped successfully.
Service <plc> stopped successfully.
Service <purger> stopped successfully.
root@mn01:/gpfs# pmcadmin stop          // stops GUI
Service <WEBGUI> stopped successfully.
```

IBM PPM

To start PPM, navigate to the `/ppm/conf` directory, source the `profile.js` file, and start the **`jadmin`** program as shown in Example 3-6.

Example 3-6 Starting PPM

```
root@mn01:/gpfs/ppm/conf# source profile.js
root@mn01:/gpfs/ppm/conf# jadmin start
Starting up jfd ...
```

It takes a moment for the `jfd` daemon to start. You can check if `jfd` started by issuing the following command in the command line: **`ps -aux|grep jdf|grep -v grep`**.

To stop PPM, source the `profile.js` file if you have not done so, and stop using the **`jadmin`** as shown in Example 3-7.

Example 3-7 Stopping PPM

```
root@mn01:/gpfs/ppm/conf# source profile.js
root@mn01:/gpfs/ppm/conf# jadmin stop
Shutting down jfd ...
jfd is being shut down.
```

It takes a moment for the `jfd` daemon to close itself. Check whether `jfd` stopped properly by issuing the **`ps -aux|grep jdf|grep -v grep`** command.

Note: While stopping or starting Platform Computing software, follow the correct order. For example, PPM must be started before PAC. Also, ensure that the `jfd` daemon stopped properly before starting PAC.

3.2.5 Workflow management in PAC

Users can submit jobs from IBM Platform Application Center by using submission forms. Submission forms make job submission less error prone because they provide an interface that makes it easier for the user to understand the required configuration and input for the job to run successfully.

IBM Platform Application Center includes a set of application templates that can be published (after they are adapted to your environment) and used for job submission. You see all of the available application templates in the Resources tab under the Submission Templates view. When you select any of the available templates and publish them (if they are in the state Unpublished), they are available for users at the Jobs tab under the Submission Forms view.

Job directory

The job directory is the directory in which job data is located. The job directory is created automatically when the user submits a job. A user's job files are stored in the job repository. By default, the job repository for a user is defined at the cluster level, and the job files for all applications are located together under one directory. The default job repository is defined in the file `$GUI_CONFDIR/Repository.xml`. Example 3-8 shows that the job directory is set in `/gpfs/home`.

By default, if the job is submitted without a job name, the job directory is named with the application name, the time stamp, and five random characters in the form `applicationname_timestampfiverandomcharacters`.

Example 3-8 Repository.xml file content

```
root@mn01:/opt/pac/gui/conf# cat Repository.xml
<?xml version="1.0" encoding="UTF-8"?>
<ParamConfs>
        <Configuration>
                <Repository>
                        <User>all</User>
                        <Path>/gpfs/home</Path>
                </Repository>
                <!--
                <ShareDirectory>
                        <Path>/data/share</Path>
                </ShareDirectory>
                -
        </Configuration>
</ParamConfs>
```

3.2.6 Flow definitions in PPM

Unlike application templates, flow definitions cannot be created in IBM Platform Application Center. They must be created in IBM Platform Process Manager. You can see and manage (**`trigger/release/publish/unpublish/hold`**) flow definitions from IBM Platform Application Center. You can see the available flow definitions by clicking the **Resources tab** and selecting **Submission Templates** → **Flow Definitions**.

3.2.7 Managing job flows

After the installation, you can run the Flow Editor by running the command **`floweditor.`** From the Flow Editor, you can open job flows that are already available or create new job flows.

After you submit a job flow, you can visualize it in the Flow Manager. To run the Flow Manager, run the command **`flowmanager`**.

3.2.8 Monitoring and statistics

IBM Platform Application Center offers a dashboard with details about cluster health and cluster performance statistics, and reports about resources. Click the Resources and then Dashboard tabs as shown in Figure 3-2.

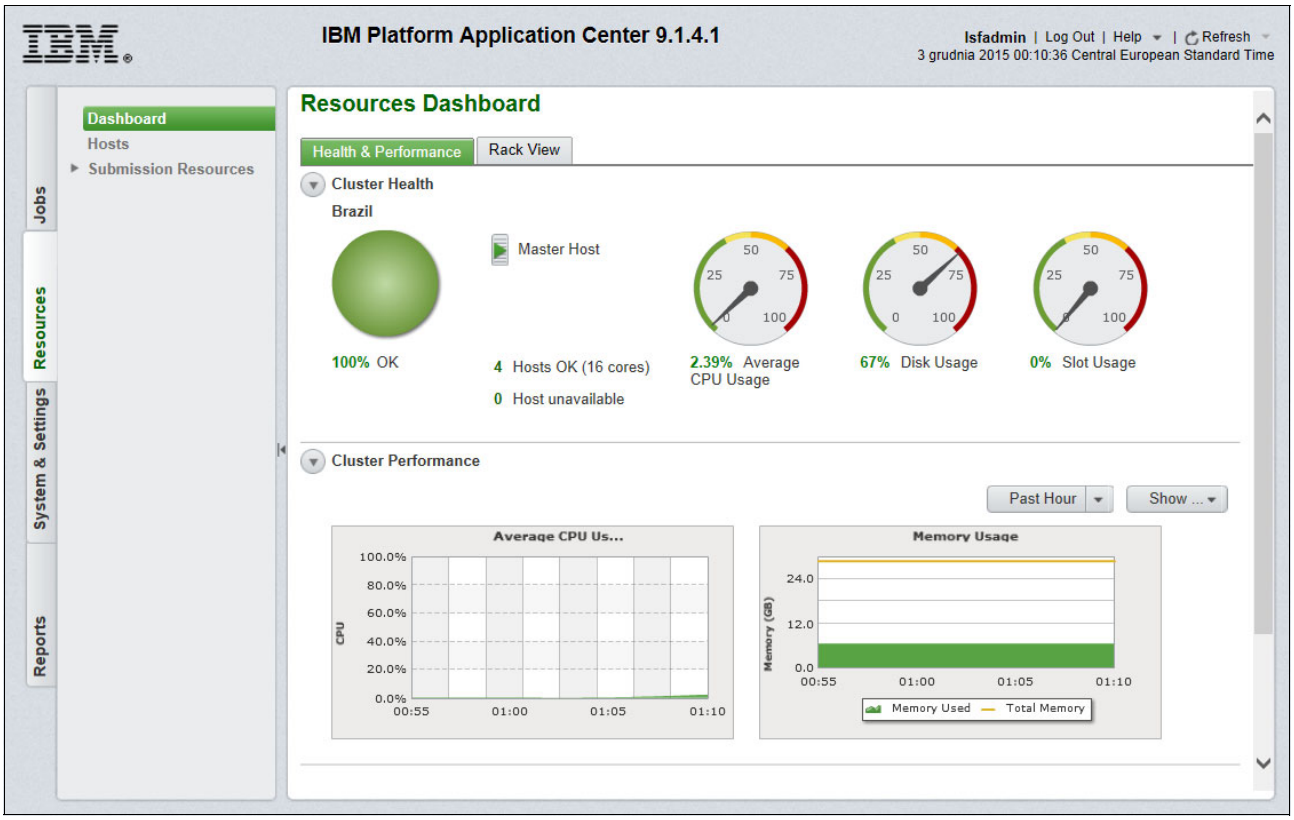


Figure 3-2 Resources dashboard - IBM Platform Application Center

The product offers a set of built-in reports for cluster analysis. These reports are the most common reports that are required for identifying areas of improvement on your cluster. Custom reports can also be created from the web interface to satisfy specific needs. Click the Reports tab. You can choose between Job and Resource reports. See the list of available job reports in Figure 3-3. The section of Resource reports includes:

- Cluster Availability - LSF: LSF host availability in an LSF cluster.
- Host Resource Usage: Resource usage trends for selected hosts.
- License Usage: The license usage under License Scheduler.

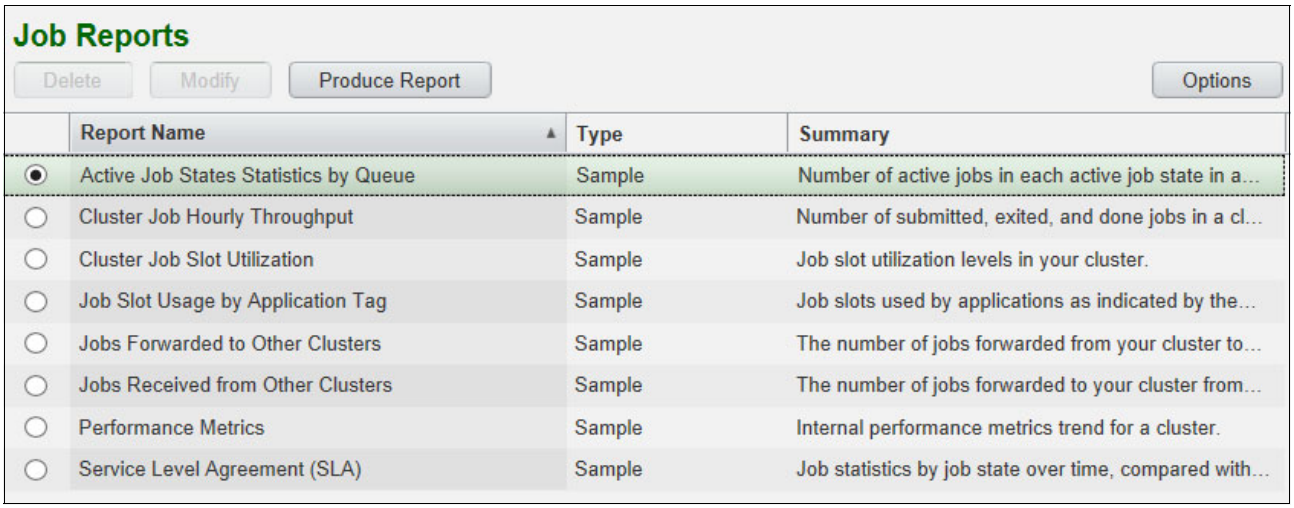


Figure 3-3 Job Reports in PAC

If you are interested in more detailed reports, you can use IBM Platform RTM, which is a real-time dashboard for monitoring global workloads and resources.

Note: Platform RTM provides a complete, integrated monitoring facility that is designed specifically for Platform LSF environments.

You can find out more about Platform RTM at the following website:

`http://www.ibm.com/systems/platformcomputing/products/lsf/rtm.html`

Section 4.3.3 IBM Platform RTM implementation in *IBM Platform Computing Solutions*, SG24-8073 describes IBM Platform RTM installation steps and provides many usage examples.

In IBM Platform RTM from the Grid tab → Job Info section on the left menu of the IBM Platform RTM GUI, you can see all sorts of information about jobs in your clusters. By accessing the **Details** menu under Job Info, you can see the following information:

- Name and status of a job
- State changes
- Submission user
- CPU usage and efficiency
- Execution host
- Start and end time

You can also configure the items that you see on this page or see all available information about the job by clicking the job ID. If you click a job ID, you can also access more job details, job graphs, host graphs, and pending reasons for the job. Each element is on a different tab.

3.3 Introduction to genomic workflow

This section provides an introduction to genomic workflows.

3.3.1 Terminology

The following terminology is used throughout this section:

Genomics The study of genes and their functions, and related techniques. Genomics goal is to understand the structure of the genome, including the mapping genes and sequencing the DNA. For more information, see http://www.who.int/genomics/geneticsVSgenomics/en/.

Genome The total of genetic information of an individual, which is encoded in the structure of deoxyribonucleic acid (DNA). A major challenge still must be met in understanding what the genome contains and how the genome functions. For more information, see the Genomics and World Health: Report of the Advisory Committee on Health research, Geneva, WHO (2002) at: http://apps.who.int/iris/bitstream/10665/42453/1/a74580.pdfhttp://apps.who.int/iris/bitstream/10665/42453/1/a74580.pdf.

NGS Next generation sequencing determines the order of nucleotides within a DNA molecule. DNA polymerase catalyzes the incorporation of fluorescent labeled nucleotides into a DNA template strand during sequential cycles of DNA synthesis. For more information, see NGS primer for pathologists, An introduction to NGS technology and applications for molecular pathologists at: http://bit.ly/1Qdx7rL.

DNA Deoxyribonucleic acid, a molecule that encodes an organism's genetic blueprint.

Nucleotides Organic molecules that build nucleic acids like DNA and RNA. The five bases are adenine (A), guanine (G), cytosine (C), thymine (T), and uracil (U).

3.3.2 Introduction to genomics

Capillary electrophoresis (CE)-based (Sanger) sequencing allows scientists to illuminate genetic information from many biological systems. Although widely adopted, this technology has limitations in throughput, scalability, resolution, and speed.[1] Next generation sequencing (NGS) technologies have been instrumental in significantly accelerating biological research and discovery of genomes for humans, mice, snakes, plants, bacteria, virus, and cancer cells.

NGS technologies parallelize the sequencing process, producing thousands or millions of sequences at a time. High-throughput sequencing technologies generate millions of short reads from a library of nucleotide sequences.

Researchers now process immense data sets to build analytical deoxyribonucleic acid (DNA) models for large genomes, and use reference-based analytic methods to further their understanding of genomic models. This analysis is useful for drug discovery, personalized medicine, toxicology, forensics, agriculture, nanotechnology, and other emerging use cases.

[1] HLA NGS primer, *An introduction to Illumina next-generation sequencing for HLA typing.* http://www.illumina.com/content/dam/illumina-marketing/documents/products/HLA_NGS_primer_110713_Web.pdf

The output of sequencers is raw data and needs to be processed by computational pipelines to extract meaningful information. The purpose is to map the resulting sequences to the correct position in the genome or analyze the mapped genomic sequences for aberrations against a reference genome.

3.3.3 Genomic sequencing pipeline

Sequencing analysis pipelines consist of a number of computationally intensive steps to transform raw data from a sequencer into human-readable data. For example, comparing gene expression differences between a diseased tissue sample and a healthy reference in a patient population or to predict disease-causing chromosomal aberrations, such as fusion genes[2] are difficult tasks.

Pipelines are highly modular, which means that several algorithms can work in sequence and data communities pass from one step to the next in different formats. Moreover, many individuals from the scientific communities can work on composing their steps in the pipeline and publish them into one processing queue.

The GATK best practices workflows[3] provide recommendations for performing variant discovery analysis in high-throughput sequencing (HTS) data:

- Pre-processing starts from raw sequence data, either in FASTQ or uBAM format, and produces analysis-ready BAM files. Processing steps include alignment to a reference genome and some data cleanup operations to correct for technical biases, and make the data suitable for analysis.
- Variant Discovery starts from analysis-ready BAM files and produces a callset in VCF format. Processing involves identifying sites where one or more individuals display possible genomic variation, and applying appropriate filtering methods to the experimental design.
- Callset Refinement starts and ends with a VCF callset. Processing involves using metadata to assess and improve genotyping accuracy, attach additional information, and evaluate the overall quality of the callset.

Figure 3-4 shows a typical genomic pipeline.

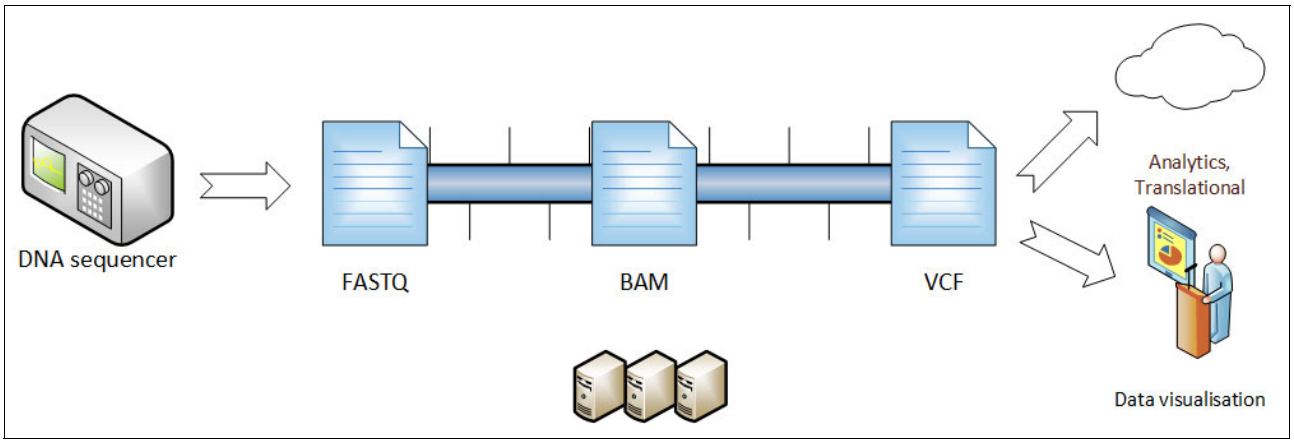


Figure 3-4 Typical genomic pipeline

[2] *Big Data Analytics in Bioinformatics and Healthcare*, by Baoying Wang, Ruowang Li, and William Perrizo (eds), IGI Global (c) 2015, 9781466666115; Chapter 9 - Analysis of Genomic Data in a Cloud Computing Environment

[3] https://www.broadinstitute.org/gatk/guide/best-practices

The described implementation presents the Genome Analysis Toolkit (or GATK) framework, the core system of which uses the standard Sequence Alignment/Map (SAM) format to represent reads using a production-quality SAM library *Picard*, which is publicly available and also included in the BioBuilds package. SAMtools implements various utilities for post-processing alignments in the SAM format, such as indexing, variant caller, and alignment viewer. This combination gives scientists universal tools for processing read alignments.

The SAM file specification is the standard format for storage of platform-independent next-generation sequencing data. It has been designed to scale to alignment sets of (10^11) or more base pair, which is typical for the deep resequencing of one human individual.[4]

The SAM format consists of one header section and one alignment section. Table 3-1 shows mandatory fields in the SAM format.

Table 3-1 Mandatory fields in SAM format

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based left most POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based left most Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

A detailed format specification and the complete documentation of SAMtools are available at the following SAMtools website:

http://samtools.sourceforge.net

The equivalent binary alignment version of the SAM format, called Binary Alignment/Map (BAM), is compressed and indexed, and is used by the GATK for performance reasons. Using positional sorting and indexing, applications can perform stream-based processing on specific genomic regions without loading the entire file into memory.[5]

Before you get SAM or BAM analysis-ready files, you need to process raw data for instance in FASTQ format, which is a simple extension from traditional FASTA format.

[4] Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, 1000 Genome Project Data Processing Subgroup. 2009. The Sequence Alignment/Map format and SAMtools. Bioinformatics 25: 2078–2079. http://bioinformatics.oxfordjournals.org/content/25/16/2078.full

[5] Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, 1000 Genome Project Data Processing Subgroup. 2009. The Sequence Alignment/Map format and SAMtools. Bioinformatics 25: 2078–2079. http://bioinformatics.oxfordjournals.org/content/25/16/2078.full

FASTQ Format Specification:

Notations:

- <fastq>, <blocks> and so on represents non-terminal symbols.
- Characters in red are regex-like operators.
- '\n' stands for the Return key.

Syntax:

```
<fastq>      :=<block>+
<block>      :=@<seqname>\n<seq>\n+[<seqname>]\n<qual>\n
<seqname>    :=[A-Za-z0-9_.:-]+
<seq>        :=[A-Za-z\n\.~]+
<qual>       :=[!-~\n]+
```

For the requirements for the FASTQ files, refer to the following website:

http://maq.sourceforge.net/fastq.shtml

3.4 Preparing the environment for Life Sciences solution for genomics workloads

IBM has created an exemplary solution for Genomics with workflow samples. The solution as a base incorporates GATK software and programs from Lab7 BioBuilds.

- BioBuilds package is Lab7 Systems' collection of pre-built binary files for common bioinformatics tools that run on IBM POWER® servers. BioBuilds package includes open source programs like BWA, SAMtools, and Picard, and is delivered as a platform independent package with binary files.
- The Genome Analysis Toolkit (GATK) is a software package developed at the Broad Institute to analyze high-throughput sequencing data. It is a structured programming framework that is designed to ease the development of efficient and robust analysis tools for next-generation DNA sequencers by using the functional programming philosophy of MapReduce.[6]

For information, see Appendix A, "Useful software information" on page 119.

3.4.1 BioBuilds package

BioBuilds is distributed as a tarball for Debian, Ubuntu (both little-endian) and Red Hat Linux (big-endian) distributions. The tarball includes all binary files, their dependencies, and source code. The top-level directory in the tarball contains the following subdirectories:

