

IBM Spectrum Scale Best Practices for Genomics Medicine Workloads

Joanna Wong

Kevin Gildea

Kumaran Rajaram

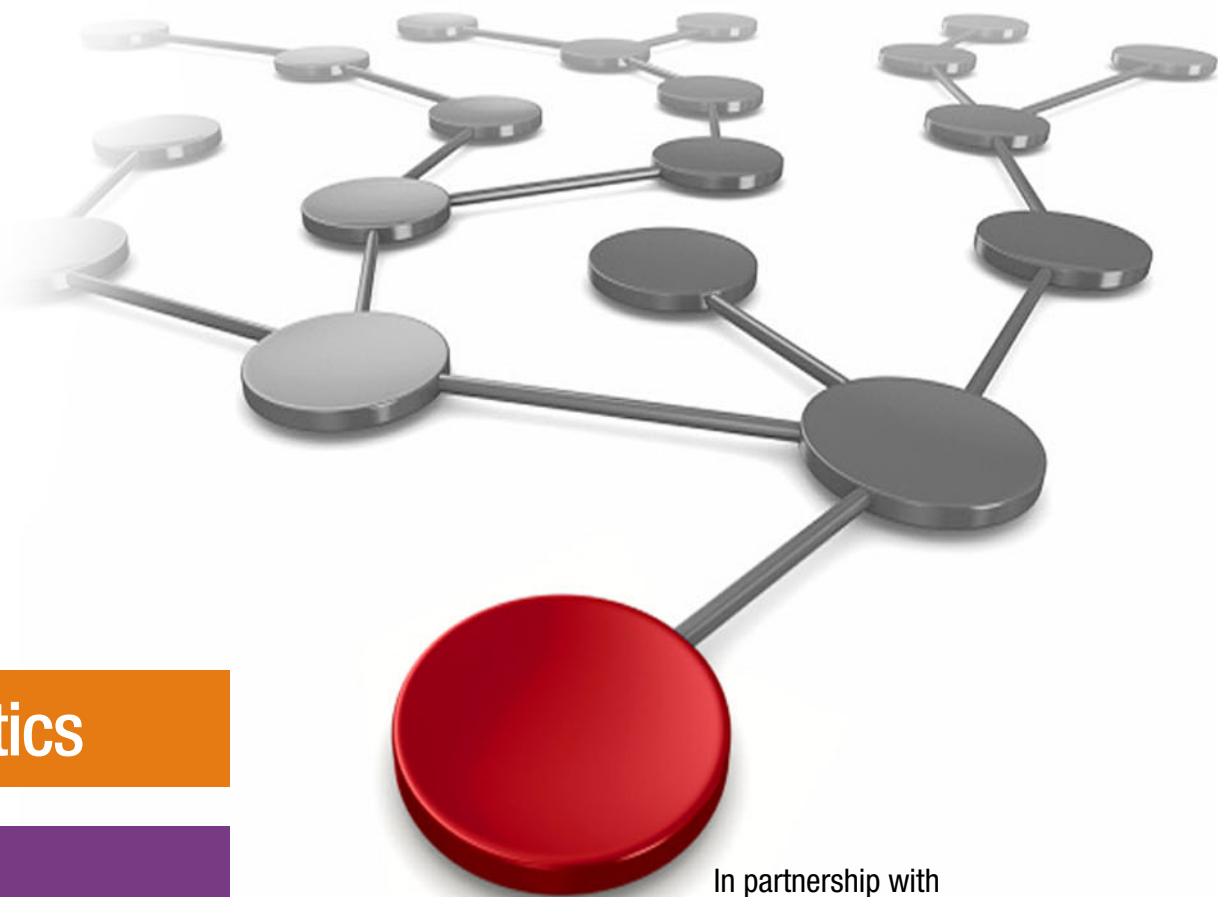
Luis Bolinches

Monica Lemay

Piyush Chaudhary

Sandeep R. Patil

Ulf Troppens



 **Analytics**

Storage

In partnership with
IBM Academy of Technology



International Technical Support Organization

**IBM Spectrum Scale Best Practices for Genomics
Medicine Workloads**

April 2018

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

Second Edition (April 2018)

This edition applies to Version 4, Release 2, Modification 3 of IBM Spectrum Scale and Version 5, Release 2, Modification 0 of IBM Elastic Storage Server

This document was created or updated on April 25, 2018.

© Copyright International Business Machines Corporation 2017, 2018. All rights reserved.

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

Contents

Notices	v
Trademarks	vi
Preface	vii
Authors	viii
Now you can become a published author, too	ix
Comments welcome	x
Stay connected to IBM Redbooks	x
Summary of changes	xi
April 2018, Second Edition	xi
December 2017, First Edition	xi
Chapter 1. The IBM Spectrum Scale Blueprint for Genomics Medicine Workloads . . .	1
1.1 Genomics medicine	2
1.1.1 Genomics medicine overview	2
1.1.2 Genomics workload	3
1.2 Solution approach	3
1.2.1 Composable infrastructure	3
1.2.2 Composable building blocks	4
1.2.3 Driven by design thinking	5
1.2.4 Driven by agile development	5
1.3 Blueprint capabilities	6
1.3.1 Capabilities of the compute services	6
1.3.2 Capabilities of the storage services	7
1.3.3 Capabilities of the private network services	7
1.4 Example environment	7
1.4.1 Physical configuration	8
1.4.2 Logical configuration	9
Chapter 2. The compute services	11
2.1 Overview	12
2.1.1 Capabilities and solution elements	12
2.1.2 Software levels	13
2.2 Application layer	13
2.2.1 The Broad Institute GATK	13
2.3 Orchestration layer	16
2.3.1 IBM Spectrum LSF	16
2.4 Data layer	18
2.5 General recommendations	18
2.5.1 Designation of the compute nodes	19
2.5.2 IBM Spectrum Scale node roles	20
2.5.3 IBM Spectrum LSF host types	21
2.5.4 IBM Spectrum LSF add-ons	21
2.5.5 External dependencies	22
2.5.6 Communication and security aspects	22
2.6 Tuning	23
2.6.1 Operating system	23
2.6.2 Network	23

2.6.3 IBM Spectrum Scale	24
2.7 Monitoring	25
Chapter 3. The storage services	27
3.1 Overview	28
3.1.1 Capabilities and solution elements	28
3.1.2 Software levels	29
3.2 File storage layer	29
3.2.1 IBM Spectrum Scale file systems	29
3.2.2 Recommendations for genomics medicine workloads	31
3.2.3 IBM Spectrum Scale filesets	34
3.3 Block storage layer	35
3.3.1 IBM Elastic Storage Server	36
3.4 File access layer	37
3.4.1 NFS and SMB	37
3.5 General recommendations	37
3.5.1 Recommendations for IBM Spectrum Scale	38
3.5.2 External dependencies	39
3.5.3 Communication and security aspects	39
3.6 Data management	39
3.7 Tuning	40
3.7.1 IBM Elastic Storage Server	40
3.7.2 Protocol nodes	41
3.8 Monitoring	42
Chapter 4. The private network services	45
4.1 Overview	46
4.1.1 Capabilities and solution elements	46
4.1.2 Shared network	47
4.2 High-speed data network	49
4.2.1 IBM Spectrum Scale network requirements	49
4.2.2 Recommendations for genomics medicine workloads	50
4.2.3 Miscellaneous comments	51
4.3 Management networks	51
4.3.1 Provisioning network	51
4.3.2 Service network	52
4.4 Network designs	52
4.4.1 Network design for small configuration	52
4.4.2 Network design for large configuration	53
Appendix A. Profiling GATK	55
Related publications	61
Online resources	61
Help from IBM	62

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.

Redbooks (logo) ®
AIX®
developerWorks®
GPFS™
IBM®

IBM Elastic Storage™
IBM Spectrum™
IBM Spectrum Scale™
LSF®
Power Systems™

Redbooks®
Redpaper™
Redpapers™

The following terms are trademarks of other companies:

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

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

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

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

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

Preface

Advancing the science of medicine by targeting a disease more precisely with treatment specific to each patient relies on access to that patient's genomics information and the ability to process massive amounts of genomics data quickly. Although genomics data is becoming a critical source for precision medicine, it is expected to create an expanding data ecosystem.

Therefore, hospitals, genome centers, medical research centers, and other clinical institutes need to explore new methods of storing, accessing, securing, managing, sharing, and analyzing significant amounts of data. Healthcare and life sciences organizations that are running data-intensive genomics workloads on an IT infrastructure that lacks scalability, flexibility, performance, management, and cognitive capabilities also need to modernize and transform their infrastructure to support current and future requirements.