```
install directory/bin
                 /include
                 /lib
                 /src
                 /share
                 /env.sh
                 /MD5SUMS.txt
```

[6] The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data, http://genome.cshlp.org/content/20/9/1297.full

The implementation scenario described in this book uses the Lab7 BioBuilds package components BWA, SAMtools, and Picard. Each of the programs has a set of commands and utilities that perform different tasks. In the next sections, you can read an overview of these selected applications.

BWA

The Burrows-Wheeler Alignment tool (BWA) is an open source, high-performance tool, and is available for no extra fee, 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. Example 3-9 shows the usage of BWA binary files and commands (methods) you can use.

Example 3-9 BWA usage

```
root@mn01:/gpfs/genomics/BioBuilds-2015.04/bin# ./bwa

Program: bwa (alignment via Burrows-Wheeler transformation)
Version: 0.7.12-r1039
Contact: Heng Li <lh3@sanger.ac.uk>

Usage:   bwa <command> [options]

Command: index         index sequences in the FASTA format
         mem           BWA-MEM algorithm
         fastmap       identify super-maximal exact matches
         pemerge       merge overlapping paired ends (EXPERIMENTAL)
         aln           gapped/ungapped alignment
         samse         generate alignment (single ended)
         sampe         generate alignment (paired ended)
         bwasw         BWA-SW for long queries

         shm           manage indices in shared memory
         fa2pac        convert FASTA to PAC format
         pac2bwt       generate BWT from PAC
         pac2bwtgen    alternative algorithm for generating BWT
         bwtupdate     update .bwt to the new format
         bwt2sa        generate SA from BWT and Occ

Note: To use BWA, you need to first index the genome with `bwa index'.
      There are three alignment algorithms in BWA: `mem', `bwasw', and
      `aln/samse/sampe'. If you are not sure which to use, try `bwa mem'
      first. Please `man ./bwa.1' for the manual.
```

SAMtools

The Sequence Alignment/Map tools (SAMtools) provide various utilities for manipulating alignments in the SAM format, including sorting, merging, indexing, and generating alignments in a per-position format. Example 3-10 presents the usage of SAMtools binary files and commands (methods) that you can use.

Example 3-10 SAMtools usage

```
root@mn01:/gpfs/genomics/BioBuilds-2015.04/bin# ./samtools

Program: samtools (Tools for alignments in the SAM format)
Version: 1.0 (using htslib 1.0)
```

```
Usage:   samtools <command> [options]

Commands:
  -- indexing
        faidx       index/extract FASTA
        index       index alignment
  -- editing
        calmd       recalculate MD/NM tags and '=' bases
        fixmate     fix mate information
        reheader    replace BAM header
        rmdup       remove PCR duplicates
        targetcut   cut fosmid regions (for fosmid pool only)
  -- file operations
        bamshuf     shuffle and group alignments by name
        cat         concatenate BAMs
        merge       merge sorted alignments
        mpileup     multi-way pileup
        sort        sort alignment file
        split       splits a file by read group
        bam2fq      converts a BAM to a FASTQ
  -- stats
        bedcov      read depth per BED region
        depth       compute the depth
        flagstat    simple stats
        idxstats    BAM index stats
        phase       phase heterozygotes
        stats       generate stats (former bamcheck)
  -- viewing
        flags       explain BAM flags
        tview       text alignment viewer
        view        SAM< BAM< CRAM conversion
```

Picard

Consists of Java-based command-line utilities that manipulate SAM files, and a Java API (SAM-JDK) for creating programs that read and write SAM files. Both SAM text format and SAM binary (BAM) format are supported.

For an overview of the Picard command-line tools, see the following website:

https://broadinstitute.github.io/picard/command-line-overview.html

Example 3-11 shows how you call the Picard JAR file and tools that you want use.

Example 3-11 Picard usage

```
root@mn01:/gpfs/genomics/BioBuilds-2015.04/share/java/picard# java -jar picard.jar

USAGE: PicardCommandLine <program name> [-h]

Available Programs:
--------------------------------------------------------------------------------
Fasta:                                          Tools for manipulating FASTA, or related
data.
    CreateSequenceDictionary                    Creates a SAM or BAM file from reference
sequence in fasta format
```

```
    ExtractSequences                            Extracts intervals from a reference
sequence, writing them to a FASTA file
    NormalizeFasta                              Normalizes lines of sequence in a fasta
file to be of the same length

--------------------------------------------------------------------------------
Illumina Tools:                                 Tools for manipulating data specific to
Illumina sequencers.
    CheckIlluminaDirectory                      Asserts the validity of the data in the
specified Illumina basecalling data
    CollectIlluminaBasecallingMetrics           Given an Illumina basecalling and a lane,
produces per-lane-barcode basecalling metrics
    CollectIlluminaLaneMetrics                  Collects Illumina lane metrics for the
given basecalling analysis directory
    CollectIlluminaSummaryMetrics               Collects summary metrics according to
Illumina specifications.
    ExtractIlluminaBarcodes                     Tool to determine the barcode for each
read in an Illumina lane
    IlluminaBasecallsToFastq                    Generate fastq file(s) from data in an
Illumina basecalls output directory
    IlluminaBasecallsToSam                     Generate a SAM or BAM file from data in an
Illumina basecalls output directory
    MarkIlluminaAdapters                        Reads a SAM or BAM file and rewrites it
with new adapter-trimming tags

--------------------------------------------------------------------------------
Interval Tools:                                 Tools for manipulating Picard interval
lists.
    BedToIntervalList                           Converts a BED file to an Picard Interval
List.
    IntervalListTools                           General tool for manipulating interval
lists
    LiftOverIntervalList                        Lifts over an interval list from one
reference build to another
    ScatterIntervalsByNs                       Writes an interval list based on splitting
the reference by Ns

--------------------------------------------------------------------------------
Metrics:                                        Tools for reporting metrics on various
data types.
    CalculateHsMetrics                          Calculates Hybrid Selection-specific
metrics for a SAM or BAM file
    CollectAlignmentSummaryMetrics              Produces from a SAM or BAM a file
containing summary alignment metrics
    CollectBaseDistributionByCycle              Program to chart the nucleotide
distribution per cycle in a SAM or BAM file.
    CollectGcBiasMetrics                        Collects information about GC bias in the
reads in the provided SAM or BAM
    CollectHiSeqXPfFailMetrics                  Classify PF-Failing reads in a HiSeqX
Illumina Basecalling directory into various categories.
    CollectInsertSizeMetrics                    Writes insert size distribution metrics
for a SAM or BAM file
    CollectJumpingLibraryMetrics                Produces jumping library metrics for the
provided SAM/BAMs
    CollectMultipleMetrics                      A "meta-metrics" calculating program that
produces multiple metrics for the provided SAM/BAM
    CollectOxoGMetrics                          Collects metrics quantifying the CpCG
CpCA error rate from the provided SAM/BAM
    CollectQualityYieldMetrics                  Collects a set of metrics that quantify
the quality and yield of sequence data from the provided SAM/BAM
```

```
    CollectRnaSeqMetrics                        Produces RNA alignment metrics for a SAM
or BAM file
    CollectRrbsMetrics                          Collects metrics about bisulfite
conversion for RRBS data
    CollectTargetedPcrMetrics                   Produces Targeted PCR-related metrics
given the provided SAM/BAM
    CollectWgsMetrics                           Writes whole genome sequencing-related
metrics for a SAM or BAM file
    EstimateLibraryComplexity                   Estimates library complexity from the
sequence of read pairs
    MeanQualityByCycle                          Writes mean quality by cycle for a SAM or
BAM file
    QualityScoreDistribution                    Charts quality score distributions for a
SAM or BAM file

--------------------------------------------------------------------------------
Miscellaneous Tools:                            A set of miscellaneous tools.
    BaitDesigner                                Designs baits or oligos for hybrid
selection reactions.
    FifoBuffer                                  FIFO buffer used to buffer input and
output streams with a customizable buffer size

--------------------------------------------------------------------------------
SAM/BAM:                                        Tools for manipulating SAM, BAM, or
related data.
    AddCommentsToBam                            Adds comments to the header of a BAM file
    AddOrReplaceReadGroups                      Replaces read groups in a BAM or SAM file
with a single new read group
    BamIndexStats                               Generates index statistics from a BAM file
    BamToBfq                                    Create BFQ files from a BAM file for use
by the Maq aligner.
    BuildBamIndex                               Generates a BAM index (.bai) file
    CalculateReadGroupChecksum                  Creates a hash code based on the read
groups (RG) in the SAM or BAM header.
    CheckTerminatorBlock                        Asserts the provided gzip file's (e.g.,
BAM) last block is well-formed; RC 100 otherwise
    CleanSam                                   Cleans the provided SAM/BAM, soft-clipping
beyond-end-of-reference alignments and setting MAPQ to 0 for unmapped reads
    CompareSAMs                                 Compares two input SAM or BAM files
    DownsampleSam                               Down-sample a SAM or BAM file to retain a
random subset of the reads
    FastqToSam                                  Converts a fastq file to an unaligned BAM
or SAM file
    FilterSamReads                             Creates a new SAM or BAM file by including
or excluding aligned reads
    FixMateInformation                          Ensure that all mate-pair information is
in sync between each read and its mate pair
    GatherBamFiles                              Concatenates one or more BAM files
together as efficiently as possible
    MarkDuplicates                              Examines aligned records in the supplied
SAM or BAM file to locate duplicate molecules.
    MarkDuplicatesWithMateCigar                 Examines aligned records in the supplied
SAM or BAM file to locate duplicate molecules.
    MergeBamAlignment                           Merges alignment data from a SAM or BAM
with data in an unmapped BAM file
    MergeSamFiles                               Merges multiple SAM or BAM files into one
file
    ReorderSam                                  Reorders reads in a SAM or BAM file to
match ordering in reference
```

```
    ReplaceSamHeader                              Replace the SAMFileHeader in a SAM file
with the given header
    RevertOriginalBaseQualitiesAndAddMateCigar   Reverts the original base qualities and
adds the mate cigar tag to read-group BAMs
    RevertSam                                     Reverts SAM or BAM files to a previous
state
    SamFormatConverter                           Convert a BAM file to a SAM file, or a SAM
to a BAM
    SamToFastq                                    Converts a SAM/BAM into a FASTQ
    SortSam                                       Sorts a SAM or BAM file
    SplitSamByLibrary                             Splits a SAM or BAM file into individual
files by library
    ValidateSamFile                               Validates a SAM or BAM file
    ViewSam                                       Prints a SAM or BAM file to the screen

--------------------------------------------------------------------------------
VCF/BCF:                                          Tools for manipulating VCF, BCF, or
related data.
    FilterVcf                                     Hard filters a VCF.
    GatherVcfs                                    Gathers multiple VCF files from a scatter
operation into a single VCF file
    GenotypeConcordance                           Calculates the concordance between
genotype data for two samples in two different VCFs
    MakeSitesOnlyVcf                              Creates a VCF bereft of genotype
information from an input VCF or BCF
    MergeVcfs                                     Merges multiple VCF or BCF files into one
VCF file or BCF
    RenameSampleInVcf                             Rename a sample within a VCF or BCF.
    SortVcf                                       Sorts one or more VCF files
    SplitVcfs                                     Splits an input VCF or BCF file into two
VCF or BCF files
    UpdateVcfSequenceDictionary                   Takes a VCF and a second file that
contains a sequence dictionary and updates the VCF with the new sequence dictionary.
    VcfFormatConverter                            Converts a VCF file to a BCF file, or BCF
to VCF
    VcfToIntervalList                             Converts a VCF or BCF file to a Picard
Interval List.
```

3.4.2 GATK

The Genome Analysis Toolkit open source version (GATK) was created using the functional programming paradigm of MapReduce, which supports generating large data sets. Programs written in this functional style are automatically parallelized and run on a large cluster of commodity machines. The runtime system takes care of the details of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing the required inter-machine communication.

Dependencies

GATK requires Java 1.7+ OpenJDK.

Installation

Create a directory in the shared file system where it is accessible by all of the management and compute nodes. Copy the GATK package to that directory and extract it using the **tar** command as shown in Example 3-12.

Example 3-12 Extracting GATK binary files

```
$ tar xvjf GenomeAnalysisTK-3.4-46.tar.bz2
```

Test the installation by confirming that the GATK installation was successful with the following test. Test as both a root user and as a non-root user as shown in Example 3-13.

Example 3-13 Testing GATK installation

```
$ cd /gpfs/gpfs02/gatk3.4
$ java -jar GenomeAnalysisTK.jar --help
$ su -c "java -jar GenomeAnalysisTK.jar --help" lsfadmin
```

Basic Java syntax

For information about the basic Java syntax to use the GATK tool, see the following website:

https://www.broadinstitute.org/gatk/guide/article?id=4669

The commands for GATK always follow the same basic syntax:

java [Java arguments] -jar GenomeAnalysisTK.jar [GATK arguments]

The core of the command is **java -jar GenomeAnalysisTK.jar**, which starts the GATK program in a Java virtual machine (JVM). Any additional java-specific arguments (such as **-Xmx** to increase memory allocation) can be inserted between **java** and **-jar** as follows:

java -Xmx4G -jar GenomeAnalysisTK.jar [GATK arguments]

The order of arguments between **java** and **-jar** are not important.

GATK command arguments[7]

Two universal arguments are required for every GATK command (with few exceptions, the **clp**-type utilities), **-R** for Reference (for example, **-R human_b37.fasta**), and **-T** for Tool name (for example, **-T HaplotypeCaller**).

Additional arguments fall in two categories:

- Engine arguments like **-L** (for specifying a list of intervals) can be given to all tools. They are technically optional, bu they can be effectively required at certain steps for specific analytical designs (for example, the **-L** argument for calling variants on exomes).
- Tool-specific arguments, which can be required like **-I** (to provide an input file that contains sequence reads to tools that process BAM files), or optional such as **-alleles** (to provide a list of known alleles for genotyping).

All available engine and tool-specific arguments are listed in the tool documentation section at:

https://www.broadinstitute.org/gatk/guide/tooldocs/

Flags are arguments that have Boolean values, such as TRUE or FALSE. They are typically used to enable or disable specific features, such as **--keep_program_records** makes certain GATK tools output additional information in the BAM header that can be omitted otherwise. In

[7] https://www.broadinstitute.org/gatk/guide/article?id=4669

GATK, all flags are set to FALSE by default, so if you want to set one to TRUE, add the flag name to the command. You do not need to specify an actual value.

Examples of GATK commands

Example 3-14 shows a simple command that runs *HaplotypeCaller* in default mode on a single input BAM file containing sequence data, and outputs a VCF file containing raw variants.

Example 3-14 GATK example run from CLI

```
$ java -Xmx4G -jar GenomeAnalysisTK.jar -R human_b37.fasta -T HaplotypeCaller -I
sample1.bam -o raw_variants.vcf
```

Optional: Additional commands

If the data is from exome sequencing, provide the exome targets by using the **-L** argument as shown in Example 3-15.

Example 3-15 Providing exome targets on GATK

```
$ java -Xmx4G -jar GenomeAnalysisTK.jar -R human_b37.fasta -T HaplotypeCaller -I
sample1.bam -o raw_variants.vcf -L exome_intervals.list
```

If you want to genotype specific sites of interest using known alleles based on results from a previous study, change the HaplotypeCaller genotyping mode by using **-gt_mode**, provide those alleles by using **-alleles**, and restrict the analysis to just those sites by using **-L** as shown in Example 3-16.

Example 3-16 Using alleles from a previous study on GATK

```
$ java -Xmx4G -jar GenomeAnalysisTK.jar -R human_b37.fasta -T HaplotypeCaller -I
sample1.bam -o raw_variants.vcf -L known_alleles.vcf -alleles known_alleles.vcf
-gt_mode GENOTYPE_GIVEN_ALLELES
```

For more examples of commands and for specific tool commands, see the tool documentation section at the following website:

https://www.broadinstitute.org/gatk/guide/tooldocs/

3.5 IBM Life Science Platform Provisioning package

IBM provides the IBM Life Science solution with provisioning package. The purpose of this package is to install, configure, and start Platform LSF, Platform PPM, and PAC to create a fully functional HPC Genomic Life Science Cluster. Sample PPM genomic workflows are then integrated into the PAC environment and application templates become available to the administrator.

Example 3-17 shows the LSSol3.0.tar package after its files are extracted using the **tar** command.

Example 3-17 Extracting LSSol 3.0 software

```
# tar -xvf LSSolInst3.0.tar
```

The software package contains installation and deployment tools only. It does not include any platform or open source products. The files in the package are shown in Example 3-18.

Example 3-18 LSSol 3.0 software contents

```
igwa-lifescience-sample-prov-3.0.0.0-0.all.deb
igwa-lifescience-sample-ppm-3.0.0.0-0.all.deb
LS-Install.xml
DataEngineforGenomics-PowerSystemsEdition-Runbook-Version3.0.pdf
LSSolInst3.0.README
```

The description of these files are shown in Example 3-19.

Example 3-19 Description of files included in the LSSolInst3.0 package

```
LSSolInst3.0.README - Package description

DataEngineforGenomics-PowerSystemsEdition-Runbook-Version3.0.pdf - The runbook
guide to create a life science cluster on Power 8 systems

igwa-lifescience-sample-prov-3.0.0.0-0.all.deb - The provisioning package to
deploy a Life Science cluster which allows to provision LifeScience cluster
through Platform Cluster Manager or allows to installation LifeScience cluster
manually.

igwa-lifescience-sample-ppm-3.0.0.0-0.all.deb - The workflow package to deploy
sample life science workflows to the cluster

LS-Install.xml - The Platform Cluster Manager (Platform Cluster Manager) template
to provision alife science cluster
```

3.5.1 Installation of sample life science workflows

To deploy sample life science workflows to the cluster, select a package that contains PPM workflows (in this example, `igwa-lifescience-sample-ppm-3.0.0.0-0.all.deb`) and install it as shown in Example 3-20.

Example 3-20 Installing LSSol 3.0

```
# dpkg -i igwa-lifescience-sample-ppm-3.0.0.0-0.all.deb
Selecting previously unselected package igwa-lifescience-sample-ppm.
(Reading database ... 110449 files and directories currently installed.)
Preparing to unpack igwa-lifescience-sample-ppm-3.0.0.0-0.all.deb ...
Unpacking igwa-lifescience-sample-ppm (3.0.0.0-0) ...
Setting up igwa-lifescience-sample-ppm (3.0.0.0-0) ...
```

The solution files are placed in the `/opt/ibm/lifescience` directory. Move them to the subdirectory `samples` to explore the files as shown in Example 3-21.

Example 3-21 Content of the samples directory

```
root@mn01:/opt/ibm/lifescience/samples# l
common/           ppmLSinstall.config*  ppm_modules/
pac_templates/    ppmLSinstall.py*      ppm_workflows/
ppm_executables/  ppmLSinstall.README
```

The ppmLSinstall.py installation script integrates the open source paths into the workflows and deploys them into the PAC and PPM environment.

The install script must run on the master head node where PPM and PAC are installed. First, update ppmLSinstall.config to specify the environment details. The install script takes values from the configuration file and deploys the workflows into PAC and PPM. The workflow scripts that are used for the PPM workflows are stored in the directory <PPMSCRIPT_DIR>/ppm_executables. The xmls for PAC are copied to the directory <PAC_TOP>/gui/conf/application/draft.

Note: The open source and PPMSCRIPTS_DIR must be installed on a shared file system. The Master untars the open source software and one of the compute nodes does the make. If the PPMSCRIPTS_DIR+"/pac/scripts/" directory already exists, it is deleted and replaced by the installation.

If PAC already has the HPC sample workflow, it is moved to ~/gui/conf/application/WORKFLOW_NAME_date_timestamp.

Update the ppmLSinstall.config file. Uncomment the lines with the relevant variables, entering the values for these variables that are appropriate for the particular installation. The full paths to the application binary files recorded earlier are used in this step. Example 3-22 shows a possible arrangement of configuration values.

Example 3-22 ppmLSinstall.config file content

```
#************************************************************
[ppmLSinstall]
#************************************************************
#
# --  IBM Platform Products Information  --
#
# -----------------
SAMPLE_TOP=/opt/ibm/lifescience/samples
# -----------------
# Full path to the top-level IBM HPC Life Science sample directory {REQUIRED}
# Path the sample workflow engine is installed into on the PAC/PPM server node
# -----------------
PAC_TOP=/opt/pac
# -----------------
# Full path to the top-level IBM Platform Application Center (PAC) directory
{REQUIRED}
# Put PAC onto local directory.  Can be modified to put into a shared directory.
# -----------------
PAC_ADMIN=lsfadmin
# -----------------
# PAC administrator ID {REQUIRED}
#
# -----------------
PPM_TOP=/gpfs/ppm
# -----------------
# Full path to the top-level IBM Platform Process Manager (PPM) directory
{REQUIRED}
#
# -----------------
PPM_ADMIN=lsfadmin
# -----------------
```

```
# PPM Administrator ID {REQUIRED}
#
# --  IBM Life Science Sample Workflows Information  --
#
# -----------------
PPMSCRIPTS_DIR=/gpfs/genomics
# -----------------
# Full path to the workflow scripts directory {REQUIRED}
#
# This directory path is where all the IBM HPC Life Science workflow scripts will
# stored. PPMSCRIPTS_DIR must be on a shared file system and be accessible. If the
# given directory already exists, it will be removed and replaced.
#
# -----------------
NUM_PROCS=4
# -----------------
# Number of threads for workflow {REQUIRED}
#
# -----------------
REQ_MEM=4096
# -----------------
# Required memory for workflow in MB {REQUIRED}
#
#
# --  Direct Open Source Executable Paths --
#
# If an executable other than one found in OPEN_SOURCE_SHARED_DIRECTORY is
desired,
# define the full path to the executable in this section.
#
# -----------------
BWA_DIR_EXE=/gpfs/genomics/BioBuilds-2015.04/bin/bwa
# -----------------
# Full path of the BWA executable.(file)
#
# -----------------
SAMTOOLS_DIR_EXE=/gpfs/genomics/BioBuilds-2015.04/bin/samtools
# -----------------
# Full path of the Samtools executable.(file)
#
# -----------------
PICARD_DIR=/gpfs/genomics/BioBuilds-2015.04/share/java/picard
# -----------------
# Full path of the Picard Tools jar file.(directory)
#
# -----------------
GATK_DIR_EXE=/gpfs/genomics/GATK/GenomeAnalysisTK.jar
# -----------------
# Full path of the GenomeAnalysisTKLite jar file.(file)
#
#************************************************************
#                      END OF FILE
#************************************************************
```

NUM_PROC is a number of threads for workflow. REQ_MEM is required memory for workflow in MB. You can change NUM_PROC and REQ_MEM values for each individual pipeline that is imported in by changing the PAC templates. When you change the values, the change is propagated to all the pipelines when importing it.

After you are ready for installation, run the installation script as shown in Example 3-23. After the successful installation, you can see the list of updated flows and the outcomes of the operation.

Example 3-23 Installation of workflows

```
root@mn01:/opt/ibm/lifescience/samples# ./ppmLSinstall.py
INFO: Copying /opt/ibm/lifescience/samples/ppm_executables scripts to
/gpfs/genomics/ppm_executables
Flow <lsfadmin:MOD_BWA_ALN_igP01> is updated. Version <1.0>.
Flow <lsfadmin:MOD_BWA_ALN_igP01> is published.

Flow <lsfadmin:MOD_SAMTOOLS_igP01> is updated. Version <1.0>.
Flow <lsfadmin:MOD_SAMTOOLS_igP01> is published.

Flow <lsfadmin:MOD_ppm_setenv_igP01> is updated. Version <1.0>.
Flow <lsfadmin:MOD_ppm_setenv_igP01> is published.

Flow <lsfadmin:MOD_EXOME_GATK_igP01> is updated. Version <1.0>.
Flow <lsfadmin:MOD_EXOME_GATK_igP01> is published.

Flow <lsfadmin:MOD_checkCWDdir_igP01> is updated. Version <1.0>.
Flow <lsfadmin:MOD_checkCWDdir_igP01> is published.

Flow <lsfadmin:MOD_BWA_MEM_igP01> is updated. Version <1.0>.
Flow <lsfadmin:MOD_BWA_MEM_igP01> is published.

Flow <lsfadmin:MOD_PICARD_igP01> is updated. Version <1.0>.
Flow <lsfadmin:MOD_PICARD_igP01> is published.

Flow <lsfadmin:BWA_ALN_igP01> is updated. Version <1.0>.
Flow <lsfadmin:BWA_ALN_igP01> is published.

Flow <lsfadmin:EXOME_GATK_igP01> is updated. Version <1.0>.
Flow <lsfadmin:EXOME_GATK_igP01> is published.

Flow <lsfadmin:IGWA_WHOLE_GENOME_igP01> is updated. Version <1.0>.
Flow <lsfadmin:IGWA_WHOLE_GENOME_igP01> is published.

Flow <lsfadmin:IGWA_EXOME_igP01> is updated. Version <1.0>.
Flow <lsfadmin:IGWA_EXOME_igP01> is published.

Flow <lsfadmin:Trigger_BWA_MEM_igP01> is updated. Version <1.0>.
Flow <lsfadmin:Trigger_BWA_MEM_igP01> is published.

Flow <lsfadmin:ReferenceIndexing_igP01> is updated. Version <1.0>.
Flow <lsfadmin:ReferenceIndexing_igP01> is published.

Flow <lsfadmin:BWA_MEM_SAMTOOLS_DynSub_igP01> is updated. Version <1.0>.
Flow <lsfadmin:BWA_MEM_SAMTOOLS_DynSub_igP01> is published.
```

```
Flow <lsfadmin:SAMTOOLS_igPO1> is updated. Version <1.0>.
Flow <lsfadmin:SAMTOOLS_igPO1> is published.

Flow <lsfadmin:BWA_MEM_DynSub_igPO1> is updated. Version <1.0>.
Flow <lsfadmin:BWA_MEM_DynSub_igPO1> is published.

Flow <lsfadmin:PICARD_igPO2> is updated. Version <1.0>.
Flow <lsfadmin:PICARD_igPO2> is published.

Install script 'ppmLSinstall.py' completed successfully!
```

The workflow scripts used for the PPM workflows are stored into the directory <PPMSCRIPT_DIR>/ppm_executables. Example 3-24 shows the content of the directory.

Example 3-24 PPM workflow scripts

```
root@mn01:/gpfs/genomics/ppm_executables# ls
Base_sample_env_variables.txt
BWA_MEM_sample_variable_inputfile.txt
BWA_MEM_SAMTOOLS_sample_variable_inputfile.txt
BWA_OPT
GATK_OPT
hpcc.py
igwa_envFile_igPO1
PICARD_OPT
ppm_bwa_setvar.py
ppm_checkdir.py
ppm_common.py
ppm_exome_gatk_setvar.py
ppm_fastq_vcf_setvar.py
ppm_generateFilelist.py
ppm_picard_setvar.py
ppm_refindex_setvar.py
ppm_samtools_merge_bam.py
ppm_samtools_setvar.py
ppm_setSamlist.py
ppm_setvar.py
SAMTOOLS_OPT
Trigger_BWA_MEM_variable_input.txt
```

The xmls for PAC have been copied to <PAC_TOP>/gui/conf/application/draft. See the content of the directory in Example 3-25. The published PAC templates are placed in /opt/pac/gui/conf/application/published.

Example 3-25 PAC xmls

```
root@mn01:/opt/pac/gui/conf/application/draft# ls
ABAQUS                      FLUENT
ANSYS                       GATK_igTO2
AppDCV                      HMMER
AppDCVonLinux               HP-RGS
BLAST                       IGWA_EXOME_igTO1
BWA_ALN_igTO1               IGWA_WHOLE_GENOME_igTO1
BWA_MEM_igTO1               JOBFLOW_TEMPLATE_Sample
CATIA                       LS-DYNA
CFX                         MATLAB
```

```
ClustalW                    NASTRAN
CMGL_GEM                    NWCHEM
CMGL_IMEX                   PICARD_igT01
CMGL_STARS                  ReferenceIndexing_igT
ECLIPSE                     SAMTOOLS_igT01
EXCEED_Desktop              STAR-CCM
EXCEED_Desktop_Enterprise   TurboVNC
EXCEED_Xterm                Windows_Remote_Desktop
EXCEED_Xterm_Enterprise
```

Verifying the installation

After the successful installation and after these workflows are submitted to PAC and PPM, check the application template to verify that these entries are correct and all needed fields are completed properly.

Log in to the PAC GUI to verify that the PAC template and PPM pipelines are installed as shown in Figure 3-5 and Figure 3-6 on page 82.

Click the Resources tab and select **Submission Resources** → **Submission Templates** to list the sample PAC submission templates for genomics as shown in Figure 3-5. The names of the sample templates end with *_igT* (for example, `ReferenceIndexing_igT`).

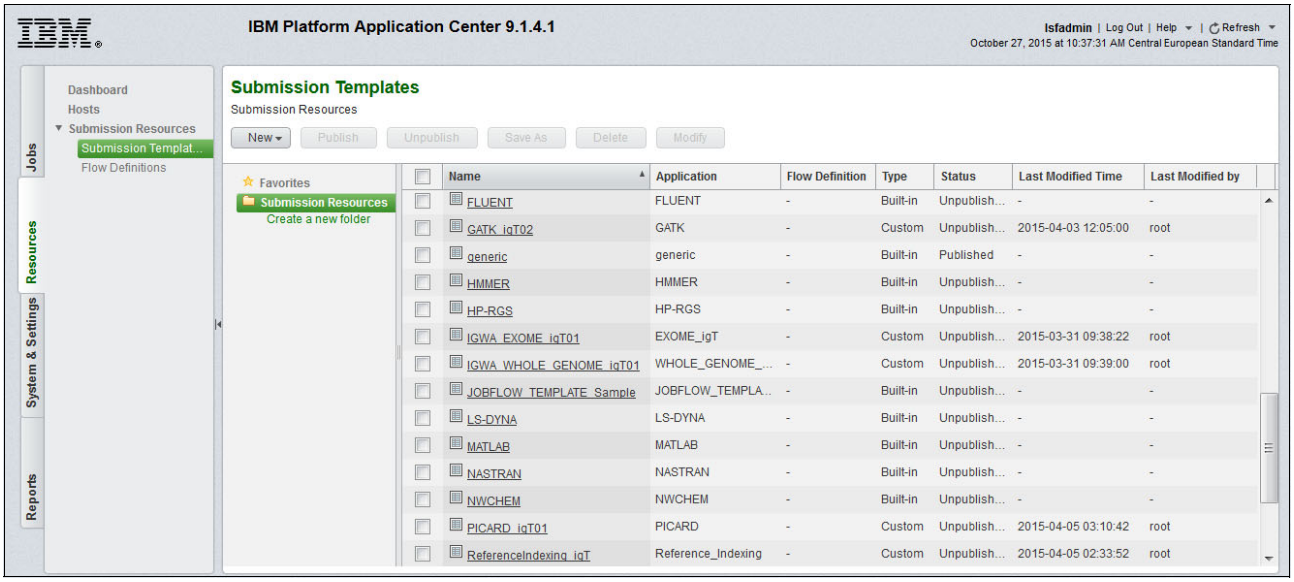


Figure 3-5 View of the PAC templates after installation

Click the Resources tab and select **Submission Resources** → **Flow Definitions** to list the sample PPM workflows for genomics as shown in Figure 3-6. The names of the sample workflows (pipelines) end with *_igP* (for example, `ReferenceIndexing_igP01`).

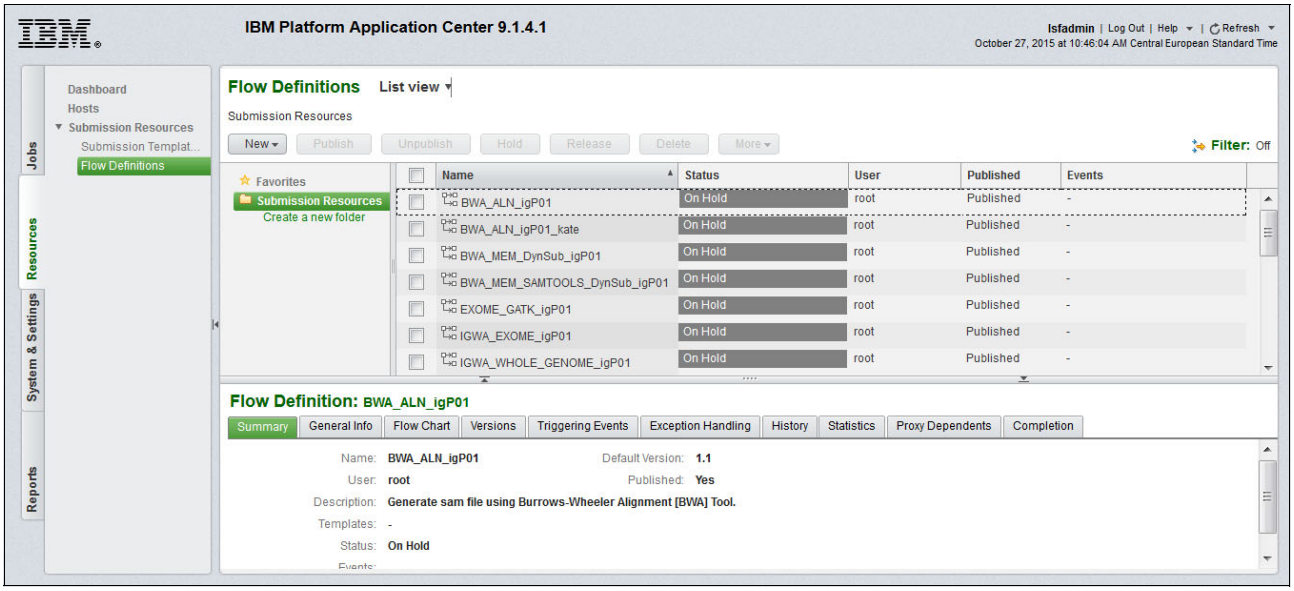


Figure 3-6 View of the PPM pipelines after installation

For the first workflows installation, the administrator must set the permission of the workflows to the correct users through the PAC GUI window as shown in Figure 3-7 on page 83.

For more information, see the PAC documentation on how to set the proper permissions at the following website:

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

3.5.2 User management and permissions

After the cluster is up and running the Platform Computing software, the sample genomic workflows are installed along with IBM Life Science Platform Provisioning package and made available in the PAC. The administrator must review and customize the sample workflows. After verifying all workflows, and setting the correct roles and permissions for usage, the workflows can be published by the administrator.

Figure 3-7 shows the genomic workflows from the user perspective. This workflow involves two roles: Administrator and user. The distinction is based on the role in the organization or project of the particular person assigned to the task.

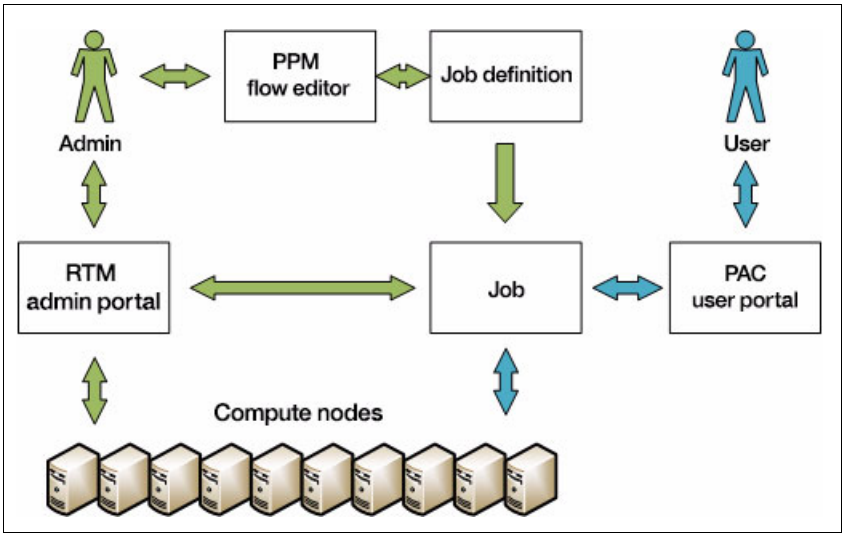


Figure 3-7 Genomic workflows from Platform Computing users perspective

You do not create users or user groups in the PAC. It automatically loads users and user groups defined in LSF. This load is controlled with the parameter ENABLE_USERGROUP in `/opt/pac/gui/conf/pmc.conf`.

Users and groups are preinstalled when you start Platform Application Center, and are updated every 3600 seconds. You can configure this interval in the configuration file `$GUI_CONFDIR/pmc.conf` with the parameter ACL_SYNC_INTERVAL. When this parameter is set to `0`, user and group synchronization from LSF is disabled.

To give permissions to users and user groups, you define user roles. A user role is a common group of permissions. You can assign more than one role to a user or user group. When multiple roles are assigned, the user or user group receives all the permissions defined for all the roles.

Platform Application Center has several predefined user roles with default permission settings. If you create a user role, the new user gets the same permissions as the normal user role. You can add permissions to the following built-in roles:

- Normal user
- Cluster administrator
- Administrator of user group
- Flow administrator
- Flow control administrator
- Report administrator

This book focuses on two main roles as described in the following sections.

Normal user role

The following are the characteristics of the normal user role:

- View all jobs submitted by all users.
- Control only jobs that the application user submitted.
- View and use job submission forms created by other users.
- View Job Flow definitions

Flow administrator role

The flow administrator role has the same permissions as the normal user, but holds these additional permissions as well:

- View, control, configure all job definitions and job flows.
- Set the correct roles and permissions for workflow usage.

Platform Application Center has a flexible access control mechanism. You can control access to all pages of the web interface, and also control whether users can take actions on objects in pages. For additional information, check the section objects and available permissions at:

http://ibm.co/1SxbOFD

3.5.3 Sample genomic workflow customization

To start your work with the genomic software and take the advantage of Platform Computing software built for POWER8, you can use the sample templates and customize them according to your needs.

You can modify the sample templates directly, or you can copy a sample and modify the copy to create a new template. Generally, copy a template and modify it to keep the original files intact for further usage or reference. To do so, copy the sample of interest, and set the proper environment values so that it can run in your environment.

To demonstrate the workflow, the examples use the GATK collection of standard files for working with human resequencing data, Reference and SNP b37 package. These files can be downloaded from the following website:

https://www.broadinstitute.org/gatk/guide/article.php?id=1213

The following sections provide examples to customize, publish, and submit the *Reference* and *Exome* sample templates.

Example: Reference

This section uses the `ReferenceIndexing_igT` template. This genome reference indexing picks the FASTA file and creates the dictionary and indexing files that are used for the genome sequencing workflow.

]

Commands used in the workflow:

BWA:

```
bwa index -a bwtsw
```

PICARD:

```
java -XmxJAVA_REQMEM_VALUEg -XX:-UseGCOverheadLimit -XX:-UseParallelGC
-Djava.io.tmpdir PICARD/CreateSequenceDictionary TMP_DIR
MAX_RECORDS_IN_RAM=REFINDEX_MAX_RECORD_INRAM_VALUE"
```

SAMTOOLS:

```
samtools faidx
```

Local working directory

Before working with the templates, create a local directory where the user stores data and saves results as shown in Example 3-26.

Example 3-26 Create local working directory

```
root@mn01:/gpfs/home/lsfadmin# su - lsfadmin
$ mkdir localReference
$ ls -la
drwxr-xr-x 9 lsfadmin lsfadmin 4096 Nov 30 02:07 .
drwxr-xr-x 3 root     root     4096 Oct 10 03:48 ..
drwxrwxr-x 2 lsfadmin lsfadmin 4096 Nov 30 02:07 localReference
```

Example 3-26 uses `/gpfs/home/lsfadmin/localReference`.

Prepare genomic data

Move files into the local work directory. Example 3-27 shows moving the `human_g1k_v37_decoy.fasta` file.

Example 3-27 Moving the human_glk_v37_decoy.fast file

```
root@mn01:/gpfs/genomics/data/broadinstitute/b37# cp human_g1k_v37_decoy.fasta
/gpfs/home/lsfadmin/localReference
root@mn01:/gpfs/home/lsfadmin/localReference# chown lsfadmin:lsfadmin
human_g1k_v37_decoy.fasta
```

Template customization

To start using PAC templates, click the Resources tab and select **Submission Resources** → **Submission Templates**. The view corresponds to the content of the `<PAC_TOP>/gui/conf/application/draft` directory, which in the example implementation is `/opt/pac/gui/conf/application/draft`. This example uses the `ReferenceIndexing_igT` template.

Select `ReferenceIndexing_igT` and click **Save as** to create a copy of the template, named for example `ReferenceIndexing_igT_test1`. A new directory with the same name `ReferenceIndexing_igT_test1` is created in the `/opt/pac/gui/conf/application/draft` directory with two files in it as shown in Example 3-28.

Example 3-28 Template files in (..)/draft/ReferenceIndexing_igT_test1 directory