IBM® offers an integrated solution for genomics that is based on composable infrastructure. This solution enables administrators to build an IT environment in a way that disaggregates the underlying compute, storage, and network resources. Such a composable building block based solution for genomics addresses the most complex data management aspect and allows organizations to store, access, manage, and share huge volumes of genome sequencing data.

IBM Spectrum™ Scale is software-defined storage that is used to manage storage and provide massive scale, a global namespace, and high-performance data access with many enterprise features. IBM Spectrum Scale™ is used in clustered environments, provides unified access to data via file protocols (POSIX, NFS, and SMB) and object protocols (Swift and S3), and supports analytic workloads via HDFS connectors. Deploying IBM Spectrum Scale and IBM Elastic Storage™ Server (IBM ESS) as a composable storage building block in a Genomics Next Generation Sequencing deployment offers key benefits of performance, scalability, analytics, and collaboration via multiple protocols.

This IBM Redpaper™ publication describes a composable solution with detailed architecture definitions for storage, compute, and networking services for genomics next generation sequencing that enable solution architects to benefit from tried-and-tested deployments, to quickly plan and design an end-to-end infrastructure deployment. The preferred practices and fully tested recommendations described in this paper are derived from running GATK Best Practices work flow from the Broad Institute.

The scenarios provide all that is required, including ready-to-use configuration and tuning templates for the different building blocks (compute, network, and storage), that can enable simpler deployment and that can enlarge the level of assurance over the performance for genomics workloads. The solution is designed to be elastic in nature, and the disaggregation of the building blocks allows IT administrators to easily and optimally configure the solution with maximum flexibility.

The intended audience for this paper is technical decision makers, IT architects, deployment engineers, and administrators who are working in the healthcare domain and who are working on genomics-based workloads.

Authors

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

Joanna Wong is an Executive IT Specialist with the IBM Systems Client Centers. She has extensive experience in high-performance computing (HPC) application optimization and solution architecture implementation, recently focusing on software-defined solutions in life sciences. She has an A.B. in Physics from Princeton University, M.S., a Ph.D. from Cornell University, and an MBA from Walter Haas School of Business (University of California, Berkeley).

Kevin Gildea is a Distinguished Engineer with extensive experience in distributed computing, cluster computing, and high-performance interconnects. Currently, he contributes to the development of IBM Spectrum Scale. He holds a PhD in Computer Science from Rensselaer Polytechnic Institute and a BS in Computer Science from the University of Scranton.

Kumaran Rajaram is a Performance Engineer in the IBM Spectrum Scale Development team. He has more than 15 years of professional experience in the field of high-performance computing, data-intensive compute environments, parallel I/O middleware, and cluster- and SAN-based file systems, specifically IBM Spectrum Scale (formerly IBM GPFS™) and scalable storage solutions. Kumaran holds a Master's Degree in Computer Science from Mississippi State University.

Luis Bolinches has worked with IBM Power Systems™ servers for over 15 years and with IBM Spectrum Scale (GPFS) for over 10 years. He worked with IBM in Spain, Estonia, and currently in Finland. In addition to his IBM experience, he worked several years at the European Council for Nuclear Research (CERN). He is the co-author of *IBM Reference Architecture for Genomics, Power Systems Edition*, SG24-8279.

Monica Lemay is a Software Test Engineer who has been involved with driving system quality and defect resolution in various IBM projects during her career, including IBM AIX®, Red Hat Linux, NFS, SMB, and IBM Spectrum Scale and IBM Spectrum Scale AFM/ADR.

Piyush Chaudhary is a Senior Technical Staff Member who works as a Big Data and Analytics Architect in the IBM Spectrum Scale development team. He has over 18 years of product architecture and design experience. Piyush has an extensive background in high performance computing and distributed systems. His current focus is on enabling big data frameworks like Hadoop and Spark on IBM Spectrum Scale.

Sandeep R. Patil is a Senior Technical Staff Member who works as a Storage Architect with IBM System Labs. He has over 15 years of product architecture and design experience. Sandeep is an IBM Master Inventor, an IBM developerWorks® Master Author, and a member of the IBM Academy of Technology. Sandeep holds a Bachelor of Engineering (Computer Science) degree from the University of Pune, India.

Ulf Troppens is a Consulting IT Specialist with IBM Spectrum Scale Development. He embraces to lead complex proof-of-concepts to examine, elaborate, and validate complex requirements in close collaboration with customers. Ulf has a specific interest in file- and object-based workflows for data intensive science, where data needs to be acquired from instruments and then transferred to a data center for analysis and archive. Ulf co-authored the award-winning book *Storage Networks Explained: Basics and Application of Fibre Channel SAN, NAS, iSCSI, InfiniBand and FCoE*.

Special thanks to Tomer Perry, IBM Spectrum Scale Development, who made a significant contribution to the development of this book by extensively consulting and guiding the team on general preferred practices for the solution design.

Special thanks to Frank Lee, IBM High Performance Data Analytics WW Sales Leader and Chief Architect, IBM Reference Architecture for Genomics, Richard Rupp, Sales Specialist IBM Public Sector, and Constantine Arnold, IBM Almaden Research Center, who made a significant contribution to the development of this book by extensively consulting and guiding the team on the capabilities that the genomics blueprint must provide to be relevant for the market.

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

Denise Ruffner, Janis Landry-Lane, Yinhe Cheng
IBM Healthcare and Life Science

Christof Westhues
IBM Spectrum LSF® Tech Seller

Chris Maestas
IBM Spectrum Scale Solution Architect

Felipe Knop, Ingo Meents, Jay Vaddi, John Lewars, Lyle Gayne, Puneet Chaudhary, Scott Fadden, Steve Xiao, Ted Hoover
IBM Spectrum Scale Development

Steve Zehner
IBM IT Systems Architect, Public Market

Larry Coyne, Ann Lund, Debbie Willmschen
IBM International Technical Support Organization

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 papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks® publications in one of the following ways:

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

ibm.com/redbooks

- Send your comments in an email:

redbooks@us.ibm.com

- Mail your comments:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

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

- Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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

Summary of changes

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

Summary of Changes
for IBM Spectrum Scale Best Practices for Genomics Medicine Workloads
as created or updated on April 25, 2018.

April 2018, Second Edition

This revision reflects the addition, deletion, or modification of new and changed information, which is summarized here.

Changed Information

- ▶ Minor corrections and clarifications

December 2017, First Edition

New information

- ▶ GATK3 on Power8
- ▶ IBM Spectrum Scale Release 4.2.3.4
- ▶ IBM Elastic Storage Server 5.2



The IBM Spectrum Scale Blueprint for Genomics Medicine Workloads

This chapter introduces the IBM Spectrum Scale Blueprint for Genomics Medicine Workloads. It includes the following topics:

- ▶ Genomics medicine
- ▶ Solution approach
- ▶ Blueprint capabilities
- ▶ Example environment

1.1 Genomics medicine

This section briefly introduces the concept of *genomics medicine* and characterizes a *genomics medicine workload*.

1.1.1 Genomics medicine overview

Genomics is a branch of biotechnology that focuses on genomes. A *genome* is an organism's complete set of DNA and contains all the information that is needed to build and maintain that organism. A *gene* is a segment of the DNA that instructs the cell about how to put together the building blocks for one specific protein, which are the functional molecules of the cell. Table 1-1 provides information about the relationship between these components.

Table 1-1 The relationship between genomes, DNA, genes, and proteins

Term	Description
Genome	<ul style="list-style-type: none">▶ A genome is the genetic material of an organism.▶ The human genome is composed of 23 chromosomes.
DNA	<ul style="list-style-type: none">▶ Each chromosome is an organized structure with a DNA molecule.▶ DNA molecules are made up of nucleobases that are abbreviated as <i>A</i>, <i>T</i>, <i>C</i>, and <i>G</i>.
Gene	<ul style="list-style-type: none">▶ A DNA region that influences a function in an organism is called <i>gene</i>.▶ There are about 20,000 genes in the human genome.▶ Genes encode <i>proteins</i>.
Proteins	<ul style="list-style-type: none">▶ Proteins have a biological function in the cell.

Genomics involves applying the techniques of genetics and molecular biology to sequence, analyze, or modify the DNA of an organism. It finds its use in many fields, such as diagnostics, personalized healthcare, agricultural innovation, forensic science, and others.