```
ReferenceIndexing_igT_test1.cmd
ReferenceIndexing_igT_test1.xml
```

At the same time, the new template appears in the Submission Templates view. Now you can freely modify the copied template.

Tip: Notice that the Job Flow Owner [JOBFLOW_OWNER] field changes and needs to be updated to the desired value after copying the template.

Select the `ReferenceIndexing_igT_test1` template, and click **Modify**. You see a window as shown in Figure 3-8.

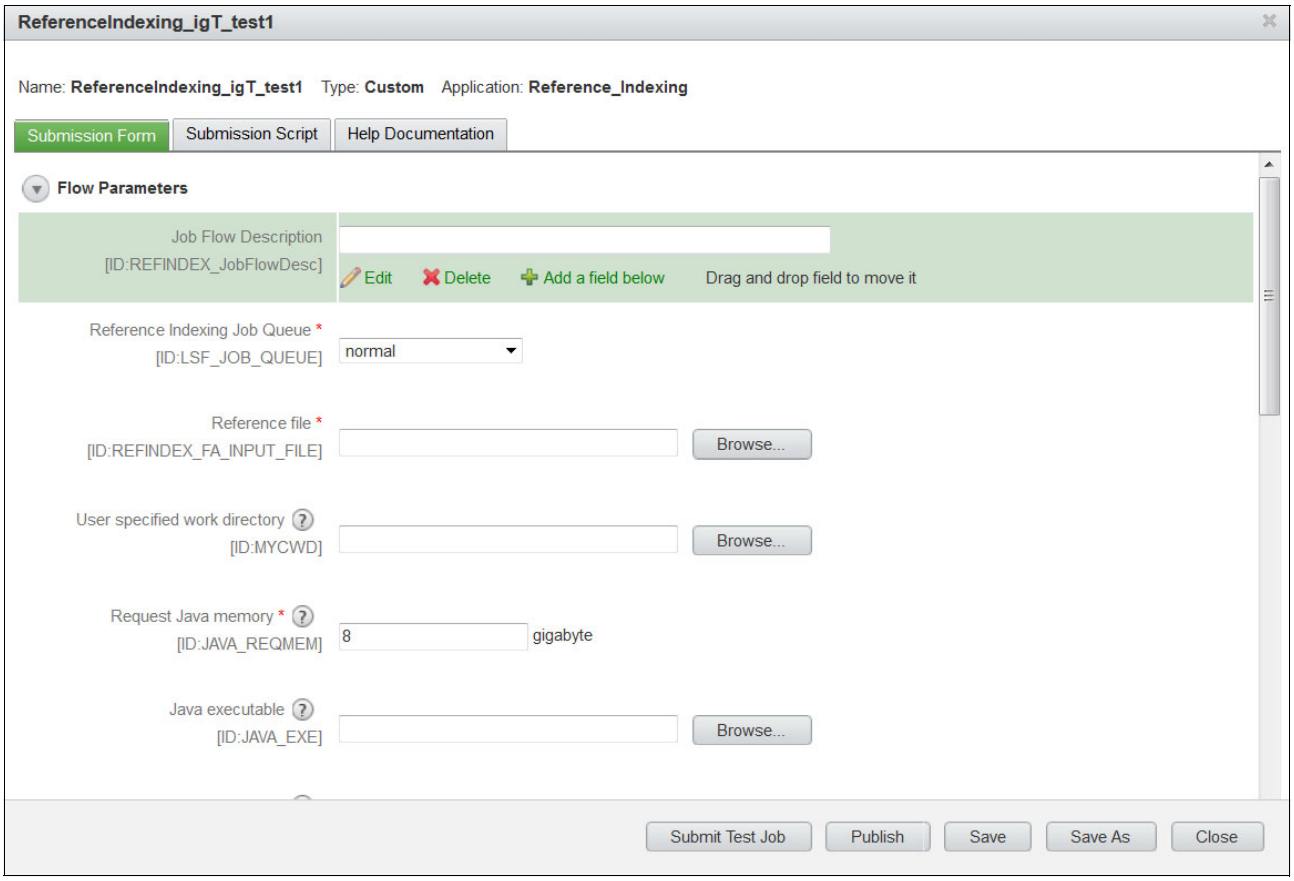


Figure 3-8 Modifying the template

The Submission Form tab holds flow parameters that are defined in the directory dire`ReferenceIndexing_igT_test1.xml`.

In the Submission Script tab as shown in Figure 3-9, you can modify the content of an application script that applies to the `ReferenceIndexing_igT_test1.cmd` file found in the `/opt/pac/gui/conf/application/draft/ReferenceIndexing_igT_test1` directory.

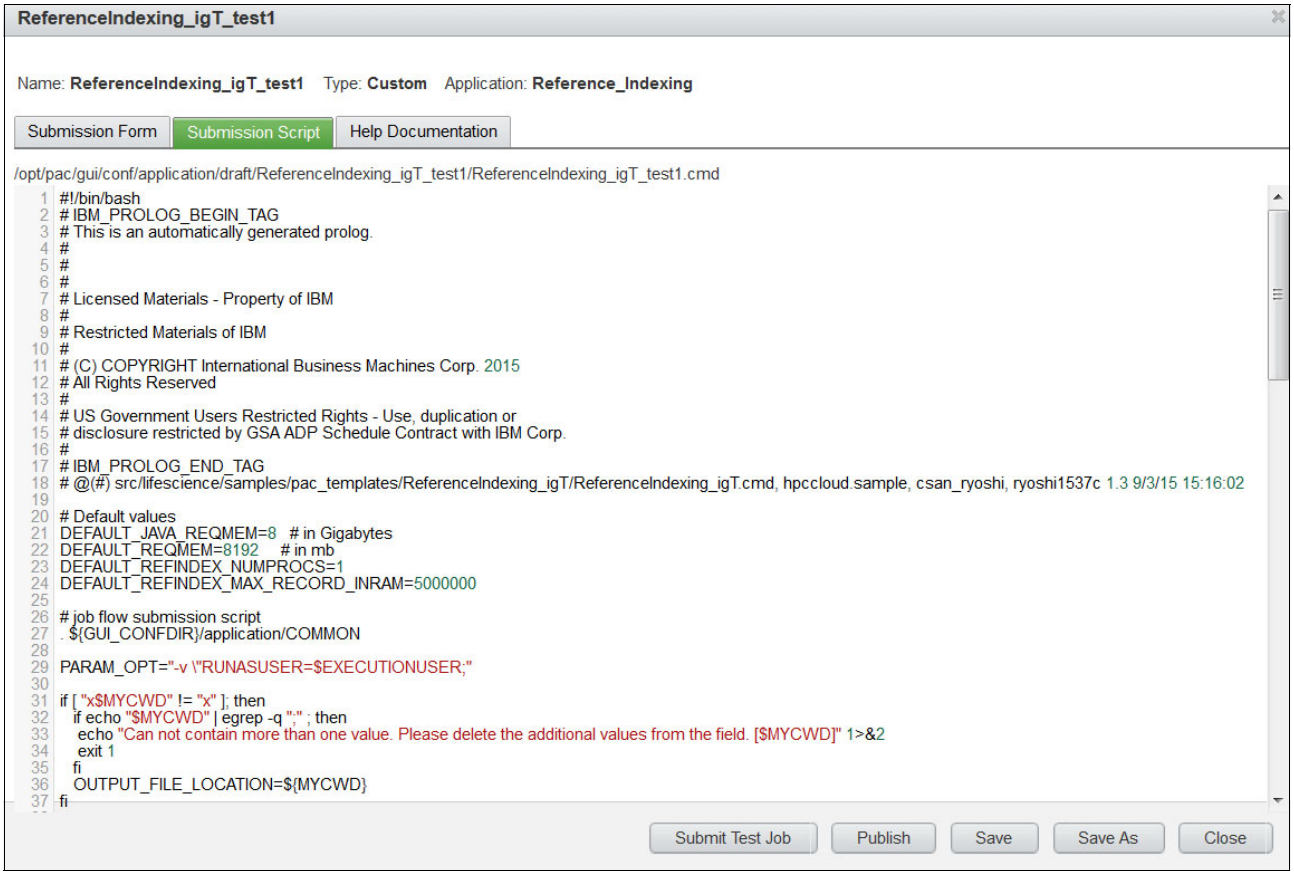


Figure 3-9 Submission script tab

The Help Documentation tab explains the purpose of the template, tools, and commands used as shown in Figure 3-9.

To check what flow definition is set in the template, look at the Submission Form for the REFINDEX_HOBFLOW_NAME variable value as shown in Figure 3-10.

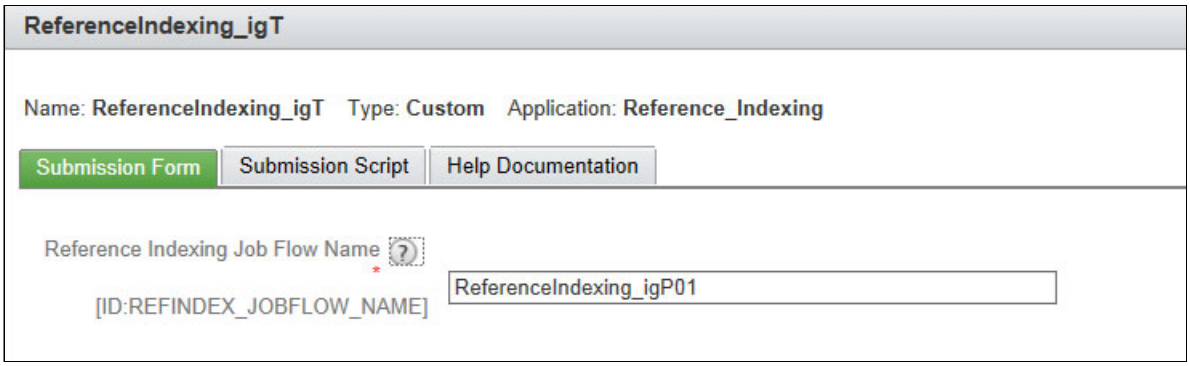


Figure 3-10 Reference Indexing Job Flow name

To understand what is supposed to be processed, display the flow under the `ReferenceIndexing_igT` template. Click the Resource tab, and select **Submission Resources** → **Flow Definitions** and click the **Flow Chart** tab as shown in Figure 3-11.

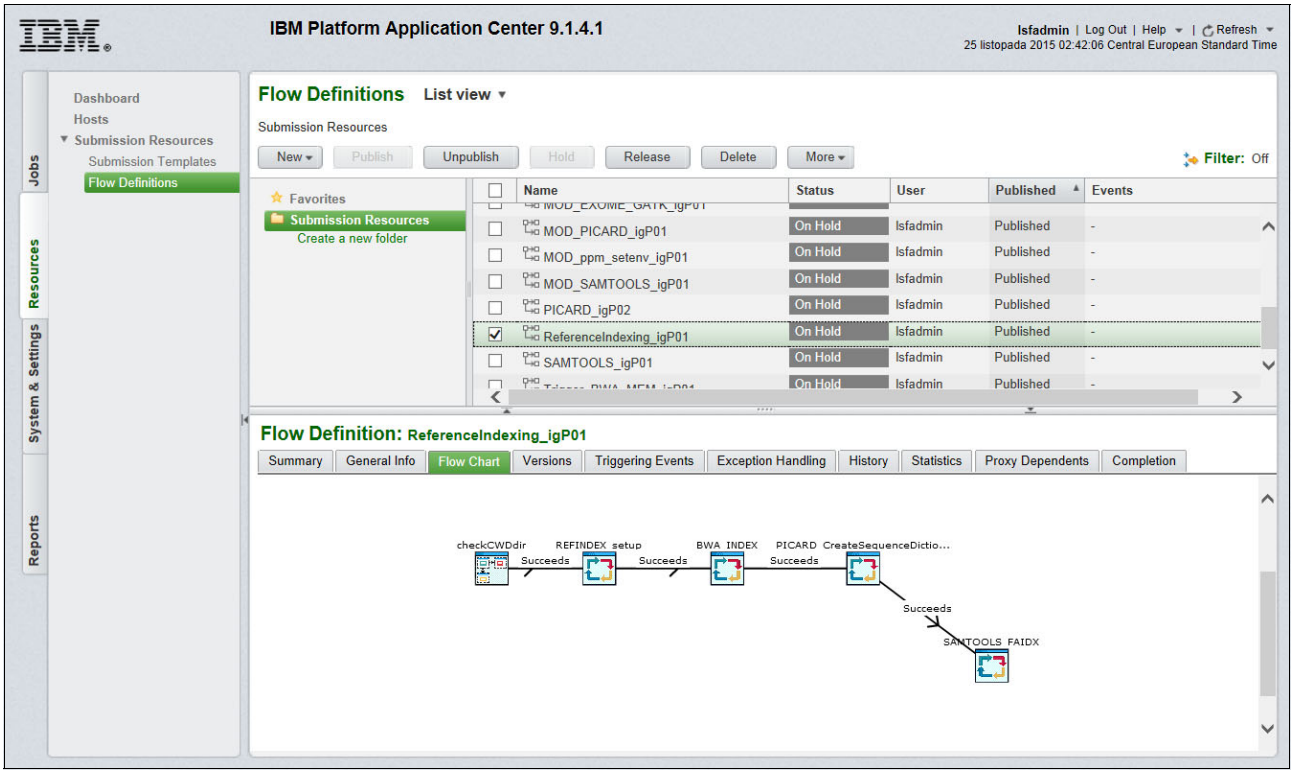


Figure 3-11 ReferenceIndexing_igT flow chart

Tip: To list all templates, click the Resources tab, and select **Submission Resources** → **Submission Templates**.

While customizing a template, examine all of the fields that the administrator wants as defaults, and set them before publishing the template for users. The fields can be marked with these indicators:

- A red asterisk indicates a mandatory field.
- A question mark can be clicked for more information.
- A blank field that when set to be visible can be completed by a user.

Check the values and make any updates. The application reads data from and writes data to the same directory in this example.

As shown in Figure 3-12 on page 89, for the `ReferenceIndexing_igT_test1` template the user provides this information:

- Flow description [REFINDEX_JobFlowDesc]: The user describes the job flow.
- Reference file [REFINDEX_FA_INPUT_FILE]: A field where the fasta file can be provided, for example the `human_g1k_v37_decoy.fasta` file.
- User specified work directory [MYCWD]: The working directory created in the `lsfadmin` directory, which is `/gpfs/home/lsfadmin/localReference`.
- Job Flow Owner [JOBFLOW_OWNER]: This example uses `lsfadmin`.

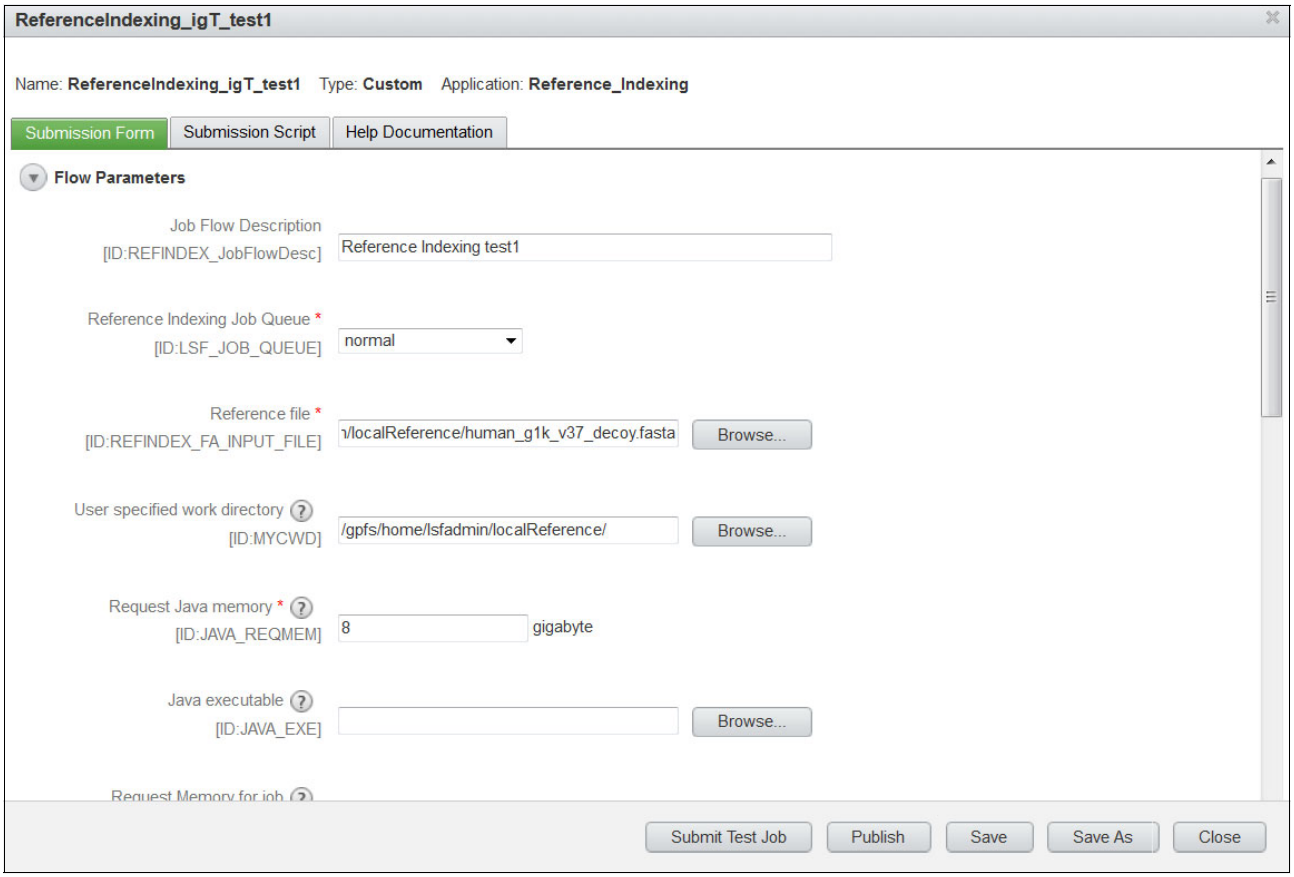


Figure 3-12 Customization of ReferenceIndexing_igT_test1 template

Run a test by completing the values and clicking **Submit Test Job**. A fragment of the output of the submitted application is shown in Example 3-29.

Example 3-29 Running and submitting a test job

```
+ DEFAULT_JAVA_REQMEM=8
+ DEFAULT_REQMEM=8192
+ DEFAULT_REFINDEX_NUMPROCS=1
+ DEFAULT_REFINDEX_MAX_RECORD_INRAM=5000000
+ . /opt/pac/gui/conf/application/COMMON
+ PARAM_OPT=-v "RUNASUSER=lsfadmin;
+ [ x/gpfs/home/lsfadmin/localReference/ != x ]
+ echo /gpfs/home/lsfadmin/localReference/
+ egrep -q ;
+ PARAM_OPT=-v
"RUNASUSER=lsfadmin;JS_FLOW_WORKING_DIR=/gpfs/home/lsfadmin/localReference/;
+ [ x/gpfs/home/lsfadmin/localReference/human_g1k_v37_decoy.fasta != x ]
+ echo /gpfs/home/lsfadmin/localReference/human_g1k_v37_decoy.fasta
+ egrep -q ;
+ PARAM_OPT=-v
"RUNASUSER=lsfadmin;JS_FLOW_WORKING_DIR=/gpfs/home/lsfadmin/localReference/;REFIND
EX_FA_INPUT_FILE=/gpfs/home/lsfadmin/localReference/human_g1k_v37_decoy.fasta;
+ basename /gpfs/home/lsfadmin/localReference/human_g1k_v37_decoy.fasta
...
+ JOBFLOW_RESULT=Flow  is triggered: Flow id <8>.
```

```
+ export JOBFLOW_RESULT OUTPUT_FILE_LOCATION
+ /opt/pac/gui/conf/application/jobflow-result.sh
```

At the end of the output, the line `JOBFLOW_RESULT=Flow is triggered: Flow id <8>` means that the run was successful. You can publish the PAC template by clicking **Publish**, which opens the window shown in Figure 3-13. Click **Yes** to continue. After you publish it, the template is available to the other users.

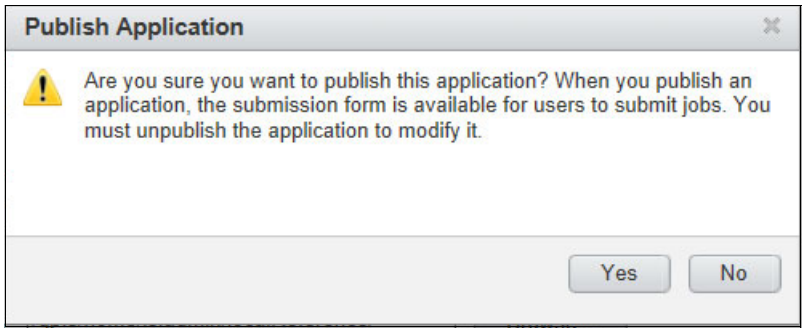


Figure 3-13 Prompt window informing a user about the submission form availability

The published workflow can be submitted by clicking the Jobs tab and selecting **Submission Forms** as shown in Figure 3-14.

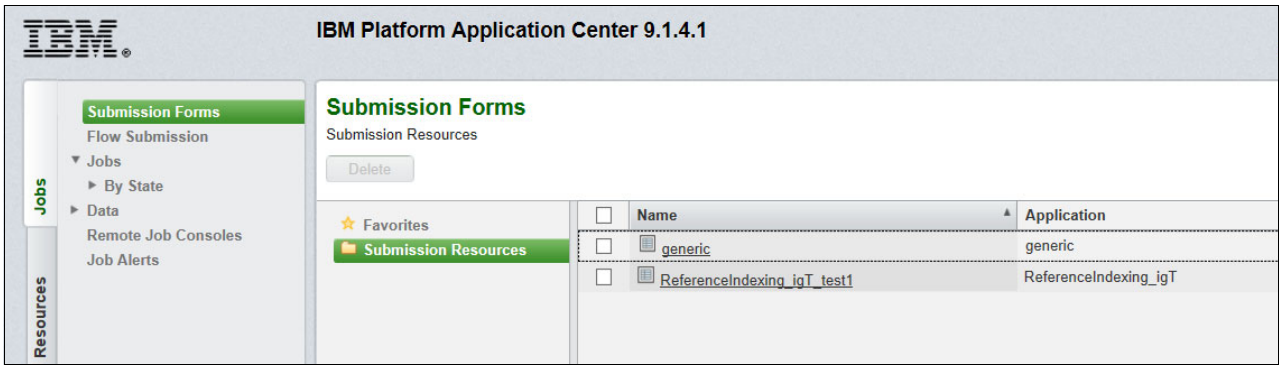


Figure 3-14 Published templates in Submission Forms

Figure 3-15 shows the form that is accessible to the users. The users complete the required values on the submission form and submit the job through PAC.

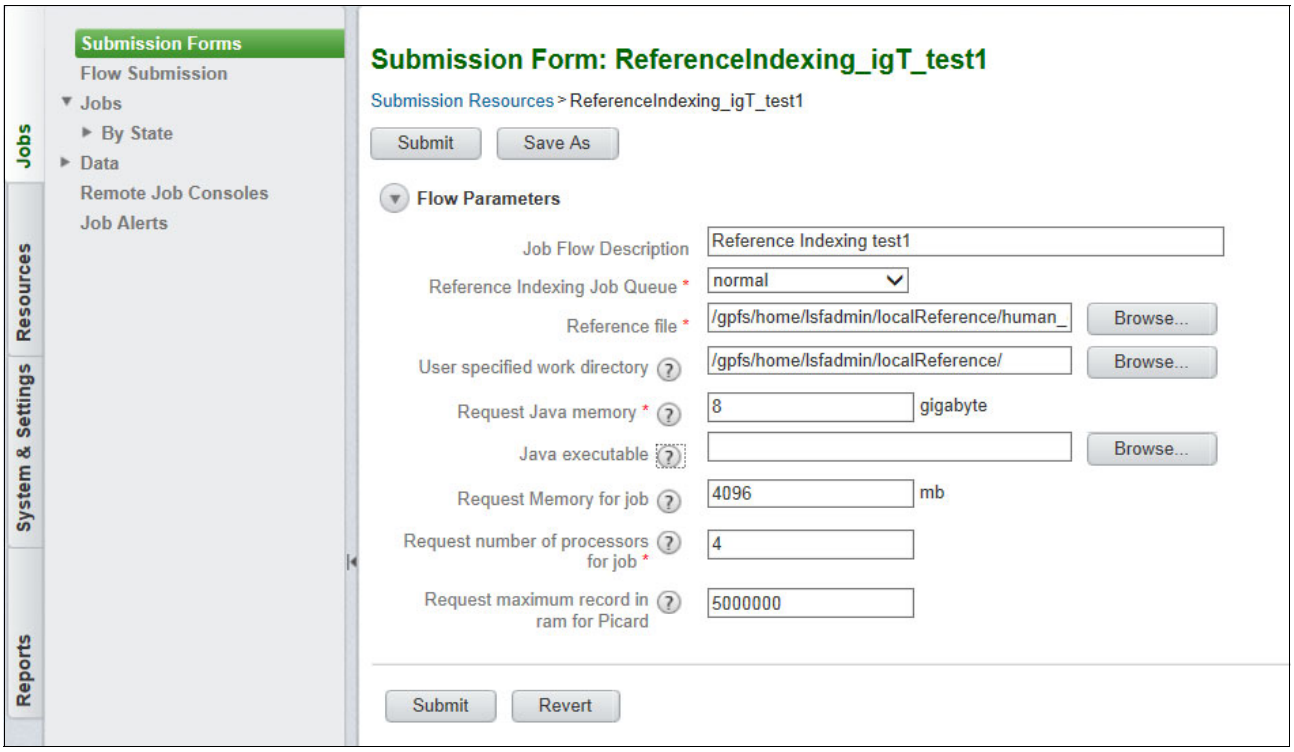


Figure 3-15 Verifying the published template in Submission Forms

To submit the reference Indexing workflow, click **Submit**. The chart view opens (Figure 3-16).

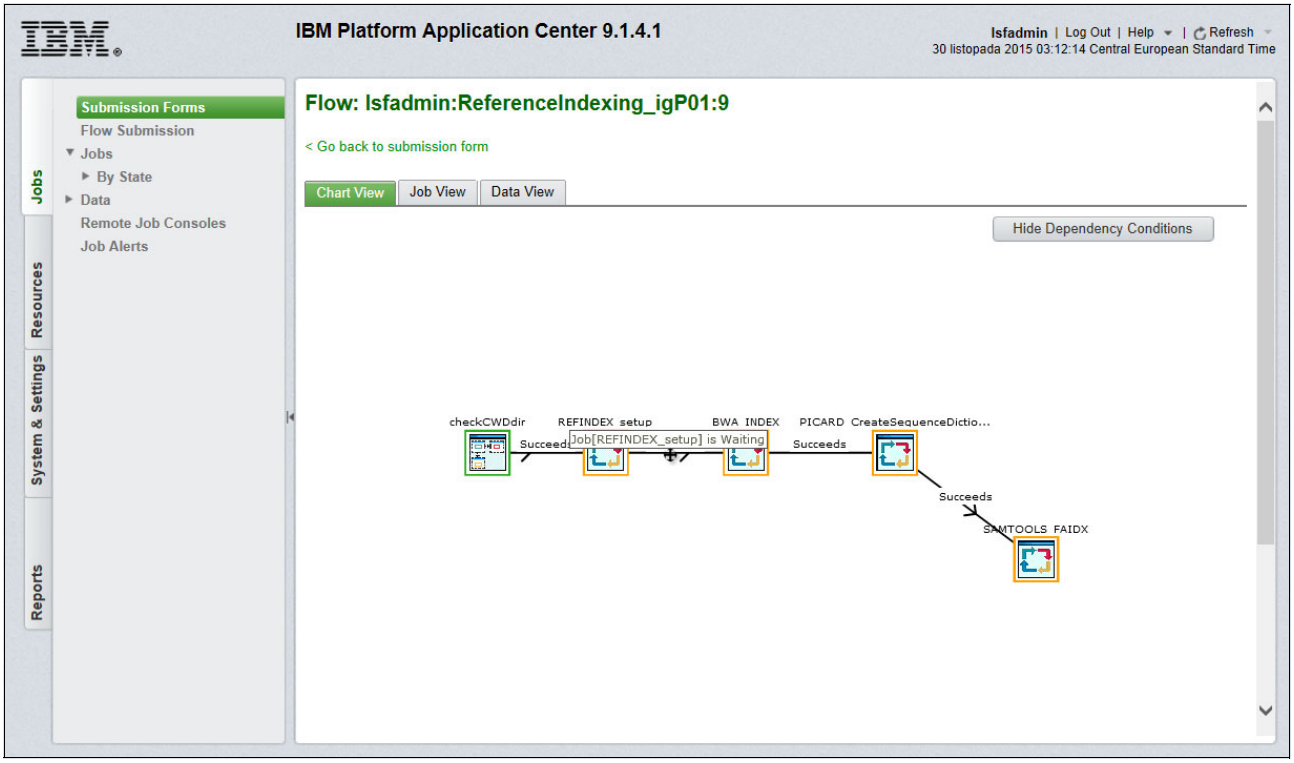


Figure 3-16 Flow: lsfadmin:ReferenceIndexing_igP01 of the submitted job

To display the running jobs, click the Jobs tab and select **Jobs**. From the view shown in Figure 3-17, you can read the Flow ID and the Status, and information about what time the job started.

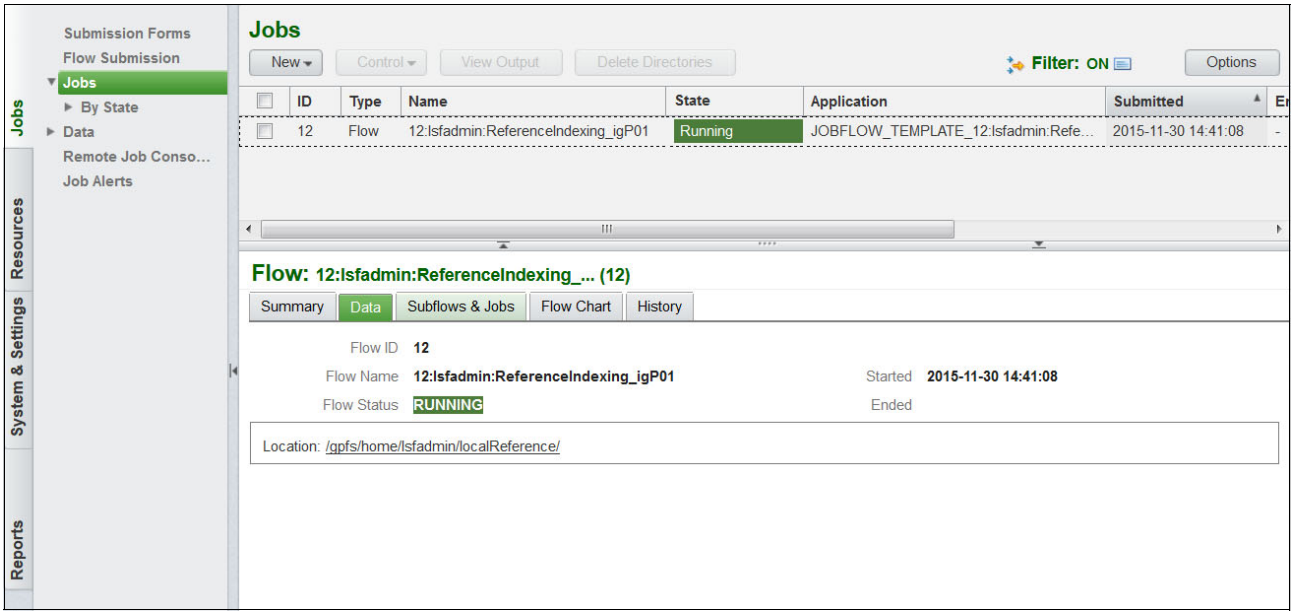


Figure 3-17 Displaying the running jobs

When you move to the tab Subflows and Jobs, you see detailed information about jobs and subflows with the state and Job ID (Figure 3-18).

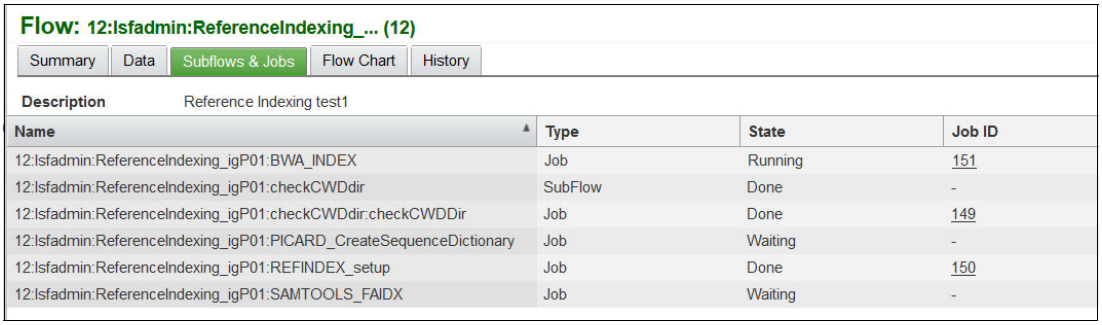


Figure 3-18 Subflows and jobs view

To get more information about a specific job, click the **Job ID**. The window is similar to the one as shown in Figure 3-19. In Job Details section, you see the content of the `/gpfs/home/lsfadmin/localReference/` directory. You can sort the files by file type, file size, and date modified.

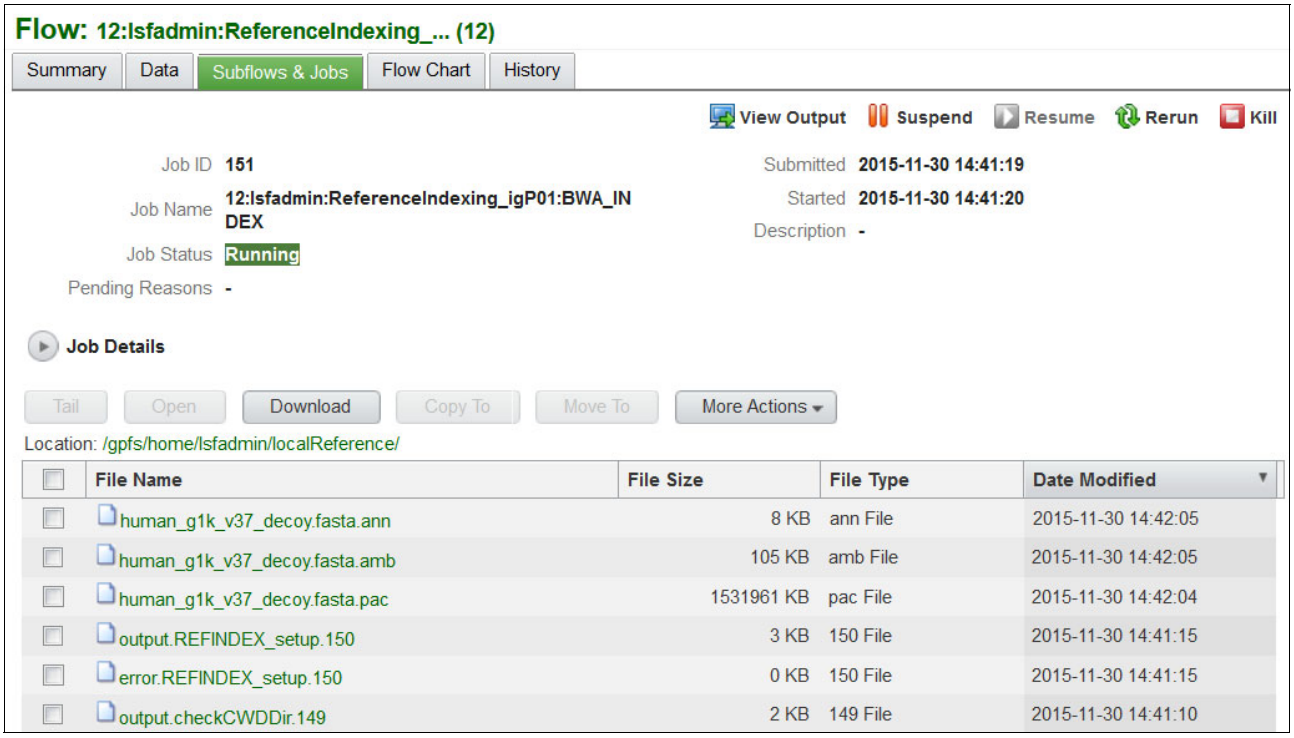


Figure 3-19 Display the specific job details along with files and duration time

You can run commands such as **View output**, **Suspend**, **Resume**, **Rerun**, and **Kill**. Figure 3-20 shows the View Job Output pane, which provides information about the running job. This pane has a useful auto-refresh function.

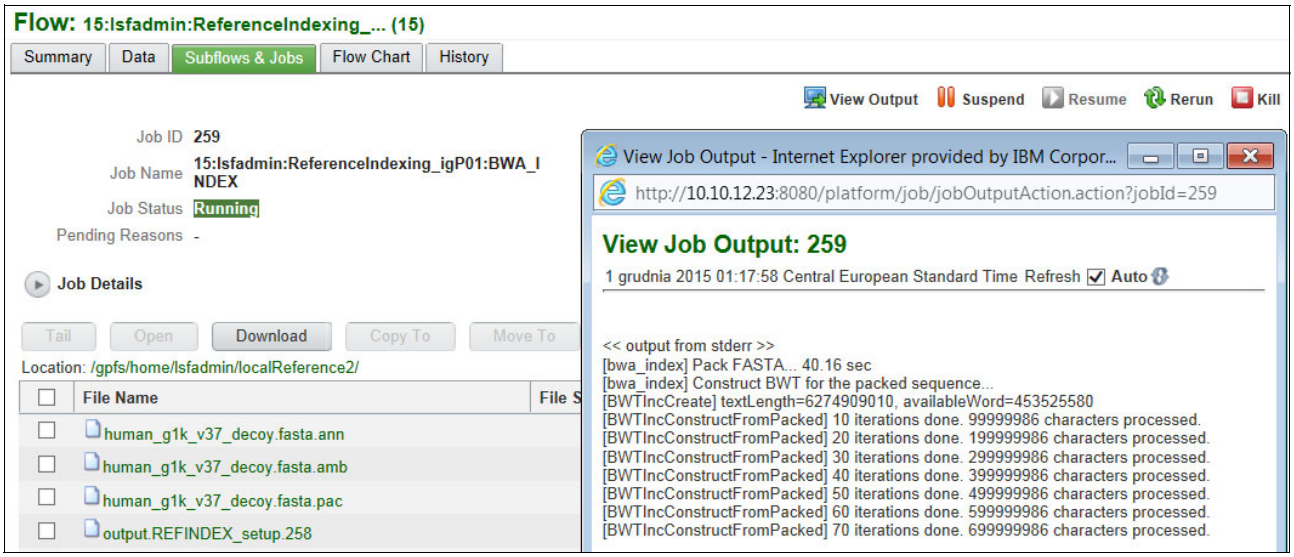


Figure 3-20 View job output window with auto-refreshed content

You can access the output written through PAC, but also in the working directory directly. For the BWA job, the output is stored in the `error.RefIndex_BWA_INDEX.[JOBID]` file. For Picard jobs, the output is written to the `error.RefIndex_CreateSeqDict.[JOBID]` file as shown in Example 3-30.

Example 3-30 Exemplary output of Picard processing job

```
root@cn02:/gpfs/home/lsfadmin/localReference# cat error.RefIndex_CreateSeqDict.260

[Tue Dec 01 04:05:50 EET 2015] picard.sam.CreateSequenceDictionary

REFERENCE=/gpfs/home/lsfadmin/localReference/human_g1k_v37_decoy.fasta
OUTPUT=human_g1k_v37_decoy.fasta.dict
TMP_DIR=[/gpfs/home/lsfadmin/localReference/tmp] MAX_RECORDS_IN_RAM=5000000
TRUNCATE_NAMES_AT_WHITESPACE=true

NUM_SEQUENCES=2147483647 VERBOSITY=INFO QUIET=false VALIDATION_STRINGENCY=STRICT
COMPRESSION_LEVEL=5 CREATE_INDEX=false

CREATE_MD5_FILE=false
[Tue Dec 01 04:05:50 EET 2015] Executing as lsfadmin@cn01
```

Example 3-31 shows the new files created from by processing `human_g1k_v37_decoy.fasta`.

Example 3-31 Files created after the job completion