The impact of genomics information and technology has the potential to improve healthcare outcomes, quality, and safety and to result in cost savings. For example:

- ▶ Genomics is being used to predict which treatment option is likely to be most effective for a specific person and how the specific person is likely to respond, which can lead to personalized medicine.
- ▶ Genomics is set to transform the way we diagnose and treat infectious diseases. If we have a patient with tuberculosis, sequencing the genome of the microbe can be used to predict which antimicrobials are most likely to work. In public health, genomics information can allow tracking and planning of strategies to combat potential epidemics.
- ▶ Genomics is already being used for addressing cancer, common complex diseases, and rare diseases.

Genomics is becoming essential for data scientists and physicians for investigation and precision medicine. Moreover, advances in the fields of diagnostics, drug discovery and development, personalized medicine, and agriculture and animal research are triggering the market growth for genomics products.

1.1.2 Genomics workload

IT administrators, physicians, data scientists, researchers, bioinformaticians, and other professionals who are involved in the genomics workflow need the right foundation to achieve their objectives efficiently while improving patient care and outcomes. Thus, it is important to understand the different stages of the genomics workload and the key characteristics of it.

Genomics requires a significant focus on big data management as the sequencing of the genome results in the production of a large amount of data. Genomics data analysis requires the following processing steps:

1. Sequencers convert the physical sample to raw data.
2. Raw data is put in a sequence corresponding to the genome.
3. Analytics (for example, matching mutations with certain diseases) is performed.

Raw data of a single sample is up to a few hundred gigabytes of data. For average organizations, the total data that needs to be acquired, stored, analyzed, managed, and archived sums up to tens of petabytes.

From an IT perspective, genomics medicine relates to technical computing. The workload is characterized by high-throughput execution of batch jobs where scalability and performance are key drivers. The workload poses I/O performance, data management, multiprotocol support, and big data integration challenges on storage systems. The intense processing part of the workflow sets high expectations on the compute nodes.

Optimization of workload orchestration and collaboration along with greater resource allocation are critical capabilities that must be addressed in the compute layer to manage the genomics analysis workflow efficiently. The network becomes a critical conduit component across optional professional services for an optimal genomics deployment.

1.2 Solution approach

The previous section explained that a sizeable IT infrastructure in terms of compute, storage, and network resources is required to enable genomics medicine. This section describes the approach for creating a respective IT infrastructure. It includes the following topics:

- ▶ Composable infrastructure
- ▶ Composable building blocks
- ▶ Driven by design thinking
- ▶ Driven by agile development

1.2.1 Composable infrastructure

The previous section explained that genomics medicine depends on a sizable IT infrastructure that scales up to petabytes (PBs) of storage to own, manage, and access genomics data and up to hundreds of servers to analyze the genomics data. Even though, customers have varying performance and functional needs.

For example, the data is generated by different kinds of instruments. Although most of the genomics data today is generated by genome sequencers, super microscopes are emerging as data intensive instruments. Modern cryo-electron microscopes generate data with up to 5 GBps that must be acquired, stored, managed, and analyzed.

That data quickly sums up to PBs of data that need to be kept online for analysis and archived to support preferred practices in research and regulatory requirements in drug development and translational medicine. The instruments might be placed in the same building where the storage and compute infrastructure is hosted. However, in many cases, the data is acquired on remote sites or is provided by a remote partner.

There are also variances in the application workload. Although most workflows that analyze data that is acquired from genome sequencers are based on the Broad Institute Genome Analysis Toolkit (GATK), there is an ecosystem of several hundred applications. Research is ongoing to improve selected steps of the workflow by adjusting mathematical algorithms to increase the quality of statistical results and by trying to use new data analysis technologies, such as Apache Spark, machine learning, and deep learning.

Last but not least, many customers have an existing infrastructure, such as servers and network, authentication services, monitoring, and tooling, to deploy and configure operating systems and applications. Solutions for storing and analyzing genomics data need to integrate into the customer's existing infrastructure and services to protect investments.

1.2.2 Composable building blocks

A flexible solution architecture is required to support such varying requirements. This paper provides *composable building blocks* for the compute services, the storage services, and the private network services (Figure 1-1). The components of the blueprint interact with the external shared network (for example, the data center network). The external shared network provides a high-speed Network File System (NFS) and Server Message Block (SMB) data access and enables a user login to submit and manage batch jobs and to access interactive applications.

The compute services provide a scalable compute cluster to analyze genomics data. The storage services provide a scale-able storage cluster to store, manage and access genomics data. The private network services provide a high-speed data network that is not connected to the data center network nor any other shared external network. In addition, the private network services provide provisioning networks and service networks for administrative login and hardware services. The provisioning networks and the service networks are optionally connected to external shared network.

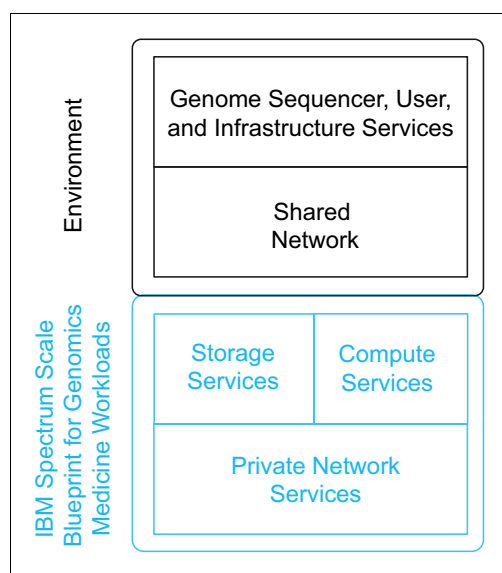


Figure 1-1 Composable building blocks for genomics medicine workloads

Most of those building blocks are self-contained and have well-defined interfaces to the other building blocks. This design allows you to replace or adjust one building block, such as selecting different storage systems or adding servers with GPUs, without requiring all other solution elements to change. Ideally there is no impact or change to the other building blocks.

1.2.3 Driven by design thinking

Customers are overwhelmed by the sheer amount of data that is generated by instruments like genome sequencers and cryo-electron microscopes. A Design Thinking Workshop was conducted to better understand the market requirements. In the workshop we have identified a couple of insights that needs to be considered for the solution architecture of the composable building blocks.

Many genomics medicine customers quickly realize that they need to deploy and operate sizeable IT infrastructure. In addition, the average IT administrators and user in this field do not have the IT skills or experience on such a scale. Thus, they benefit from the efficiency and scalability principles from high-performance computing (HPC) in fields where IBM Spectrum Scale is deployed successfully. However, the IBM Spectrum Scale Blueprint for Genomics Medicine Workloads blueprint needs to favor ease of use and operational efficiency over best possible performance or lowest acquisition costs.

Advanced genomics medicine customers are outgrowing NAS storage. The move from a traditional NAS system or a modern scale-out NAS system to a parallel file system such as IBM Spectrum Scale requires a new set of skills. Thus, the IBM Spectrum Scale Blueprint for Genomics Medicine Workloads needs to provide basic background information and offer optional professional services to successfully transition to the new infrastructure.

Therefore, for each of the expertly engineered composable building blocks illustrated in Figure 1-1 on page 4, the IBM Spectrum Scale Blueprint for Genomics Medicine Workloads blueprint provides the following benefits:

- ▶ *Best practice guides* that describe architecture and configuration settings
- ▶ *Runbooks* that describe in detail how to install, configure, monitor, and upgrade selected example configurations
- ▶ *Sizing guidelines* that help to define a solution that meets the customer's performance requirements
- ▶ *Education materials* for deployment workshops that guide the customer to customize the general preferred practices to the client's specific needs

This paper is based on the Broad Institute GATK [best practices](#). The runbooks, the sizing guidelines, and the educational materials for deployment workshops are provided by different documents. Contact IBM to get access to those complementary documents.

1.2.4 Driven by agile development

The IBM Spectrum Scale Blueprint for Genomics Medicine Workloads blueprint is delivered in phases (as shown in Figure 1-2 on page 6). Phase I describes a minimal viable product (MVP) to acquire, store, manage, and analyze genomics data. It is envisioned that the blueprint will be extended to add more functional capabilities and to improve operational efficiency (as shown in Phases II and III).