```
-rw-r--r--  1 lsfadmin lsfadmin 3189750467 Nov 30 23:18 human_g1k_v37_decoy.fasta
-rw-rw-r--  1 lsfadmin lsfadmin     106669 Dec  1 03:38 human_g1k_v37_decoy.fasta.amb
-rw-rw-r--  1 lsfadmin lsfadmin       6924 Dec  1 03:38 human_g1k_v37_decoy.fasta.ann
-rw-rw-r--  1 lsfadmin lsfadmin 3137454592 Dec  1 03:37 human_g1k_v37_decoy.fasta.bwt
-rw-rw-r--  1 lsfadmin lsfadmin      11362 Dec  1 04:06 human_g1k_v37_decoy.fasta.dict
-rw-rw-r--  1 lsfadmin lsfadmin       2813 Dec  1 04:07 human_g1k_v37_decoy.fasta.fai
-rw-rw-r--  1 lsfadmin lsfadmin  784363628 Dec  1 03:38 human_g1k_v37_decoy.fasta.pac
-rw-rw-r--  1 lsfadmin lsfadmin 1568727304 Dec  1 04:05 human_g1k_v37_decoy.fasta.sa
```

3.5.4 Visualization of data

Data can be displayed with the Integrative Genomics Viewer (IGV). The IGV desktop supports browsing of reads from APIs and also from BAM files, and it allows interactive exploration of large data sets.

The IGV installation package can be downloaded from the following website:

http://www.broadinstitute.org/software/igv/download

You can try IGV directly from the following website:

http://igv.org/web/demo/ga4gh-demo.html

An embedded version is also available at the following website:

http://igv.org/web/examples/ga4gh.html

Chapter 4, “Medicine of the future with IBM” on page 111 presents the IBM analytics software that can be used for further processing of the genomic workflow output. For example, IBM analytics software solutions described in 4.2, “Healthcare areas” on page 112 and 4.2.1, “IBM Watson” on page 115 allow you to use the VCF data to process, convert, and visualize the data.

3.6 Storage options using IBM Spectrum Scale

IBM Spectrum Scale offers various options for the file system that stores the genomics data. The base reference architecture uses Elastic Storage Server (ESS) as base storage.

Note: To read more about the IBM Elastic Storage Server, refer to the following website:

http://www.ibm.com/systems/storage/spectrum/ess/

Also refer to the *Introduction Guide to the IBM Elastic Storage Server*, REDP-5253 available at the following website:

http://www.redbooks.ibm.com/abstracts/redp5253.html?Open

Within the same common name space, using IBM Spectrum Scale, it is trivial to have multiple tiers. You might have a smaller portion of the capacity that is available on the file system on high performance storage such as IBM Flash V9000, another portion might be composed of slower SATA drives, and the rest might be on tape storage, either using IBM Linear Tape File System™ (LTFS) or hierarchical storage management (HSM). You can then use the integrated Spectrum Scale information lifecycle management (ILM) tools to move the data to where it can give the most value at the correct cost in a way that is invisible to the file system user as shown in Figure 3-21.

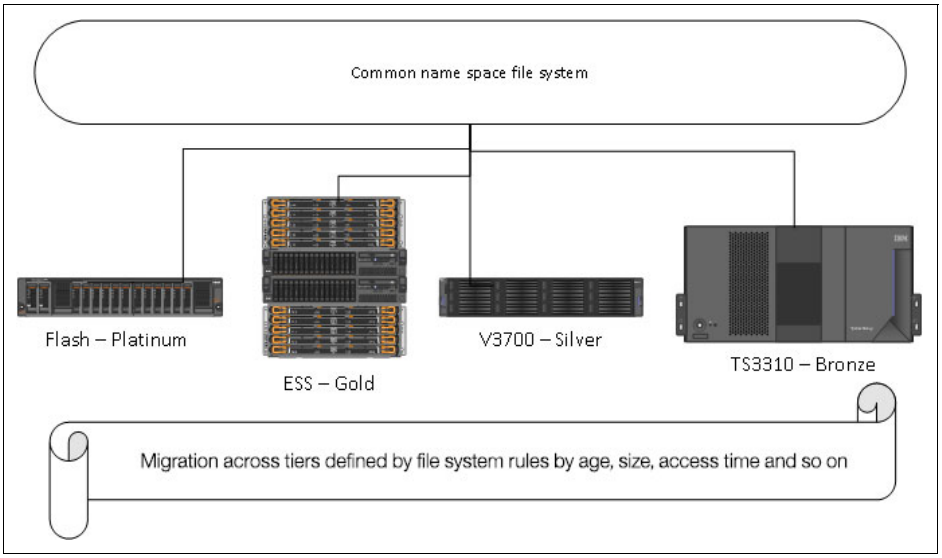


Figure 3-21 IBM Spectrum Scale multitier example

For an overview of Spectrum Scale tiers and ILM, see Chapter 5 Information Life Management in *IBM Spectrum Scale (formerly GPFS)*, SG24-8254 publication at the following website:

http://www.redbooks.ibm.com/abstracts/sg248254.html?Open

IBM Spectrum Scale product offers a native Hadoop connector. For more information about IBM Spectrum Scale native Hadoop connector, see the following website:

http://ibm.co/23E6FAM

By using this connector, the information stored on this solution can be directly used without the need of copying any data because it is stored on a Hadoop Distributed File System

(HDFS). The analytical work by Hadoop or any other computational cluster that uses HDFS, like Apache Spark, can be also used.

Note: For more information about Apache Spark and IBM, see the following website:

http://www.ibm.com/analytics/us/en/technology/spark/

3.7 Adding compute nodes to a running cluster

During the lifetime of your cluster, additional tasks will be required such as:

- Temporary increase of local compute power
- Temporary increase of remote compute power
- Permanent increase of remote compute power
- Permanent increase of local compute power
- Refresh compute nodes to a newer generation, both locally and remotely

The reference architecture explained in Chapter 2, “Reference architecture” on page 7 provides the needed tools to support the process off adding local compute nodes. Use the IBM Platform Cluster Manager interface to deploy the nodes from the base operating system installation until the node is ready to accept compute jobs. All of these nodes can be managed from the Platform Cluster Manager interface.

The steps to deploy compute nodes for IBM Life Sciences solution is explained in detail in the *IBM Data Engine for Genomics - Power Systems Edition Architecture and Reference Design*, which is available at:

http://ibm.co/1i1NPzi

The steps to add new local compute nodes do not differ from the steps to add the initial compute nodes to the IBM Life Science solution.

In some cases, the tools that are provided by the Life Science solution cannot be used. However, the architecture and the solution allows the use of nodes added without Platform Cluster Manager, which means all the steps must be done manually. This process includes installing the components needed to manage and start jobs.

The following scenarios show how to manually add compute nodes to the cluster. These configurations involve the following assumptions:

- Nodes are local or remote
- Nodes are POWER8 servers
- IBM Spectrum Scale is directly accessible, and nodes can mount the file system
- The operating system is installed by a cloud provider, IT department, or similar third party

Note: IBM Spectrum Scale supports various configurations. Some include local cache for remote sites where the hot portion of the file system is local. The configuration beyond the simplest all local nodes and no routing at networks falls beyond the scope of this book. For more information about the options that are offered by the product, see *IBM Spectrum Scale (formerly GPFS)*, SG24-8254, at:

http://www.redbooks.ibm.com/abstracts/sg248254.html

3.7.1 Adding POWER CPU nodes

This section shows how to manually add to the IBM Life Science cluster a compute node. The node, when added to the cluster, provides CPU and memory resources to the cluster and can be used to run jobs. The node being added here is an IBM Power Systems S822LC for commercial computing. For more information about this Power Systems server, see the following website:

http://www.ibm.com/marketplace/cloud/commercial-computing/us/en-us

The node can be on premises or remote. In this example, the node runs the Ubuntu 14.04.3 operating system on an IBM Power Systems S822LC for commercial computing and has the network already configured for remote access.

Note: For more information about the IBM Power Systems S822LC for commercial computing, see the following link:

http://www.ibm.com/marketplace/cloud/commercial-computing/us/en-us

Additional details can be found in the *IBM Power Systems S822LC Technical Overview and Introduction*, REDP-5283, at the following link:

http://www.redbooks.ibm.com/abstracts/redp5283.html

Also, perform the following steps to install IBM Spectrum Scale:

Note: These steps assume that certain prerequisites on the configuration are already done and not only limited to:

- Network Time Protocol (NTP) is already configured.
- SSH keys passwordless access from ESS nodes is done.
- Hosts can be DNS resolved or added to `/etc/hosts`.

For other prerequisites for IBM Spectrum Scale, see *IBM Spectrum Scale (formerly GPFS)*, SG24-8254, at:

http://www.redbooks.ibm.com/abstracts/sg248254.html

1. Install Spectrum Scale prerequisites on the compute node by using the **`apt-get`** command. The command shown in Example 3-32 applies to Ubuntu base installation only.

   Example 3-32 Install IBM Spectrum Scale prerequisites on Ubuntu

   ```
   # apt-get install ksh binutils m4 gcc g++ make
   ```

2. Use the **`dpkg`** command on the compute node to install Spectrum Scale as shown in Example 3-33.

   Example 3-33 Installing IBM Spectrum Scale on the Ubuntu server

   ```
   # dpkg -i gpfs.*.deb
   (Reading database ... 129912 files and directories currently installed.)
   Preparing to unpack gpfs.base_4.1.1-1_ppc64el_base.deb ...
   Unpacking gpfs.base (4.1.1-1) ...
   Preparing to unpack gpfs.docs_4.1.1-1_all.deb ...
   Unpacking gpfs.docs (4.1.1-1) ...
   Preparing to unpack gpfs.ext_4.1.1-1_ppc64el_base.deb ...
   Unpacking gpfs.ext (4.1.1-1) ...
   ```

```
Preparing to unpack gpfs.gpl_4.1.1-1_all.deb ...
Unpacking gpfs.gpl (4.1.1-1) ...
Preparing to unpack gpfs.gskit_8.0.50-40_ppc64el.deb ...
Unpacking gpfs.gskit (8.0.50-40) ...
Preparing to unpack gpfs.msg.en-us_4.1.1-1_all.deb ...
Unpacking gpfs.msg.en-us (4.1.1-1) ...
Setting up gpfs.base (4.1.1-1) ...
Setting up gpfs.docs (4.1.1-1) ...
Setting up gpfs.ext (4.1.1-1) ...
Setting up gpfs.gpl (4.1.1-1) ...
Setting up gpfs.gskit (8.0.50-40) ...
Setting up gpfs.msg.en-us (4.1.1-1) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Processing triggers for libc-bin (2.19-0ubuntu6.6) ...
```

3. Build the Spectrum Scale portability layer on the compute node as shown in Example 3-34.

Note: This method of building the portability layer requires a compiler to be installed on each node. There are other methods available to build the portability layer that do not need to have the compiler installed on the node.

Example 3-34 Building the Spectrum Scale portability layer on Ubuntu

```
#cd /usr/lpp/mmfs/bin
# ./mmbuildgpl
-------------------------------------------------------
mmbuildgpl: Building GPL module begins at Wed Sep 30 00:42:21 EEST 2015.
-------------------------------------------------------
Verifying Kernel Header...
  kernel version = 31300065 (3.13.0-65-generic, 3.13.0-65)
  module include dir = /lib/modules/3.13.0-65-generic/build/include
  module build dir   = /lib/modules/3.13.0-65-generic/build
  kernel source dir  = /usr/src/linux-3.13.0-65-generic/include
  Found valid kernel header file under
/lib/modules/3.13.0-65-generic/build/include
Verifying Compiler...
  make is present at /usr/bin/make
  cpp is present at /usr/bin/cpp
  gcc is present at /usr/bin/gcc
  g++ is present at /usr/bin/g++
  ld is present at /usr/bin/ld
make World ...
make InstallImages ...
-------------------------------------------------------
mmbuildgpl: Building GPL module completed successfully at Wed Sep 30 00:42:52
EEST 2015.
-------------------------------------------------------
```

4. Add the compute node to the Spectrum Scale cluster. To do so, run the **mmaddnode** command from a node that can administer the Spectrum Scale cluster as shown in Example 3-35.

Example 3-35 Add a node to the Spectrum Scale cluster

```
# mmaddnode -N cn03
Wed Sep 30 00:43:32 EEST 2015: mmaddnode: Processing node cn03
mmaddnode: Command successfully completed
mmaddnode: Warning: Not all nodes have proper GPFS license designations.
   Use the mmchlicense command to designate licenses as needed.
mmaddnode: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
```

5. Set the Spectrum Scale license of the added node as shown in Example 3-35 on page 99. To do so, run the command **mmchlicense** from a node that can administer the Spectrum Scale cluster as shown in Example 3-36.

Example 3-36 Add the Spectrum Scale license of a client node

```
# mmchlicense client --accept -N cn03

The following nodes will be designated as possessing GPFS client licenses:
   cn03
mmchlicense: Command successfully completed
mmchlicense: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
```

6. Start the Spectrum Scale daemon on the client node. To do so, run the **mmstartup** command from a node that can administer the Spectrum Scale cluster as shown in Example 3-37.

Example 3-37 Start Spectrum Scale daemon

```
# mmstartup -N cn03
Wed Sep 30 00:44:06 EEST 2015: mmstartup: Starting GPFS ...
```

Note: This scenario assumes that the Spectrum Scale file system automounts at daemon startup. If that is not the case in your Spectrum Scale cluster, familiarize yourself with the **mmmount** command.

7. Create and set the password for the IBM Platform LSF admin on the compute node. To do so, run the **useradd** command and the **passwd** command on the compute node as shown in Example 3-38.

Example 3-38 Create and set the password of LSF admin user on the compute node

```
# useradd lsfadmin
# passwd lsfadmin
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Note: This example are uses `lsfadmin` as the password for the `lsfadmin` user.

8. Add the node to the LSF cluster file. To do so, from the LSF master node edit the LSF cluster config file `lsf.cluster` that corresponds to your cluster and add a line that defines the new compute node. Example 3-39 shows this process being done in node `Brazil`.

Example 3-39 Adding cn03 compute node to the LSF cluster file

```
# export LSF_CONF="/gpfs/lsf/conf"
# . $LSF_CONF/profile.lsf
Edit $LSF_CONF/lsf.cluster.Brazil to add a line like on the host definition
area

cn03   !   !   1   3.5   ()   ()   (linux)
```

Note: In Example 3-39 on page 100 and Example 3-40 the example cluster is called `Brazil`. The compute node is called `cn03`, the LSF master node is `mn01`, and the LSF top directory is `/gpfs/lsf`.

9. Install LSF on the compute node. To do so, edit and run the **`rhostsetup`** command to include the computer node to the deployment target as shown in Example 3-40.

Example 3-40 Installing LSF client from LSF master

```
# export LSF_TOP="/gpfs/lsf"
# . $LSF_TOP/conf/profile.lsf
Edit the file $LSF_TOP/9.1/rhostsetup to include the compute node cn03 on the
LSF_HOST variable
# cd $LSF_TOP/9.1/
# ./rhostsetup

Setup host list: cn03
Setup command: /gpfs/lsf/9.1/install/hostsetup --top="/gpfs/lsf" --boot="y"

Do you want to continue (y/n)?
y
Logging installation sequence in /gpfs/lsf/log/Install.log

------------------------------------------------------------
   L S F    H O S T S E T U P    U T I L I T Y
------------------------------------------------------------
This script sets up local host (LSF server, client or slave) environment.

Setting up LSF server host "cn03" ...
Checking LSF installation for host "cn03" ... Done
Copying /etc/init.d/lsf, /etc/rc2.d/S95lsf and /etc/rc1.d/K05lsf
Installing LSF RC scripts on host "cn03" ... Done
LSF service ports are defined in /gpfs/lsf/conf/lsf.conf.
Checking LSF service ports definition on host "cn03" ... Done
You are installing IBM Platform LSF -  Standard Edition.

... Setting up LSF server host "cn03" is done
... LSF host setup is done.
```

10. Manually start the LSF daemons on the compute node. To do so, run the **lsf_daemons** command as root on the compute node as shown in Example 3-41.

Example 3-41 Manually start LSF client daemons

```
# export LSF_CONF="/gpfs/lsf/conf"
# . $LSF_CONF/profile.lsf
# lsf_daemons start
Starting the LSF subsystem

# lsf_daemons status
Show status of the LSF subsystem
lim (pid 15970) is running...
res (pid 15972) is running...
sbatchd (pid 15974) is running...
```

11. Restart the LSF suite on the LSF master. To do so, run the **lsfrestart** command on the LSF master as shown in Example 3-42.

Example 3-42 Restart LSF suite on master LSF server

```
# export LSF_TOP="/gpfs/lsf"
# . $LSF_TOP/conf/profile.lsf
# lsfrestart
Restarting all LIMs ...

Checking configuration files ...
No errors found.

Restart only the master candidate hosts? [y/n] n
Do you really want to restart LIMs on all hosts? [y/n] y
Restart LIM on <mn01> ...... done
Restart LIM on <cn01> ...... done
Restart LIM on <cn02> ...... done
Restart LIM on <cn03> ...... done

Waiting for Master LIM to start up .....  Master LIM is ok
Restarting all RESes ...
Restart RES on <mn01> ...... done
Restart RES on <cn01> ...... done
Restart RES on <cn02> ...... done
Restart RES on <cn03> ...... done

Restarting all slave daemons on LSBATCH hosts ...
LSF is down. Please wait ...
LSF is down. Please wait ...
LSF is down. Please wait ...
Restart slave batch daemon on <cn01> ...... done
Restart slave batch daemon on <cn02> ...... done
Restart slave batch daemon on <cn03> ...... done
Restart slave batch daemon on <mn01> ...... done

Restarting master daemon of LSBATCH ...

Checking configuration files ...
```

```
No errors found.

MBD restart initiated
```

Note: At this point, the compute node is added to the solution cluster. It is visible from the PAC interface in ready status and is ready to run jobs.

12. Check the status of the new compute node. To do so, run the **bhosts** command from the LSF master node as shown in Example 3-43.

Example 3-43 List LSF nodes from LSF master

```
# export LSF_TOP="/gpfs/lsf"
# . $LSF_TOP/conf/profile.lsf
# bhosts
HOST_NAME          STATUS       JL/U    MAX  NJOBS    RUN  SSUSP  USUSP    RSV
cn01               ok              -      128      0      0      0      0
0
cn02               ok              -      128      0      0      0      0
0
cn03               ok              -      128      0      0      0      0
0
mn01               ok              -      128      0      0      0      0
0
```

As the example will run the GATK, the compute nodes need to have Java runtime installed as shown in Example 3-44.

Example 3-44 Installing Java runtime on Ubuntu

```
# apt-get install openjdk-7-jre-headless
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  ca-certificates-java
Suggested packages:
  icedtea-7-jre-jamvm libnss-mdns sun-java6-fonts fonts-dejavu-extra
  fonts-ipafont-gothic fonts-ipafont-mincho ttf-wqy-microhei ttf-wqy-zenhei
  ttf-indic-fonts-core ttf-telugu-fonts ttf-oriya-fonts ttf-kannada-fonts
  ttf-bengali-fonts
The following NEW packages will be installed:
  ca-certificates-java openjdk-7-jre-headless
0 upgraded, 2 newly installed, 0 to remove and 15 not upgraded.
Need to get 0 B/39.5 MB of archives.
After this operation, 57.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Selecting previously unselected package ca-certificates-java.
(Reading database ... 105002 files and directories currently installed.)
Preparing to unpack .../ca-certificates-java_20130815ubuntu1_all.deb ...
Unpacking ca-certificates-java (20130815ubuntu1) ...
Selecting previously unselected package openjdk-7-jre-headless:ppcle.
Preparing to unpack
.../openjdk-7-jre-headless_7u79-2.5.6-0ubuntu1.14.04.1_ppcle.deb ...
Unpacking openjdk-7-jre-headless:ppcle (7u79-2.5.6-0ubuntu1.14.04.1) ...
```

```
Processing triggers for ca-certificates (20141019ubuntu0.14.04.1) ...
Updating certificates in /etc/ssl/certs... 0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d....
done.
done.
Setting up ca-certificates-java (20130815ubuntu1) ...
Setting up openjdk-7-jre-headless:ppcle (7u79-2.5.6-0ubuntu1.14.04.1) ...
update-alternatives: using /usr/lib/jvm/java-7-openjdk-ppcle/jre/bin/java to
provide /usr/bin/java (java) in auto mode
update-alternatives: using /usr/lib/jvm/java-7-openjdk-ppcle/jre/bin/keytool to
provide /usr/bin/keytool (keytool) in auto mode
update-alternatives: using /usr/lib/jvm/java-7-openjdk-ppcle/jre/bin/pack200 to
provide /usr/bin/pack200 (pack200) in auto mode
update-alternatives: using /usr/lib/jvm/java-7-openjdk-ppcle/jre/bin/rmid to
provide /usr/bin/rmid (rmid) in auto mode
update-alternatives: using /usr/lib/jvm/java-7-openjdk-ppcle/jre/bin/rmiregistry
to provide /usr/bin/rmiregistry (rmiregistry) in auto mode
update-alternatives: using /usr/lib/jvm/java-7-openjdk-ppcle/jre/bin/unpack200 to
provide /usr/bin/unpack200 (unpack200) in auto mode
update-alternatives: using /usr/lib/jvm/java-7-openjdk-ppcle/jre/bin/orbd to
provide /usr/bin/orbd (orbd) in auto mode
update-alternatives: using /usr/lib/jvm/java-7-openjdk-ppcle/jre/bin/servertool to
provide /usr/bin/servertool (servertool) in auto mode
update-alternatives: using /usr/lib/jvm/java-7-openjdk-ppcle/jre/bin/tnameserv to
provide /usr/bin/tnameserv (tnameserv) in auto mode
update-alternatives: using /usr/lib/jvm/java-7-openjdk-ppcle/jre/lib/jexec to
provide /usr/bin/jexec (jexec) in auto mode
Processing triggers for ca-certificates (20141019ubuntu0.14.04.1) ...
Updating certificates in /etc/ssl/certs... 0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d....
done.
done.
Processing triggers for libc-bin (2.19-0ubuntu6.6) ...
```

Note: Java runtime is only installed because it is a requirement for GATK. If the workload that you need to run on the CPU nodes requires another runtime such as Python, install that runtime before trying to run any jobs on the new nodes.

3.7.2 Adding POWER GPU nodes

Not all the implementation of genomics algorithms can take advantage of GPU processing. The following are some of the implementations that take advantage of GPU in the cluster:

- BarraCUDA: Burrows-Wheeler Aligner GPU enabled implementation. Find more information at the following website:

  http://bio-bwa.sourceforge.net

- CUDA-EC: Short-read alignment error correction using GPU. Find more information at the following website:

  http://www.nvidia.com/object/bio_info_life_sciences.html

- CUDA-MEME: Motif discovery software based on the MEME (version 3.5.4) algorithm for a single GPU. Find more information at the following website:

  https://sites.google.com/site/yongchaosoftware/Home/cuda-meme

- CUSHAW2-GPU: Fast, parallel and accurate gapped read alignment. Find more information at the following website:

  http://cushaw2.sourceforge.net

- CUDASW++: Bioinformatics software for Smith-Waterman. Find more information at the following website:

  http://cudasw.sourceforge.net

- mCUDA-MEME: Ultrafast scalable motif discovery algorithm based on the MEME algorithm for multiple GPU. Find more information at the following website:

  https://sites.google.com/site/yongchaosoftware/mcuda-meme

- NextGenMap: Burrows-Wheeler Aligner GPU enabled implementation. Find more information at the following website:

  https://github.com/Cibiv/NextGenMap/wiki

- SOAP3DP: Fast, Accurate and Sensitive GPU-based Short Read Aligner. Find more information at the following website:

  http://sourceforge.net/projects/soap3dp/

Note: The intention of this list is to show that only a portion of today's implementations that can take advantage of GPU processing. Depending on your workload, it might not be cost effective or even efficient to have GPU in the cluster. Perform an assessment of your workload and what GPU can add to the efficiency of your setup. In any case, by using an IBM POWER8 system you will take advantage of fast I/O speeds between the PCIe and the memory that you cannot if you use other systems.

This scenario adds an IBM P822LC POWER8 server running Red Hat Enterprise Linux (RHEL), 7.2 Little Endian (LE), and Compute Unified Device Architecture (CUDA) 7.5. A SOAP3DP job was run from both the CLI and GUI.

Note: Find more information about the IBM P822LC HPC system at the following website:

http://www.ibm.com/marketplace/cloud/high-performance-computing/us/en-us

More details are available in *IBM Power Systems S822LC Technical Overview and Introduction*, REDP-5283 at the following website:

http://www.redbooks.ibm.com/abstracts/redp5283.html?Open

For more information about obtaining RHEL 7.2 LE, see the following website:

http://www.redhat.com/en/technologies/linux-platforms/enterprise-linux

For more information about how to download CUDA 7.5, see the following website:

https://developer.nvidia.com/cuda-downloads

This example assumes the equivalent RHEL steps that are defined in 3.7.1, "Adding POWER CPU nodes" on page 97 are already completed. This process includes these steps:

- Install IBM Spectrum Scale prerequisites
- Install IBM Spectrum Scale
- Add node to LSF cluster

Install CUDA 7.5 by following the instructions at this link:

http://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#redhat-installation

To install CUDA as shown in Example 3-45, use the local repo file obtained from the NVIDIA CUDA website in the note below.

Example 3-45 Installing CUDA 7.5 on RHEL 7.2 LE

```
# rpm -Uvh
http://download.fedoraproject.org/pub/epel/7/ppc64/e/epel-rpm-macros-7-3.noarch.rp
m
# rpm -Uvh
http://download.fedoraproject.org/pub/epel/7/ppc64/d/dkms-2.2.0.3-30.git.7c3e7c5.e
l7.noarch.rpm
# rpm -ivh cuda-repo-rhel7-7-5-local-7.5-18.ppc64le.rpm
```

Note: At the time of writing this publication, the following symbolic link needed to be added to make CUDA work:

`ln -sf /usr/local/cuda-7.5/targets/ppc64le-linux/lib/libcudart.so.7.5 /lib64/libcudart.so.7.5`

To run the job in the CLI, create the `SOAP3-DP_example.sh` script shown in Example 3-46 which can be run from the local node where the GPU is available.

Example 3-46 Script to manually run SOAP3-DP

```
#!/bin/bash
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/lib64
echo "soap3-dp"
/gpfs/gpfs0001/SOAP3-dp-master/soap3-dp pair NC_012967.1.fasta.index
SRR030257_1.fastq SRR030257_2.fastq -u 500 -v 200
```

Run the *SOAP3-DP_example.sh* script locally in the node with GPU as shown in Example 3-47.

Example 3-47 Run of SOAP3-DP_example.sh

```
root@firestone01:/gpfs/gpfs0001/Ecoli# ./SOAP3-DP_example.sh

[Main] SOAP3-DP v2.3.178 (build)
[Main] Loading read files SRR030257_1.fastq and SRR030257_2.fastq
[Main] loading index into host...
[Main] Finished loading index into host.
[Main] Loading time :    0.1008 seconds

[Main] Finished loading index into host.
[Main] Loading time :    0.1009 seconds
[Main] Reference sequence length : 4629812

[Main] Loaded 7600360 short reads from the query file.
[Main] Elapsed time on host :    8.2708 seconds

[Main] Finished copying index into device (GPU).
[Main] Loading time :    0.0463 seconds

[Main] Finished alignment with <= 1 mismatches
[Main] Number of pairs aligned: 6488
[Main] Elapsed time :    5.3406 seconds
```

```
[Main] 14 half-aligned pairs of reads are proceeded to new default DP.
[Main] Number of pairs aligned by new default DP: 2
[Main] Elapsed time :    0.0565 seconds
[Main] Total Number of pairs aligned: 6490

[Main] 3541599 half-aligned pairs of reads are proceeded to DP.
[Main] Number of pairs aligned by DP: 81082
[Main] Elapsed time :   15.8532 seconds
[Main] Total Number of pairs aligned: 87572

[Main] Finished copying index into device (GPU).
[Main] Loading time :    0.0016 seconds

[Main] 3712608 pairs of reads are proceeded to deep DP.
[Main] Number of pairs aligned by DP: 37034
[Main] Elapsed time :   16.4501 seconds
[Main] Total Number of pairs aligned: 124606

[Main] 2311242 unaligned reads are proceeded to DP.
[Main] Finished copying index into device (GPU).
[Main] Loading time :    0.0017 seconds

[Main] Number of reads aligned by single-end DP: 2175986
[Main] Elapsed time :   20.7420 seconds

[Main] Overall number of pairs of reads aligned: 124606
[Main] Overall read load time :    8.2708 seconds
[Main] Overall alignment time (excl. read loading) :   58.4424 seconds
[Main] Free index from host memory..
[Main] Free host memory..
```

While the job is running, check the status of the GPU in the system by running the **nvidia-smi** command as shown in Example 3-48.

Example 3-48 Output of nvidia-smi command while running a job on the GPU

```
# nvidia-smi

Thu Oct  8 11:10:41 2015
+------------------------------------------------------+
| NVIDIA-SMI 352.39     Driver Version: 352.39         |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla K80           Off  | 0000:03:00.0     Off |                    0 |
| N/A   44C    P0   114W / 149W |   1729MiB / 11519MiB |     80%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla K80           Off  | 0000:04:00.0     Off |                    0 |
| N/A   34C    P8    32W / 149W |     22MiB / 11519MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla K80           Off  | 0002:03:00.0     Off |                    0 |
| N/A   37C    P8    26W / 149W |     22MiB / 11519MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla K80           Off  | 0002:04:00.0     Off |                    0 |
| N/A   35C    P8    30W / 149W |     22MiB / 11519MiB |      0%      Default |
```

```
+-----------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0     32751    C   /gpfs/gpfs001/SOAP3-dp-master/soap3-dp  1705MiB       |
+-----------------------------------------------------------------------------+
```

Defined a flow in the IBM PPM that is run by using the PAC GUI as shown in Figure 3-22.

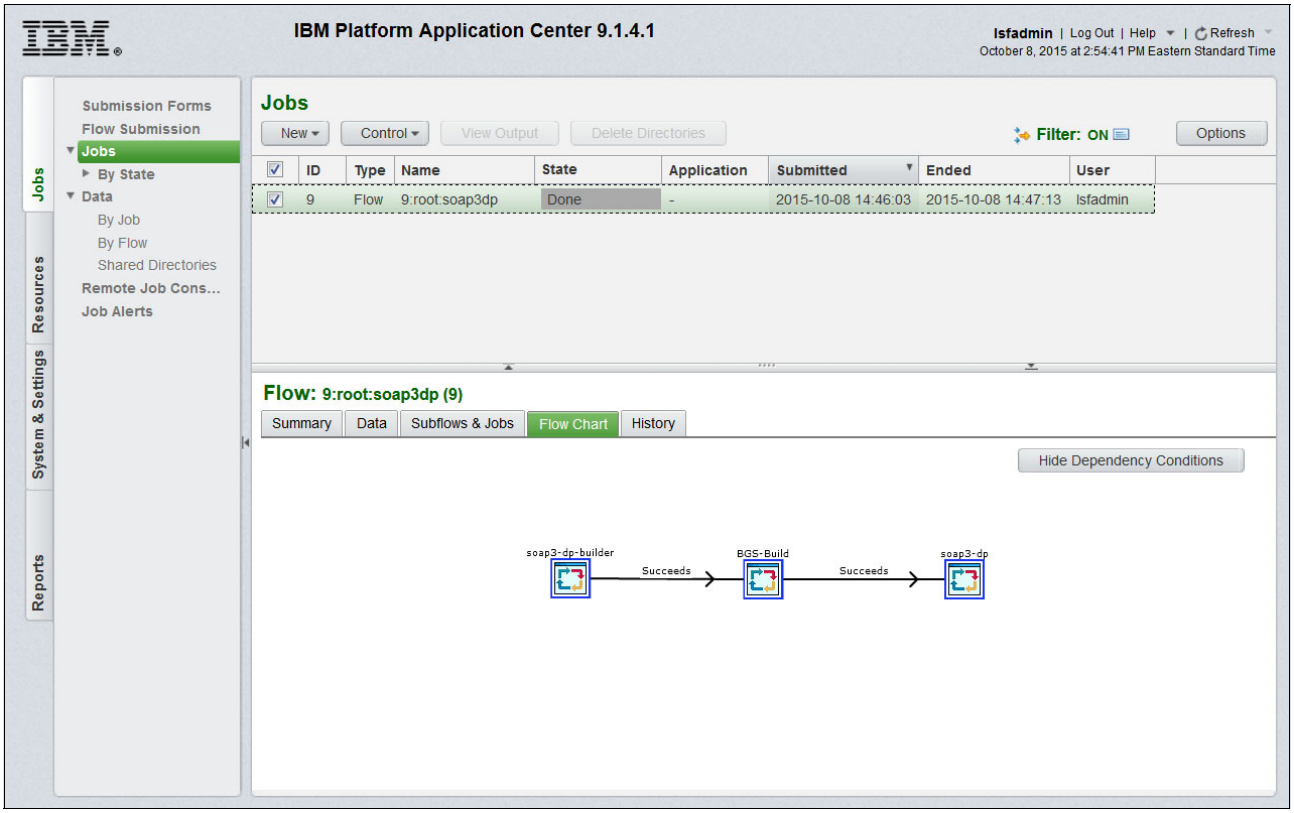


Figure 3-22 SOAP3-DP flow view

Figure 3-23 shows the SOAP3-DP job run.

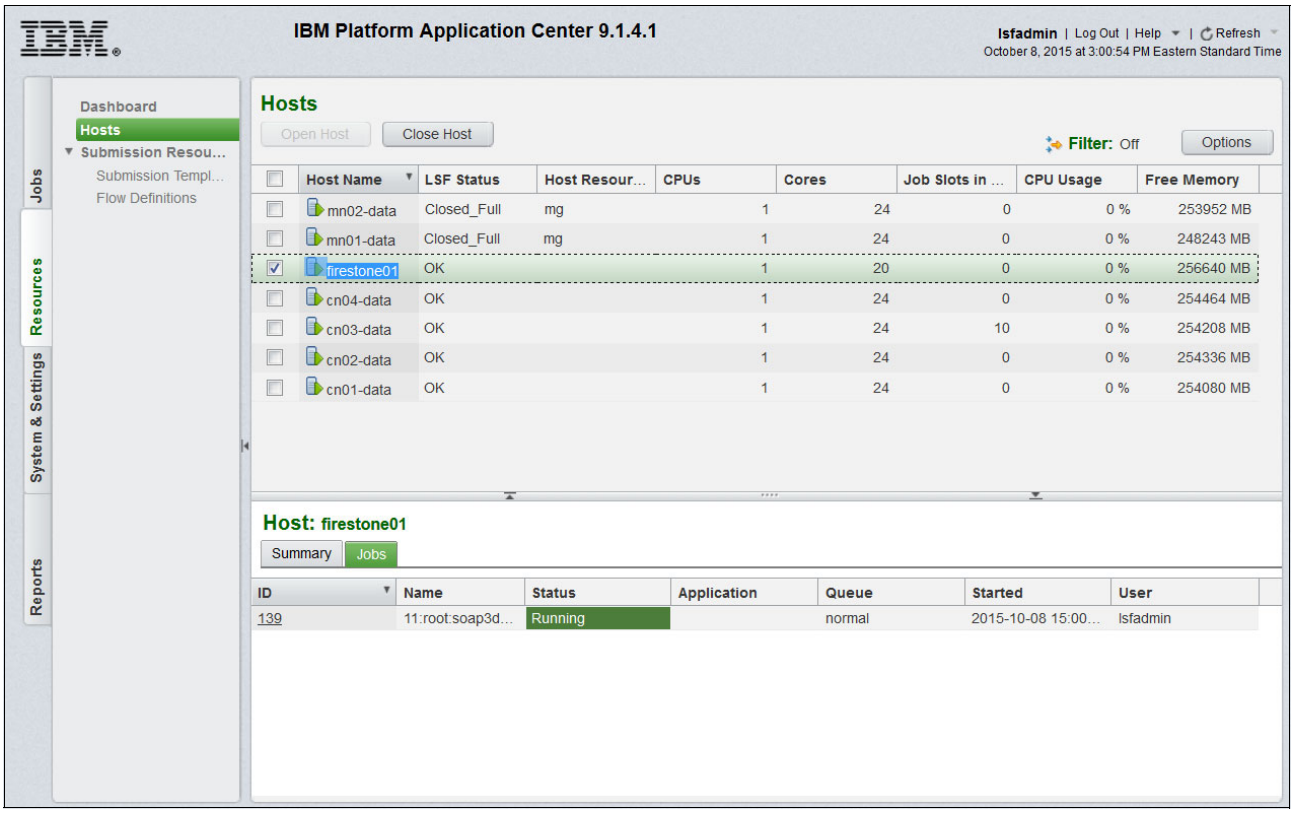


Figure 3-23 SOAP3-DP job run

3.8 Additional information

See the following Platform Computing product documentation for more information:

- PPM

  http://www.ibm.com/support/knowledgecenter/SSKTK2/welcome

- PAC

  http://www.ibm.com/support/knowledgecenter/SSGSCT/product_welcome_pac.html

- RTM

  http://www.ibm.com/support/knowledgecenter/SSVMSD/welcome

3.9 Other vendors packages for IBM Power Systems

This section provides information about third party software that can be used for the IBM Power Systems cluster implementation.

3.9.1 Databiology

For information about databiology, see the IBM Paper Databiology and IBM Platform Computing power life sciences discovery, DCS03065USEN:

http://ibm.co/23Ek5wt

More information regarding databiology can be found at the following websites:

http://www.databiology.com/
http://www.databiology.com/index.php/product#architecture

This section provides describes some of the features of the software:

- Absolute data provenance and reproducibility

  Pervasive automatic data tracking, which enables reproducible analysis workloads, debugging of analysis pipelines, and absolute data provenance.

- Orchestration of software and data together dynamically

  Applications (for example, genome browsers, data presentations, and mining tools) are associated with data and storage types and are launched directly from DBE.

- Integration by seamless collaboration with external systems

  Information and services are easily and securely requested by using the API by users or systems with appropriate access rights. This flexible and technology-agnostic approach provides many possibilities for further integration of the DBE platform with other core systems (for example, clinical management systems and electronic lab notebooks).

DBE uses the Instrument to Insight process model, which makes it applicable across life sciences and healthcare. Here are some use case examples across different industry segments:

- Enterprise-wide management of R&D omics processes (Pharma, Crop Sciences, Genomics Institutes).
- Specific I2I – run best in class tools, infrastructure (Process enablement for CROs, LABs, Healthtech).
- Add flexibility to your infrastructure, cloud burst-out, and so on. (Pharma, Crop, Biotech, CRO).
- Data organization – IM based on project and context (Public Projects, Scientific Publishers, Content Vendors).
- Omics application enabled (Health-techs, Start-ups, Diagnostics).

4

Medicine of the future with IBM

This chapter takes a closer look at the current actions of IBM in the healthcare industry. It answers these questions:

- When will genome sequencing with deep analysis become a standard service at the hospital, and how will it be included in a patient health record?
- How supportive does software have to be to move healthcare to a new level?
- What direction will personalized medicine develop toward?

While you cannot predict the future with 100% accuracy, it is important to anticipate your future needs so you can work with IBM to tailor portfolio to meet those needs.

This chapter is organized around three strategic areas in medicine, focusing mostly on translational and personalized patient care.

This chapter includes the following sections:

- Introduction
- Healthcare areas
- Read and follow

4.1 Introduction

Due to advances in DNA sequencing technologies, genomic information is being collected at an unprecedented pace, revolutionizing the depth, breadth, and pace of biomedical research. The amount of data that is being produced can be overwhelming. However, IBM researchers are developing and applying both experimental and computational approaches to store and analyze this data, and to assign structural and functional information to key genome components. Advanced algorithms, methods, and tools enable scientists to effectively understand the data produced by genomic analyses, which helps them answer important biological questions.

IBM drives exploratory research at the interface of algorithmics and genomics. We address genomics (and beyond) related questions through mathematical and statistical modeling, combinatorics, and algorithmics. IBM is focusing on these research areas:

- Plant genomics: Analyzing plant traits and populations.
- Human population genomics: Understanding the dynamics of recombining populations, human migratory history, and other stories from genetic data.
- Genomic medicine: Analyzing sequencing data (DNA, RNA, or ChIP) for genomic medicine studies.
- Comparative genomics: Understanding the common genetic architectures across related and distant species.
- Combinatorial patterns: Extensible pattern discovery on biosequences.