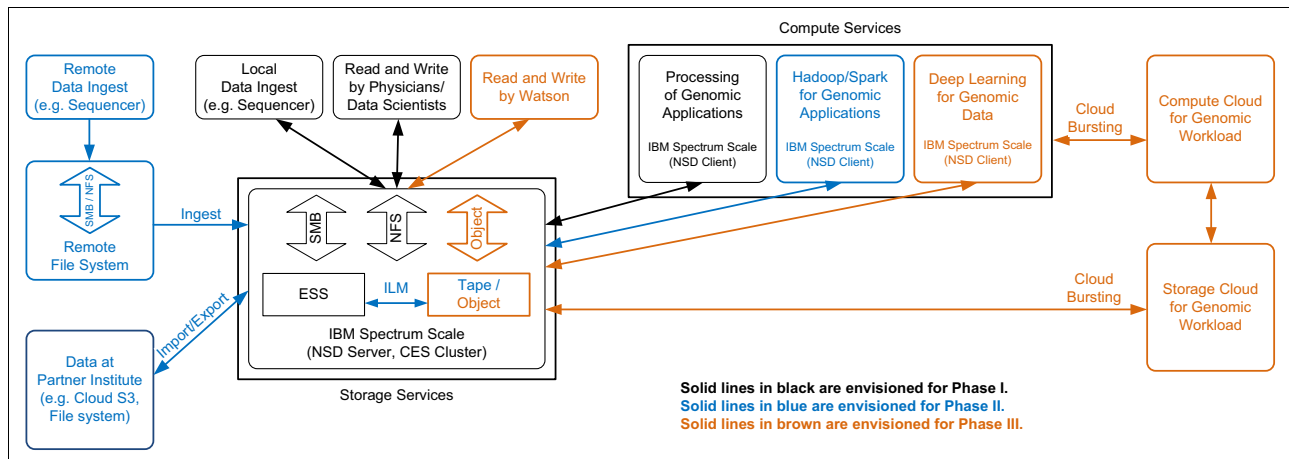


Figure 1-2 Delivery phases and their capabilities

Note: IBM Spectrum Scale already supports the features of Phase II and III; however, we have not yet evaluated these features in the context of this paper. Therefore, outside the scope of the blueprint means that the blueprint does not provide recommendations about the respective features and capabilities although they are already available. Contact IBM to get help with the configuration and deployment of such features.

1.3 Blueprint capabilities

The capabilities of Phase I have been reviewed with and prioritized by the IBM Healthcare and Life Science sales team. They are written in a product neutral language to emphasize user requirements. This section describes the capabilities of the compute services (Section 1.3.1), the storage services (Section 1.3.2), and the private network services (Section 1.3.3) building blocks. Subsequent chapters dive into the details of each building block.

1.3.1 Capabilities of the compute services

To enable the analysis of genomics data, the compute services provide the following capabilities:

- ▶ A user GUI for physician and data scientist to submit and manage batch jobs and to create and manage custom workflows
- ▶ A Workloads Management GUI for the IT administrator to view cluster status and use
- ▶ Secure, high-speed access to files stored on storage cluster
- ▶ Scaling via a Workload Scheduler that enables high-throughput execution of batch jobs
- ▶ Performance and tuning recommendations that support the [preferred practices](#) of the Broad Institute GATK.
- ▶ Support of IBM Power, x86-64, or mixed IBM Power and x86-64 environment for batch processing and for interactive login to access resources.

1.3.2 Capabilities of the storage services

To enable access to genomics data, the storage services provide the following capabilities:

- ▶ Data transfer nodes for secure high-speed external access via NFS and SMB to ingest data from genome sequencers, microscope, and so on, for access by data scientists and physicians and for sharing across sites and institutions
- ▶ Secure high-speed internal access for analysis on the compute cluster

To effectively store and manage genomics data, the storage cluster also provides the following capabilities:

- ▶ Scale-out architecture that is capable of storing from a few 100 TB to tens of PB of genomics data
- ▶ End-to-end checksum to ensure the data integrity all the way from the application to the disks
- ▶ A data management GUI to configure and monitor storage resources
- ▶ Optional professional services, ranging from management of daily operation to consultancy for major configuration changes

1.3.3 Capabilities of the private network services

To integrate the compute services and the components of the storage services into an IT infrastructure solution for genomics medicine workloads, the private network services provides the following capabilities:

- ▶ A high-speed data network for fast and secure access to genomics data:
 - Storage nodes are connected to the network with at least two links for high availability.
 - Compute nodes are connected to the network with one port or with two ports if you want high availability.
- ▶ Provisioning networks for provisioning and in-band management of the storage and compute components and for administrative login
- ▶ Service networks for out-band management and monitoring of all solution components
- ▶ A scalable design that can start small and grow to a large configuration that consists of hundreds of compute nodes and tens of PB of storage

1.4 Example environment

This section illustrates the IBM Spectrum Scale Best Practices for Genomics Medicine Workloads using an example environment. It describes the environments physical configuration and its logical configuration. Subsequent chapters provide more details, including software levels and configuration settings.

1.4.1 Physical configuration

The example environment consists of two IBM Elastic Storage Servers (ESS) and 16 servers that will be configured in different roles (as listed in Table 1-2). All servers are connected via InfiniBand and Gigabit Ethernet. This configuration is derived from real deployments and is supported by the engineering that is described in the next chapters.

Table 1-2 The example environment storage, servers, and switches

Storage services <ul style="list-style-type: none">▶ 1x ESS Management Server (EMS)▶ 1x IBM Elastic Storage Server (ESS) GS2S with SSD▶ 1x IBM Elastic Storage Server (ESS) GL6S with NL-SAS▶ 3x Cluster Export Services (CES) protocol nodes for NFS and SMB	Compute services <ul style="list-style-type: none">▶ 2x management nodes▶ 1x user node (Power)▶ 7x compute nodes (Power)▶ 2x compute nodes (Intel)
Private network services 2x InfiniBand EDR switches 1x Gigabit Ethernet switch	

The storage services of the example environment includes the following components:

- ▶ 1x ESS Management Server (EMS) to manage the storage cluster
- ▶ 1x IBM Elastic Storage Server (ESS) GS2S with SSD to store metadata
- ▶ 1x IBM Elastic Storage Server (ESS) GL6S with NL-SAS to store genomics data
- ▶ 3x CES protocol nodes for NFS and SMB to ingest and access genomics data

The compute services of the example configuration include the following components:

- ▶ 2x Management nodes to manage the compute cluster
- ▶ 1x user node (Power) for user login and to start batch jobs
- ▶ 7x compute nodes (Power) to analyze genomics data
- ▶ 2x compute nodes (Intel) to analyze genomics data

The private network services of the example configuration include the following components:

- ▶ 2x InfiniBand EDR switches for the high-speed data network
- ▶ 1x Gigabit Ethernet switch for the provisioning network and the service network

The different server types and roles are explained in Chapter 2 and Chapter 3.

1.4.2 Logical configuration

All physical components are integrated to build a software-defined infrastructure that is optimized to run genomics medicine workload (as shown in Figure 1-3).

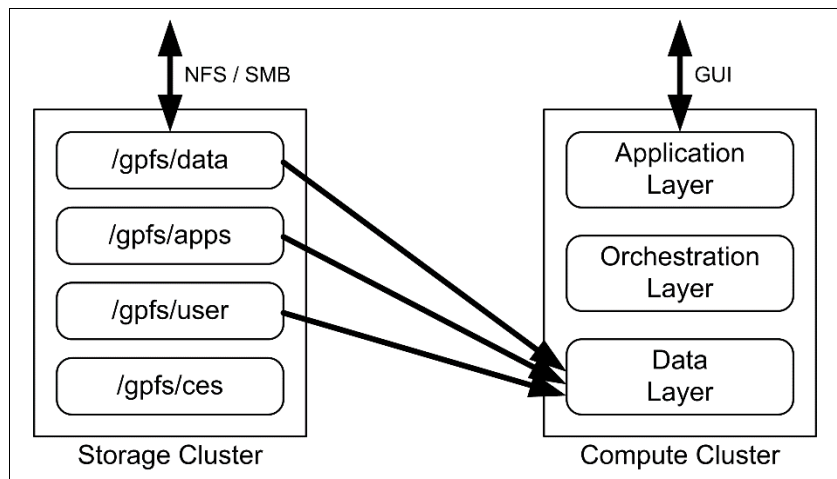


Figure 1-3 Software-defined infrastructure for genomics medicine workloads

All EMS, ESS, and CES protocol nodes build a IBM Spectrum Scale storage cluster.

The storage capacity of the two ESS are used to create the following file systems:

/gpfs/data	To store genomics data
/gpfs/apps	To store application binaries, scripts, configuration files and logs
/gpfs/user	To store user data for the execution of batch jobs
/gpfs/ces	To store metadata for the CES

The three CES protocol nodes build a CES cluster and are configured to provide NFS and SMB services. The CES cluster is part of the IBM Spectrum Scale storage cluster.

The /gpfs/data file system is exported via NFS and SMB for data ingest from devices such as genome sequencers and microscopes and for access by data scientists and physicians.

All nodes of the compute cluster build an IBM Spectrum Scale compute cluster.

The IBM Spectrum Scale compute cluster imports the /gpfs/data, /gpfs/apps, and /gpfs/user file systems via an IBM Spectrum Scale multi-cluster remote cluster mount.

All compute resources provided by the compute nodes are managed by IBM Spectrum LSF to enable high-throughput execution of batch jobs. IBM Spectrum LSF provides a workload management GUI to submit and manage batch jobs, to analyze genomics data, and to create and manage customer workflows.

The chapters that follow describe the rationale and the details of this configuration.

The compute services

The purpose of the compute services is to enable analysis of the genomics data that is ingested via the storage services. The design of this piece of the composable infrastructure provides for high performance analysis, scaling of the analysis workload, and ease of use by physicians and data scientists. The compute services design is based on six expertly engineered building blocks that enable IT architects to compose solutions that meet customers varying performance and functional needs (Figure 2-1).

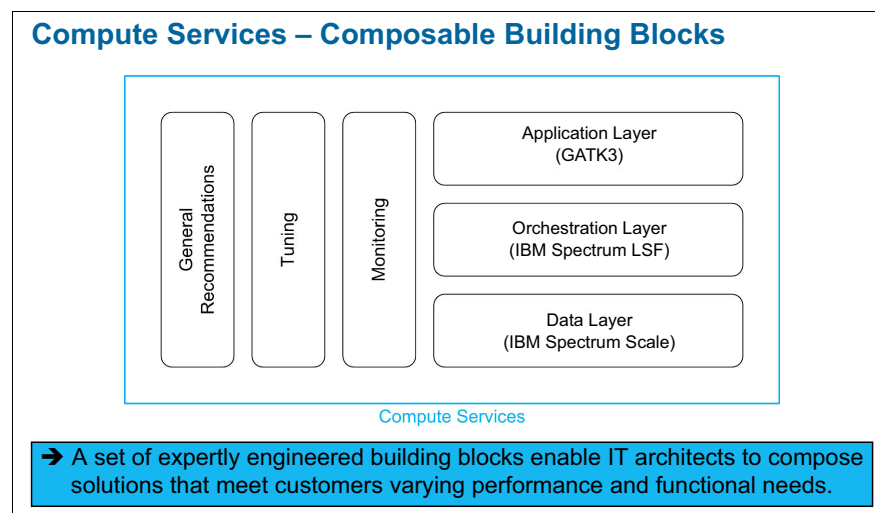


Figure 2-1 The six expertly engineered building blocks of the compute services

This chapter includes the following topics:

- ▶ Overview
- ▶ Application layer
- ▶ Orchestration layer
- ▶ Data layer
- ▶ General recommendations
- ▶ Tuning
- ▶ Monitoring

2.1 Overview

This section reiterates the capabilities of the compute services and maps them to solution elements. It also documents the respective software levels that were current when this paper was written.

2.1.1 Capabilities and solution elements

To enable the analysis of genomics data, the compute services provide the following capabilities:

- ▶ A user GUI for physician and data scientist to submit and manage batch jobs and to create and manage custom workflows
- ▶ A Workloads Management GUI for the IT administrator to view cluster status and use
- ▶ Secure, high-speed access to files stored on storage cluster
- ▶ Scaling via a Workload Scheduler that enables high-throughput execution of batch jobs
- ▶ Performance and tuning recommendations that support the [preferred practices](#) of the Broad Institute GATK.
- ▶ Support of IBM Power or x86-64 nodes for batch processing and for interactive login to access the resources

Table 2-1 shows the mapping of solution elements to capabilities. The capabilities are written in a product neutral language to emphasize user requirements. The mapping of capabilities to solution elements shows how each selected solution element supports at least one solution capability. This method assures that the selected solution elements are kept at the absolute minimum required to support the capabilities and that over engineering is avoided.

Scheduler note: The IBM Spectrum Scale Blueprint for Genomics Medicine Workloads blueprint uses IBM Spectrum LSF as the workload scheduler; however, most of the recommendations are generic and apply to other schedulers.

Table 2-1 Mapping of solution elements to the compute services capabilities

Compute services: Solution elements	
Capability	Provided by
User GUI for physicians and data scientists to submit and manage batch jobs	IBM Spectrum LSF Application Center
User GUI for physicians and data scientists to create and manage custom workflows	IBM Spectrum LSF Process manager
Workload management GUI for IT administrators to view cluster status and use	IBM Spectrum LSF Application Center
A workload scheduler for high-throughput execution of batch jobs	IBM Spectrum LSF
Tuning recommendations from the Broad Institute GATK guidelines	<i>IBM Spectrum Scale Best Practices Guide for Genomics Medicine Workloads</i> (this Redpaper)
Support of Power or x86-64 as user nodes and for batch processing	IBM Spectrum LSF

Compute services: Solution elements	
Capability	Provided by
User nodes for physicians or data scientists to log on and access the resources	IBM Spectrum LSF
Secure high-speed internal access to files stored on the storage cluster	IBM Spectrum Scale Remote Cluster Mount

2.1.2 Software levels

The following software levels were current when this paper was written:

- ▶ IBM Spectrum Scale 4.2.3.5
- ▶ IBM Spectrum LSF 10.1.0.3
- ▶ Red Hat Enterprise Linux (RHEL) 7.3 Little Endian

2.2 Application layer

The ecosystem of bioinformatics comprises hundreds of applications. However, the Broad Institute GATK is the predominant application to analyze data that is acquired from genome sequencers. The IBM Spectrum Scale Blueprint for Genomics Medicine Workloads blueprint is optimized for the Broad Institute GATK [best practices](#), although most of the recommendations are generic and apply to other workloads.

2.2.1 The Broad Institute GATK

The Broad Institute GATK is a set of applications that is used to create multi-step workflows for variant discovery analysis of both germline and somatic genomes. Each step has its own set of tools. The output from each step is the input to the next step. There are a variety of GATK-based workflows that are used in the field. For this paper, the examples profile the workflow that is documented in the Broad Institute GATK [best practices](#) (as shown in Figure 2-2).

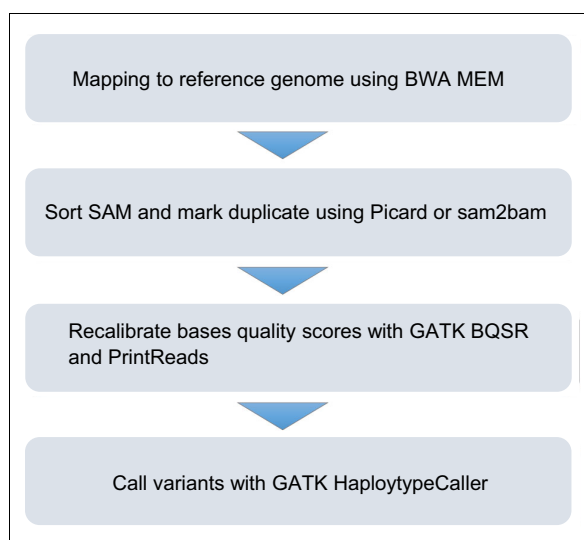


Figure 2-2 GATK based workflows

To profile that workflow, the following environment was used:

- ▶ A single Power8 node (IBM 8247-22L with SMT=8) with 256 GB of memory to execute the whole workflow
- ▶ 1x IBM Elastic Storage Server (ESS) GS4 with SSDs (≥ 23 GBps write bandwidth and ≥ 30 GBps read bandwidth)
- ▶ Dual rail FDR InfiniBand aggregating to ~13 GBps
- ▶ GATK pipeline execution using Whole Genome Sequence (WGS) input dataset with B37 reference data set

We set SMT=8 on the Compute Nodes based on “Table 3” in “Performance optimization of Broad Institute GATK Best Practices on IBM reference architecture for healthcare and life sciences”. Furthermore, BWA-Mem required SMT=8 to launch 160 threads on a node.

Note: This profiling setup is different than the example configuration. However, the insights gained by the profiling influenced the example configuration.

Figure 2-2 on page 13 is a four-step workflow that requires executing six applications. The profiling was performed with the Solexa WGS data set provided by the Broad Institute. Figure 2-3 shows also the execution time achieved on the profiling environment. Table 2-2 on page 15 summarizes the workload profile of each processing step and shows the run time achieved on the profiling environment. The actual throughput or performance that any user will experience can vary, depending upon many factors. See Appendix A for profiling details.