For more information, see *Computational Genomics* at:

`http://researcher.watson.ibm.com/researcher/view_group.php?id=1179`

Genomic sequencing can be seen as a first step towards modern medicine. Faster and more affordable next generation sequencing (NGS) is fundamentally transforming the healthcare and life sciences industries. By improving time-to-market for preventive and personalized medicine, companies can save millions of dollars in drug discovery and development while delivering innovative therapies. Nowadays, a single human genome can be sequenced for around $1,000 USD. The Human Genome Project marked the beginning of a new era of personalized medicine where the treatment of disease could be tailored to the unique genetic code of individual patients.

4.2 Healthcare areas

Today, data management, analysis, and reporting in NGS are spread across various software tools and applications that address specific parts of the workflow. Although this offers flexibility in a purely research environment, integrating these tools to work in a production environment is difficult, especially if they use different programming languages, file types, and references. As a consequence, data flow is far from streamlined and the provenance of the data, critical to regulated environments, is difficult to maintain. Smaller laboratories that do not have the advantage of a large IT infrastructure therefore need an integrated hardware and software solution to take advantage of the emerging NGS opportunities. For more information, see *Lab7 for software defined infrastructure* at:

`http://ibm.co/1QDUrD3`

To respond to such needs, IBM proposes Data Engines for specific purposes. For genomics, the IBM Reference Architecture for Genomics, Power Systems edition is a solution that helps

to process raw data with a planned workflow and prepare it for representation or analysis. Furthermore, the outcome can be used as base for translational medicine that focuses on finding relationships between genome and phenotypes, and discovering and developing treatments. The next stage after applying translational medicine methods is personalized healthcare that will bring useful clinical insights to patients. Figure 4-1 visualizes the data flow for modern medicine applications.

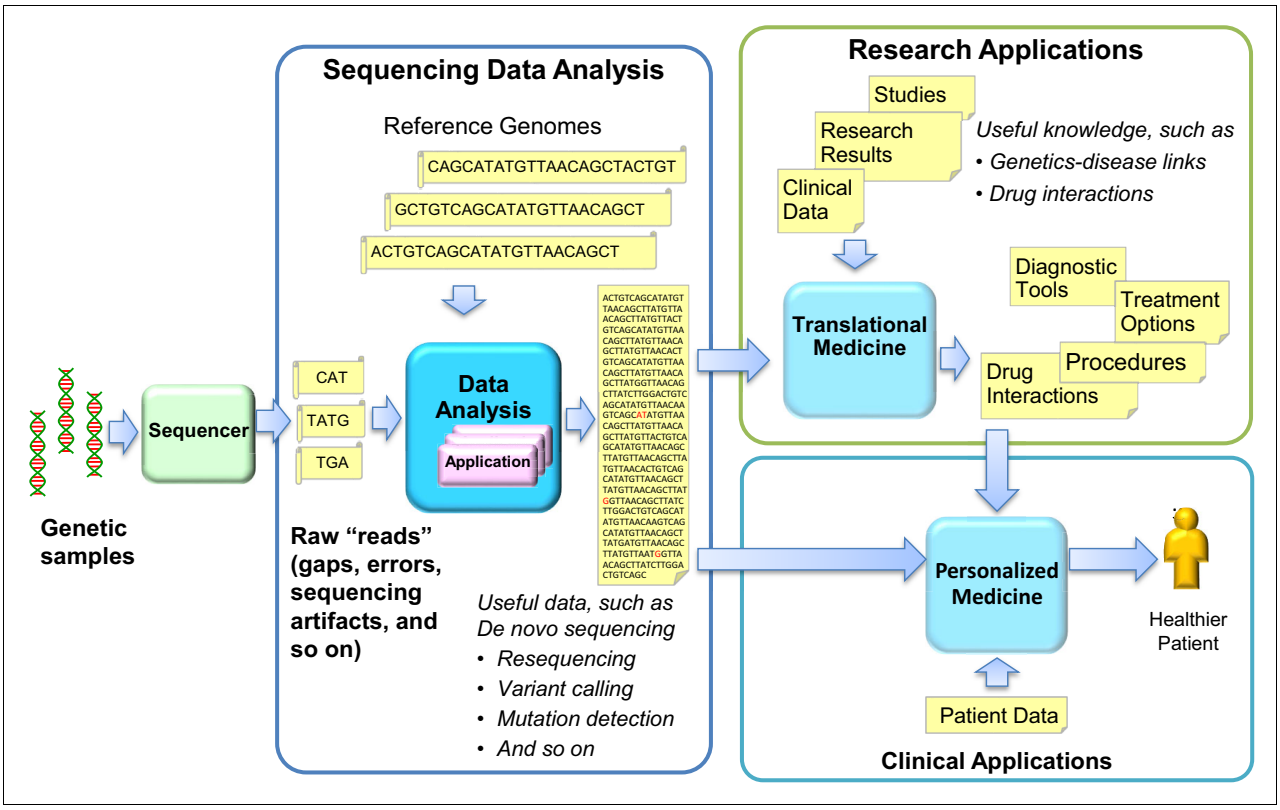


Figure 4-1 Visualizing data flow in modern medicine

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.

After embracing the translational medicine challenges, the move is into personalized healthcare, where advancements in translational medicine, accelerated by NGS technologies, enable health professionals to deliver evidence-based therapeutic intervention to improve the effectiveness of treatments. For more information, see the IBM Healthcare and Life Sciences report, *The evolving promise of genomic medicine. How advanced technologies are transforming healthcare and life sciences*, available at:

http://ibm.co/1OQ5xlP

IBM addresses the genomic medicine market across the entire genomics-healthcare chain that includes the following strategic areas:

- Sequencing, data collection from various sources, analytics, management, and archival.
- Translational medicine by using data integration and mining, which has as an outcome discovery and development of treatments.
- Personalized healthcare with diagnostics and clinical decision support.

For sequencing, IBM proposes scalable high-performance computing (HPC) solutions.

In translational medicine fields, IBM offers software analytics and BigInsights portfolio, data integration tools, and IBM Global Business Services (GBS) services.

IBM is partnering with institutions and their data analysis experts in bioinformatics to accelerate in personalized healthcare. The goal of IBM is to accelerate the development of an open source data warehouse called tranSMART. This knowledge management system has already been adopted by commercial and academic research organizations worldwide as a preferred platform for integrating, accessing, analyzing, and sharing clinical and genomic data about large patient populations. For more information, see the following website:

http://bit.ly/1TChOqQ

The cooperation takes advantage of the speed and scalability of IBM POWER8 server technology, IBM Spectrum Scale storage, and IBM Platform Computing workload management software. The biggest challenge is enabling biomedical researchers to link findings derived from documents in natural language formats (for example, electronic health records and peer reviewed journal articles) with the tranSMART database and analytics functions. IBM and Data Science Institute are integrating tranSMART with IBM Watson™ Text Analytics solutions, and have already demonstrated a first prototype of this enhanced functionality at the Bio-IT World Conference & Expo 2015. The tranSMART Platform v1.2 won the Best-of-Show Award in the Informatics & Data Tools category honoring exceptional innovation in technologies used by life science professionals.[1]

For personalized medicine, IBM has Watson solutions that show a powerful modularity, software packages such as IBM Care Management software, InfoSphere Streams, IBM SPSS®, Cognos, Netezza®, and BigInsights.

SPSS, Cognos, and BigInsights are software packages that show great potential in data analysis and presenting the output. Components of the SPSS and Cognos platforms work with IBM Netezza, InfoSphere BigInsights, and InfoSphere Streams to enable analysts to use powerful analytics tools with big data.

IBM BigInsights for Apache Hadoop helps organizations to cost effectively manage and analyze big data. It has advanced analytics that are built on Hadoop technology (IBM BigInsights Data Scientist module) to meet big data analysis requirements. It includes components designed for performance and usability (IBM BigInsights Analyst module) by using performance optimized capabilities, visualization, rich developer tools, and powerful analytic functions.

IBM Streams is an advanced analytic platform that allows user-developed applications to quickly import, analyze, and correlate information as it arrives from thousands of data stream sources. It provides submillisecond response times, allowing you to view information and events as they unfold. The solution can handle high data throughput rates, up to millions of events or messages per second. For more information about IBM Streams, see the following website:

http://www.ibm.com/software/products/en/ibm-streams

IBM Care Management is an integrated and configurable set of healthcare solutions that provides data-driven population analysis to support patient-centered care processes in healthcare organizations. It is a collaborative solution that allows you to exchange data (patient records, claims forms, and other files from multiple sources) between patients, caregivers, healthcare providers, and payers. You can also analyze it to predict poor outcomes and identify high-risk groups for early intervention.

[1] http://bit.ly/1PFmjaI

Learn more about IBM Care Management at the following website:

http://www.ibm.com/software/products/en/ibm-care-management

Healthcare is getting closer to improving individual health with care management. Better outcomes are achievable with holistic and evidence-based care approaches that span all factors of health: Clinical, genomic, behavioral, environmental, and social context. A large US state implemented an IBM Care Management solution that enables a patient-centered, multidisciplinary team-based approach to assessments, care planning, and delivery. For more information about this success story, see the following website:

http://ibm.co/1OZ7BpD

4.2.1 IBM Watson

Building on strengths in cognitive computing, analytics, security, and cloud, the new Watson Health unit aims to dramatically improve the ability of doctors, researchers, and insurers to innovate by surfacing new insights from the massive amount of personal health data being created daily.

The new cloud-based Watson system is designed to analyze genetic data along with comprehensive biomedical literature and drug databases. Watson can continually “learn” as it encounters new patient scenarios, and as more information becomes available through new medical research, journal articles, and clinical studies. Given the depth and speed of Watson’s ability to review massive databases, the goal of the collaboration is to increase the number of patients who have access to care options tailored to their disease’s DNA.

Watson complements rapid genome sequencing and is expected to dramatically reduce the time it takes to correlate an individual’s genetic mutations with reams of medical literature, study findings, and therapeutic indications that can be relevant. IBM is advancing the quality of healthcare and how these advances can be shared with clients.

The oncology field is one that can quickly experience the benefits from personalized/precision care and a new genomic health record. The New York Genome Center (NYGC) and IBM are involved in a first-of-a-kind program to accelerate personalized, life-saving treatment for cancer patients.[2] The intention is to provide comprehensive information to enable clinicians to consider various treatment options that the clinician can tailor to a patient’s genetic makeup. It will help NYGC scientists understand the data detailing gene sequence variations between normal and cancerous biopsies of brain tumors.

IBM is also working with physicians at Memorial Sloan Kettering Cancer Center in New York, the University of Texas MD Anderson Cancer Center in Houston, and the Cleveland Clinic that provide Watson the training knowledge about cancer. For more information, see the following website:

http://nyti.ms/1PQew9K

Watson can support diagnosis of other diseases. For example, at Boston Children’s Hospital, it helps diagnose and treat a type of kidney disease. In the new collaboration, Watson will be trained in nephrology by reading related medical literature and aggregating information about causative mutations for steroid-resistant nephrotic syndrome (SRNS), a rare genetic form of kidney disease. In September, the two organizations announced they will integrate Watson’s deep and iterative question and answer capability to enhance and scale the OPENPediatrics initiative, an online platform designed to bring life-saving medical knowledge to pediatric caregivers worldwide.[3]

[2] http://www.ibm.com/services/multimedia/GBE03605USEN.pdf
[3] http://openpediatrics.org/

Watson will also team with these groups:

- Apple to collect healthcare data
- Johnson & Johnson to improve care for knee and hip replacements
- Medical equipment manufacturer Medtronic to detect when diabetes patients require adjustments to insulin doses
- CVS to improve services for patients with chronic conditions

In the healthcare industry, the decreased cost of whole genome sequencing, coupled with new advances in cognitive computing and drug discovery, has created a new paradigm where genomic data will soon be combined with electronic health data. In the next three to five years, it is likely that an individual who interacts with the healthcare system will have a genome-based electronic record or *genomic health record*. It will be used in tandem with other electronic healthcare tools for decision support, prevention, and customized testing and treatments. At first, the record cannot contain the whole genome sequence, but instead perhaps the results from testing specific genes. This genomic health record will be combined with mobile, social, cloud, and analytics.

At the World of Watson opening, Ginni Rometty said that 5,000 companies from 23 industries were in the pipeline to partner with Watson. She announced Watson Genomic Analytics, introduced collaborations with 14 cancer centers, and announced a partnership with Epic to integrate Watson into Electronic Health Record (EHR) systems. For more information, see *IBM Announces Watson Genomic Analytics; Collaboration with 14 Cancer Centers* at:

http://bit.ly/1QjUUrF

Watson Health Cloud is one of the largest and fastest growing pools of individual data. IBM plans to use the Watson Health Cloud to analyze and cross-reference medical images against a deep trove of lab results, electronic health records, genomic tests, clinical studies and other health-related data sources. It already represents 315 billion data points and 90 million unique records. Insights generated by Watson could then help healthcare providers in fields that include radiology, cardiology, orthopedics, and ophthalmology to pursue more personalized approaches to diagnosis, treatment, and monitoring of patients. For more information, see *IBM Closes Deal to Acquire Merge Healthcare* at:

http://www.ibm.com/press/us/en/pressrelease/47839.wss

Note: Follow the latest blog posts from IBM healthcare and Life Sciences industry thought leaders at the following website:

http://www.ibm.com/industries/lifesciences/blog.html

4.2.2 Apache Spark

Apache Spark is an open-source cluster computing framework with in-memory processing to speed analytic applications up to 100 times faster than technologies on the market today.[4] For IBM, Apache Spark is a key project because of its speed and ease of use, and because it integrates easily with other open source components. IBM is working with Spark to include genome sequencing for personalized medicine through the Adam project at the University of California, Berkeley. Spark is also being adapted for the early detection of conditions such as diabetes by analyzing patient medical data. For more information, see *Apache community looks to address big data's 'unicorn' problem* at:

http://bit.ly/1OZ7EBM

[4] http://www.ibm.com/analytics/us/en/technology/spark/

4.3 Read and follow

IBM has a suite of solutions to create a solid foundational data and analytics platform to improve health outcomes and reduce costs. Find more and order a workshop from the IBM Analytics for healthcare website at:

http://www.ibm.com/analytics/us/en/industry/healthcare/

The latest blog posts from IBM healthcare and Life Sciences industry thought leaders can be read at the following website:

http://www.ibm.com/industries/lifesciences/blog.html

A

Useful software information

This appendix provides useful information including website to find utilities to be incorporated as part of the solution.

This appendix includes the following section:

- Genomics package

Genomics package

The following are a few software packages that were used during the residency implementation and testing. Additional websites are included to provide more information about the software, and how to procure it.

- Biobuilds package

  https://www.lab7.io/solutions/biobuilds/

- Genome Analysis Toolkit (GATK)

  https://www.broadinstitute.org/gatk/download/

  Note: To download the software, you must be logged into the forums to proceed. In the section *Download the GATK*, select **GATK**. Read and accept the license agreement in order to download GATK.

- IBM Life Science Platform Provisioning package

  The Life Science provisioning package is available from Fix Central at the following website:

  http://www.ibm.com/support/fixcentral/options

  Download the package by using the *Search Fix Central* form. Look for the `LSSolInst3.0.tar` file as shown in Figure A-1.

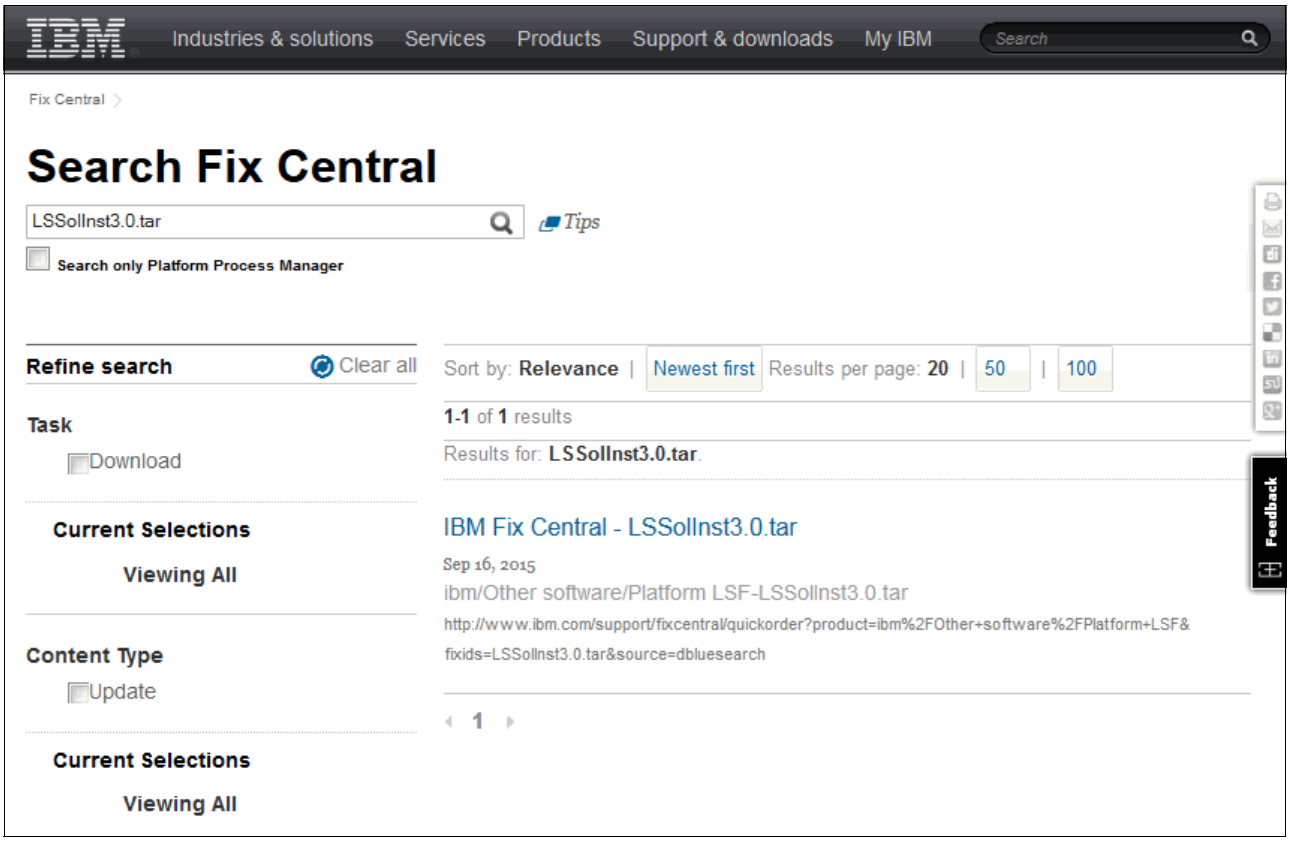


Figure A-1 Search for IBM Life Science provisioning package on Fix Central

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *IBM Platform Computing Solutions for High Performance and Technical Computing Workloads*, SG24-8264
- *IBM Platform Computing Solutions Reference Architectures and Best Practices*, SG24-8169
- *IBM Power Systems S812L and S822L Technical Overview and Introduction*, REDP-5098
- *IBM Reference Architecture for Genomics: Speed, Scale, Smarts*, REDP-5210
- *IBM Spectrum Scale (formerly GPFS)*, SG24-8254
- *IBM Technical Computing Clouds, SG24-8144*

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:

- *Genomics and World Health: Report of the Advisory Committee on Health research*, Geneva, WHO (2002):

  http://apps.who.int/iris/bitstream/10665/42453/1/a74580.pdf

Online resources

These websites are also relevant as further information sources:

- IBM Healthcare and life sciences solutions

  http://www.ibm.com/systems/power/solutions/industry/healthcare.html

- IBM Data Engine for Genomics – Power Systems Edition Architecture and Reference Design Version 3.1

  http://ibm.co/1i1NPzi

- OpenPOWER

  http://openpowerfoundation.org/blogs/imperial-college-london-and-ibm-join-forces-to-accelerate-personalized-medicine-research-within-the-openpower-ecosystem/
- IBM Knowledge Center

  http://www.ibm.com/support/knowledgecenter
- IBM Fix Central

  http://www.ibm.com/support/fixcentral/

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

IBM Reference Architecture for Genomics, Power Systems Edition

(0.2”spine)
0.17”<->0.473”
90<->249 pages

SG24-8279-00

ISBN 0738441635

Printed in U.S.A.