	Solexa WGS Broad dataset with b37 reference
BWA-Mem	303 min 47 sec
sam2bam (storage mode)	35 min 53 sec
GATK BaseRecalibrator (java setting -Xmn10g -Xms10g -Xmx10g)	87 min 21 sec
GATK PrintReads (java setting -Xmn10g -Xms10g -Xmx10g)	97 min 1 sec
GATK HaplotypeCaller (java setting -Xmn10g -Xms10g -Xmx10g)	261 min 37 sec
GATK mergeVCF (java setting -Xmn10g -Xms10g -Xmx10g)	0 min 51 sec
➔ Execution time was measured on the profiling environment using the Solexa WGS Broad dataset. The actual throughput or performance that any user will experience will vary depending	

Figure 2-3 Execution time on the profiling environment using the Solexa WGS Broad data set

Table 2-2 Workload profile of each processing step

	BWA-Mem	sam2bam (storage mode)	GATK BaseRecalibrator	GATK PrintReads	GATK HaplotypeCaller	GATK mergeVCF
Run Time ^{a b}	303 min 47 sec	35 min 53 sec	87 min 21 sec	97 min 1 sec	261 min 37 sec	0 min 51 sec
CPU	Intensive. Close to 100% CPU utilization	~93% (initial phase) and ~40% in later phases	~70% CPU utilization	~70% CPU utilization	~40% CPU utilization	Less than 1% CPU utilization
Memory	Low memory consumption	Low memory consumption	Total of 18 x Java threads with each thread customized with 10 GB → 180 GB	Total of 18 x Java threads with each thread customized with 10 GB → 180 GB	Not memory intensive	Not memory intensive
File data I/O access pattern	Pattern of writes followed by reads, Predominantly sequential I/O.	Write I/O predominantly sequential I/O. Read I/O is random access in units of 512 KiB.	Predominantly read intensive. Read is mix of sequential and random I/O.	Mix of read and write. Write I/O is mostly 512 KiB with mix of sequential and random. Read is mostly sequential.	Mix of read and write. Write I/O is mix of sequential and random. Read is mostly sequential.	Mix of read and write. Read and write I/O is predominantly sequential I/O.
File I/O bandwidth	<= 200 MBps (read and write)	Write < 2.5 GBps. Sustained read < 300 MBps. High degree of pagepool cache hits during reads (< 36 GBps).	<= 100 MBps (read and write)	Write < 150 MBps and read < 75 MBps	Write < 100 MBps and read < 100 MBps	Write < 1.5 GBps and read < 2 GBps
File Metadata	<=2 inode updates	Initial phase <= 60 inode updates. Later phase, <=2 inode updates.	~24 file open and ~24 file closes.	~24 file open and ~24 file closes.	~20 file open and ~20 file closes.	~2 file open and ~2 file closes.
Output file or files	Single output file (*.sam) <= 380 GB file size	Two output files: ~77 GB (.bam) and ~9 MB (.bam.bai).	Total of 52 files: 26 x *.table.log-4" files (<200 KB) and 26 x *.table" files (< 300 KB)	Total of 78 files: 26 x ".recal_reads*.bam" files (< 15 GB), 26 x *.bai" files (< 750 KB), and 26 x *.recal_reads*.bam.log" files (< 200 KB)	Total of 78 files: 26 x *.raw_variants*.vcf" files (< 6 GB), 26 x *.raw_variants*.vcf.log" files (< 400 KB), and 26 x *.raw_variants*.vcf.idx" files (< 20 KB)	Single output file (*.raw_variants.vcf) with ~66 GiB file size

a. Execution time was measured on the profiling environment using the Solexa WGS Broad dataset. The actual throughput or performance that any user will experience will vary depending upon many factors.

b. java setting -Xmn10g -Xms10g -Xmx10g

The analysis of the GATK workflow of the Solexa WGS Broad data set guided the details of the design of the compute cluster and the /gpf/s/data file system that is served by the storage services. In summary, BWA-Mem is CPU intensive. For optimal performance, run this application on nodes with higher core count and higher clock frequency. The sam2bam is memory intensive. It can be executed in one of two possible modes: memory mode and storage mode.

In our test, sam2bam in storage mode took 35 minutes and 53 seconds to complete. For optimal performance, execute sam2bam in memory mode on node with >= 1 TiB of memory. The GATK Base Recalibrator and PrintRead steps are memory intensive. To achieve optimal performance, execute these application on nodes with >= 512 GiB of memory.

More details about the file systems are provided in Chapter 3. An initial mention of them is provided here because the GATK performance profile influenced the choices. Place the file system metadata and data on separate storage pools. Configure the data storage pool with a larger IBM Spectrum Scale File System block size (8 MiB). Because the IBM Spectrum Scale File Systems are remotely mounted on the compute cluster from the storage services, the IBM Spectrum Scale networking should be over a low-latency and high throughput network interface.

2.3 Orchestration layer

The orchestration layer provides a workload scheduler to enable high-throughput execution of genomics analysis jobs. In this blueprint, it is accomplished with IBM Spectrum LSF, although most of the recommendations are generic and apply to other schedulers.

2.3.1 IBM Spectrum LSF

IBM Spectrum LSF is a scalable, comprehensive workload management tool. To facilitate ease of use, the following IBM Spectrum LSF add-ons are chosen:

- ▶ IBM Spectrum LSF Application Center is a rich environment for building easy-to-use application-centric web interfaces, simplifying job submission, management and remote visualization. The web-based interface is used to remotely monitor jobs, access job-related data and perform basic operations.
- ▶ IBM Spectrum LSF Process Manager is a powerful interface for designing complex engineering computational processes and capturing repeatable best practices that can be leveraged by other users. When integrated with IBM Spectrum LSF Application Center, a consistent web-based environment is presented to the users.

For the purpose of simplicity and ease of use by physicians and data scientists, the IBM Spectrum LSF Process Manager provides a GUI for creating, submitting, monitoring and modifying workflow templates (also known as a *flow* in IBM Spectrum LSF vocabulary). Figure 2-4 illustrates an example flow.

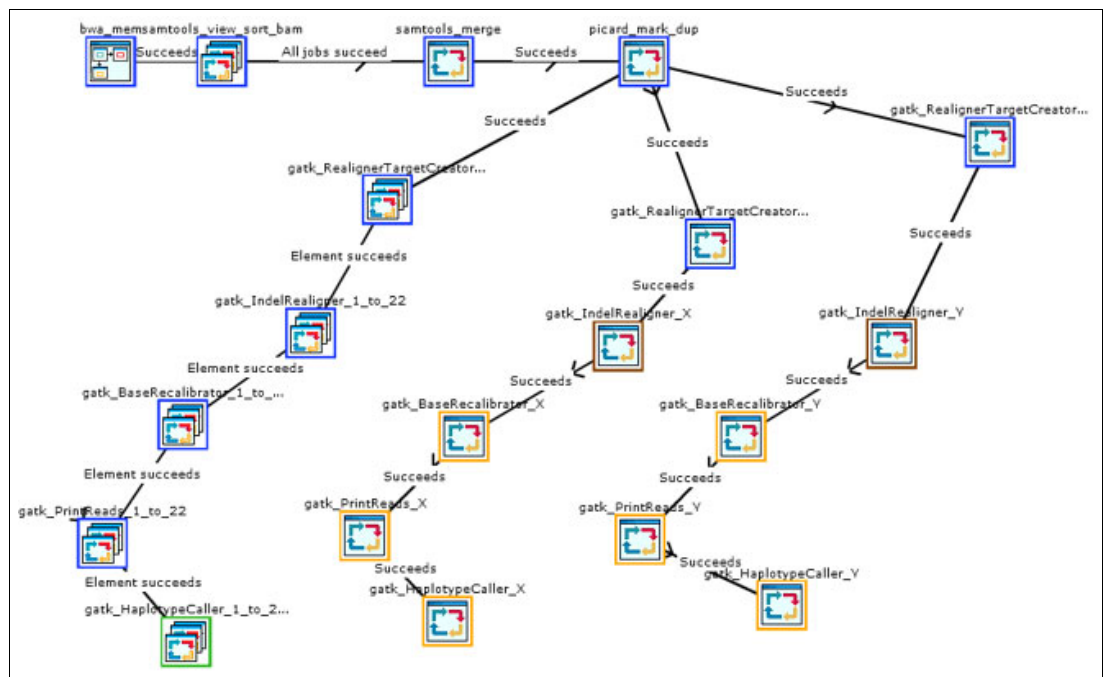


Figure 2-4 Example workflow created with IBM Spectrum LSF Process Manager

A single job queue is used. Using a single default job queue (normal) keeps the configuration easy and simple. Different possible host capabilities (for example Power versus X86-64, memory size, GPU available) are specified in a configuration file (lsb.hosts). It is the flow template that specifies the required resource requests. This configuration removes the complexity of workflow optimization from the user (for example, the physician) to the creator of the flow template (for example, the IT administrator).

Depending on the customer requirements, additional job queues can be configured, for example, a *high priority queue* or an *admin queue*.

The workload scheduler requires a shared file system (for example, NFS, IBM Spectrum Scale) to store binary files, configuration files, and log files. The shared file system can become a performance bottleneck for compute clusters with a large number of nodes and short running batch jobs. For compute clusters with up to 1,000 compute nodes that run genomics workload, store all IBM Spectrum LSF files in a IBM Spectrum Scale File System. Storing those files in IBM Spectrum Scale eliminates the need for an external NFS service. Avoiding a reliance on an NFS file service simplifies the configuration and cost.

The genomics data and the LSF application binaries must be stored in separate file systems. This configuration allows the storage services to customize the file systems (for example, blocksize, replication) and to isolate the respective I/O loads from impacting each other. Writing from multiple hosts into the same IBM Spectrum Scale directory triggers IBM Spectrum Scale Token Traffic to keep the directory structure consistent. This method impacts performance. The orchestration layer, requires a dedicated sub directory per IBM Spectrum LSF execution host (compute node) to eliminate this bottleneck. The LSF directory structure in the LSF file system is illustrated in Figure 2-5.

Purpose	Variable Name	Variable Value	Used by	Comment
Full path to the top level LSF installation directory	LSF_TOP	/gpfs/app/lfsf	All nodes	-
Directory in which the job history and accounting logs are kept for each cluster	LSB_SHAREDIR	/gpfs/app/lfsf/work	Master host	-
Defines the LSF system log file directory (*)	LSF_LOGDIR	/gpfs/app/lfsf/log/%H	All nodes	Dedicated sub directory per host
Specifies the directory for buffering batch standard output and standard error for a job (*)	JOB_SPOOL_DIR	/gpfs/app/lfsf/log/%	Execution hosts	Dedicated sub directory per host
Cluster-wide current working directory (CWD) for the job	DEFAULT_JOB_CWD	/gpfs/app/lfsf/cwd/%H	Execution hosts	Dedicated sub directory per host
Specifies the path and directory for temporary LSF internal files (*)	LSF_TMPDIR	/gpfs/app/lfsf/tmp/%H	Execution hosts	Dedicated sub directory per host
(*) IBM Spectrum Scale independent filesets will be configured for /gpfs/app/lfsf/log, /gpfs/app/lfsf/spool, and /gpfs/app/lfsf/tmp.				
➔ Having dedicated sub directories per compute node eliminates potential IBM Spectrum Scale Token Traffic.				
➔ IBM Spectrum Scale independent filesets enable automated data management on the Storage Cluster.				

Figure 2-5 The LSF directory structure for IBM Spectrum LSF

IBM Spectrum Scale independent filesets are created for some of the directories. Independent filesets enable automated data management using IBM Spectrum Scale Policies. The configuration of data management policies is outside the scope of this blueprint but will be added in a future update. It is difficult to create independent filesets later, if the respective directories already contain files. So they are created from the beginning to enable the later configuration of data management policies.

Even though IBM Spectrum LSF was chosen, the recommendations discussed previously are generic to a workload scheduler and can apply to other schedulers.

2.4 Data layer

The data layer of the compute services is provided by, primarily, two IBM Spectrum Scale file systems from the storage cluster that are mounted on the compute cluster via an IBM Spectrum Scale multi-cluster remote cluster mount. There is an optional third file system mounted on the compute cluster, making the following file systems available:

/gpfs/data	For genomics data and analysis results
/gpfs/app	For application binaries, configuration files and log files
/gpfs/user	For user data for execution of batch jobs (optional)

These file systems are configured and managed on the storage cluster. IBM Spectrum Scale provides secure and high-speed access to files that are stored and managed on the storage cluster.

Details about these file systems are provided in Chapter 3. For now, it is sufficient to know that the compute cluster nodes are configured as IBM Spectrum Scale nodes. They should set `autoload=yes`, so that they can remotely mount the file systems on boot:

```
mmchconfig autoload=yes -N <compute_node_class>
```

All nodes need to be connected via a high-speed, low-latency network. The private network services are described in Chapter 4.

2.5 General recommendations

Preferred practices increase the operational efficiency for managing the entire compute infrastructure. This section provides the following general recommendations:

- ▶ Designation of the compute nodes
- ▶ IBM Spectrum Scale node roles
- ▶ IBM Spectrum LSF host types
- ▶ IBM Spectrum LSF add-ons
- ▶ External dependencies
- ▶ Communication and security aspects

2.5.1 Designation of the compute nodes

All nodes of the compute cluster are configured as IBM Spectrum Scale nodes and as IBM Spectrum LSF nodes. There are several possible node designations that each compute node can belong to. Figure 2-6 summarizes the node designation of the example configuration that was introduced in Section 1.4.

	Compute Node Type	Memory	End User Login	IBM Spectrum Scale Node	IBM Spectrum Scale Quorum	IBM Spectrum Scale Manager	IBM Spectrum Scale Admin	IBM Spectrum Scale GUI	IBM Spectrum LSF Node Type	IBM Spectrum LSF AC	IBM Spectrum LSF PM
Power 1	Management (Primary)	256GB	No	X	X	X	X	X	Master	(X)	(X)
Power 2	Management (Standby)	256GB	No	X	X	X	X	X	Master (Stand-by)	X	X
Power 3	User Login	256GB	Yes	X	X				Submission		
Power 4	Worker	256GB	No	X					Execution		
Power 5	Worker	256GB	No	X					Execution		
Power 6	Worker	256GB	No	X					Execution		
Power 7	Worker	512GB	No	X					Execution		
Power 8	Worker	512GB	No	X					Execution		
Power 9	Worker	1024GB	No	X					Execution		
Power 10	Worker	1024GB	No	X					Execution		
Intel 1	Worker	256GB	No	X					Execution		
Intel 2	Worker	256GB	No	X					Execution		

Figure 2-6 Designation of the compute nodes

Most of the compute nodes of the example configuration are Power servers, because Power servers were easier to obtain for the example environment. The Power server are equipped with different memory configurations, because we wanted to analyze what workflow steps benefit from more memory.

In addition, two x86-64 servers are added for the following reasons:

- We want to show that IBM Spectrum Scale and IBM Spectrum LSF support mixed configurations comprising Power and x86-64 server.
- We plan to profile genomics applications on x86-64, as we did for GATK on Power (described in Section 2.2.1).

Independent of the CPU type, there are three types of compute nodes:

- *Management nodes* run all the services to dispatch and manage batch jobs:
 - IBM Spectrum LSF, to dispatch and monitor batch jobs
 - IBM Spectrum LSF Application Center GUI, to submit and manage batch jobs, and to view cluster status and utilization
 - IBM Spectrum LSF Process Manager GUI, to create and manage custom workflows

The login is restricted to administrative users only. Management nodes are the most stable nodes and, therefore, are good candidates to run additional infrastructure services. These services are critical for the entire compute cluster. Therefore, user login is not allowed on these nodes to provide additional protection.

- *User login nodes* allow, as the name says, users to log in to these nodes to compile applications and to submit jobs and flows via a command line interface (CLI). User login nodes are stable nodes and, therefore, are also reasonable candidates to run additional infrastructure services.

- *Worker nodes* execute batch jobs that are dispatched by the scheduler, in our example environment IBM Spectrum LSF. The login is restricted to administrative users only. These nodes are less stable because users might dispatch new experimental applications.

2.5.2 IBM Spectrum Scale node roles

In an IBM Spectrum Scale cluster, some nodes have to assume additional roles to maintain the status of the IBM Spectrum Scale cluster. Because all nodes of the compute cluster are also IBM Spectrum Scale nodes, we need to decide which compute nodes assume the additional roles of IBM Spectrum Scale *quorum node* and IBM Spectrum Scale *manager node*.

For the IBM Spectrum Scale quorum nodes, the general recommendation is to define three or five quorum nodes. Losing quorum due to a multiple quorum node outage, impacts the cluster's ability to mount and serve an IBM Spectrum Scale file system. This ability is, by design, necessary to preserve file system integrity. Any node that is chosen to be a quorum node must minimize the possibility of an outage. Consider avoiding single points of failure by assigning the quorum nodes to different power circuits, different racks, and different network switches. In IBM Spectrum Scale, every quorum node is automatically a configuration server node.

Another important role is IBM Spectrum Scale manager. Nodes that are assigned this role provide the cluster manager, file system manager, and token manager services. IBM Spectrum Scale assigns these services automatically to the configured IBM Spectrum Scale manager nodes.

IBM Spectrum Scale *admin nodes* issue all IBM Spectrum Scale administrative commands. By design, IBM Spectrum Scale commands maintain the appropriate environment across all nodes in the cluster. The admin nodes have similar requirements as the IBM Spectrum Scale management nodes: password-less root ssh and scp to all other IBM Spectrum Scale nodes, access restricted to administrative users only. For redundancy, it is best, if possible to have at least two IBM Spectrum Scale nodes that are admin nodes.

The IBM Spectrum Scale *GUI nodes* are always admin nodes. The GUI does not allow root login. Only an admin login exists. The GUI subsystem passes commands as root to the other IBM Spectrum Scale nodes of the cluster. Most, but not all, IBM Spectrum Scale functions can be run from the GUI, so occasionally, some commands require root login for CLI access. All GUI nodes run a performance monitoring collection daemon that is used by the GUI to report cluster health and performance.

In the example environment for this paper, the two compute cluster management nodes and the user login node are the most stable nodes. It is required to configure at least three IBM Spectrum Scale quorum nodes. Therefore, we configure each of these three nodes as IBM Spectrum Scale quorum nodes. In general, compute cluster management nodes are more stable than compute cluster user login nodes. Therefore, we configure the compute cluster management nodes as IBM Spectrum Scale manager nodes, but not the compute cluster user login node.

2.5.3 IBM Spectrum LSF host types

All compute cluster nodes are also IBM Spectrum LSF *hosts*. IBM Spectrum LSF refers to nodes as *hosts* and distinguishes the following host types:

- ▶ *Master host*: Acts as the overall coordinator for the cluster, doing all job scheduling and dispatch
- ▶ *Server host*: Submits and runs jobs
- ▶ *Client host*: Submits jobs and tasks only
- ▶ *Execution host*: Runs jobs and tasks
- ▶ *Submission host*: A host from which jobs and tasks are submitted

To keep the configuration simple, the example for this paper uses only master, submission, and execution hosts.

The purpose of the IBM Spectrum LSF master host is to coordinate IBM Spectrum LSF job scheduling and dispatching within the cluster. For redundancy purposes, configure more than one IBM Spectrum LSF master host. Only one LSF master host is active at any given time. IBM Spectrum LSF has built-in failover, in case the current IBM Spectrum LSF master host node fails. The example environment configures the first compute cluster management node as IBM Spectrum LSF master host and the second compute cluster management node as LSF master host candidate.

The IBM Spectrum LSF submission host allows users to submit batch jobs and flows via the IBM Spectrum LSF CLI. Already submitted or dispatched jobs continue to be executed when a submission host fails. IBM Spectrum LSF allows you to configure multiple submission hosts. The example configuration configures one submission host only, because it is short on compute nodes and having more execution hosts is more important for our testing purposes.

The IBM Spectrum LSF execution hosts are the workforce of the compute cluster. Execution hosts simply run jobs and tasks. Large compute clusters scale up to thousands of execution hosts. In some environments for genomics medicine workloads, we observed deployments with up to a few hundred compute nodes. In the example environment for this paper, we configured all remaining compute nodes as execution hosts.

2.5.4 IBM Spectrum LSF add-ons

The IBM Spectrum Scale Blueprint for Genomics Medicine Workloads chooses two IBM Spectrum LSF add-ons to provide GUIs for users and administrators. IBM Spectrum LSF Application Center (formerly known as *IBM Platform Application Center* or PAC) provides a WebUI for jobs submission, job monitoring, and basic IBM Spectrum LSF cluster management. IBM Spectrum LSF Process Manager (formerly known as *IBM Platform Process Manger* or PPM) provides a WebUI for flow creation and flow management.

In the example environment for this paper, the second compute cluster management node is configured as an IBM Spectrum LSF Application Center server and as an IBM Spectrum LSF Process Manager server. The first compute cluster management node acts as a stand-by server for both add-ons. All binaries and configuration files need to be installed on both servers. The standby server needs to be started manually if the active server fails. The scripting for the automatic start of the stand-by server is outside the scope of the blueprint.

2.5.5 External dependencies

There are several external dependencies that are essential for the compute services. IBM Spectrum Scale depends on a highly available name resolution service (typically DNS) for name resolution and reverse name resolution. IBM Spectrum Scale also depends on time services (typically NTP) for time synchronization. Certain user and administrative commands depend on proper ID mapping. The ID mapping can be configured either locally on each node or, preferably, on a central ID mapping service (such as LDAP or Microsoft Active Directory).

Therefore, each compute node needs to be connected to a customer provided DNS server, a customer provided NTP service, and a customer provided ID mapping service. In most cases, the customer already has such a service. Otherwise, you need to configure these services.

It is a preferred practice that each compute node contacts these customer-provided infrastructure services via the compute cluster management nodes. The compute cluster management nodes can run an instance of each service and connect it to the respective customer provided service. Alternatively, the management nodes can forward the respective network traffic on the network layer to the external services. This blueprint supports both of these approaches.

For deployment and management, customers typically have an infrastructure to install and manage servers to automatically install and configure the operating system and to automatically monitor and report hardware failures. There is a broad variety of tools available and used by customers. In most cases the customer already has such a service. Otherwise, such a service must be configured.

IBM Spectrum LSF requires that you configure the `lsfadmin` user on all LSF hosts. Because IBM Spectrum LSF spawns dispatched jobs with the UID and GID of the user who submitted the job, you also need to configure all LSF execution hosts with the user and group information of all LSF users. All user and group information for IBM Spectrum LSF, SMB, and NFS must be consistent to enable batch jobs to access and analyze data that was ingested via the NFS or SMB service that is provided by the storage services.

Recommendation: Store the user and group information for all compute nodes in an external authentication and ID mapping service, such as an LDAP or Microsoft Active Directory. In most cases, you will already have such a service. Otherwise, you might need to configure this type of service.

2.5.6 Communication and security aspects

All nodes of the compute cluster must be able to communicate with each other and, likewise, must be able to communicate to all nodes of the storage cluster. All nodes used for administering IBM Spectrum Scale must be able to do root password-less root `ssh` and `scp` into any other node of the compute cluster. IBM Spectrum Scale sudo wrappers are not used in this blueprint.

2.6 Tuning

This blueprint is based on IBM ESS 5.2 and IBM Spectrum Scale 4.2.3.4 on the ESS I/O nodes and IBM Spectrum Scale 4.2.3.4 (or higher) on the Cluster Export Services (CES) protocol nodes and compute nodes. The IBM Spectrum Scale Blueprint for Genomics Medicine Workloads blueprint is optimized for the Broad Institute GATK [best practices](#), IBM Elastic Storage Server (ESS) as back-end storage, and InfiniBand networking. However, most settings are generic and apply to other workloads. This section provides tuning recommendations.

2.6.1 Operating system

Apply the settings described in this section on each compute node for genomics workload.

Add the following lines to the `/etc/security/limits.conf` file:

```
* soft memlock unlimited
* hard memlock unlimited
* soft nofile 16384
* hard nofile 16384
```

Set the tuned profile to throughput-performance by adding the following line in the `/etc/tuned/active_profile` file:

```
throughput-performance
```

Add the following lines to the `/usr/lib/tuned/throughput-performance/tuned.conf` file:

```
[cpu]
governor=performance
energy_perf_bias=performance
min_perf_pct=100
```

2.6.2 Network

Apply the settings described in this section on each compute node for genomics workload.

This blueprint uses InfiniBand for the high-speed data transfer network. Therefore, each compute node is configured with at least one Mellanox EDR or FDR InfiniBand Host Channel Adapter. Apply the [tuning parameters](#) recommended by Mellanox.

In addition, configure the following settings in the `/etc/sysctl.conf` file:

```
net.ipv4.tcp_timestamps=0
net.ipv4.tcp_sack=0
net.core.netdev_max_backlog=250000
net.core.rmem_max=16777216
net.core.wmem_max=16777216
net.core.rmem_default=16777216
net.core.wmem_default=16777216
net.core.optmem_max=16777216
net.ipv4.tcp_rmem=4096 87380 16777216
net.ipv4.tcp_wmem=4096 65536 16777216
net.ipv4.tcp_low_latency=1
net.ipv4.tcp_adv_win_scale=2
net.ipv4.tcp_window_scaling=1
```

```
net.core.somaxconn = 8192
vm.min_free_kbytes = 512000
kernel.sysrq = 1
kernel.shmmax = 137438953472
```

In general, time stamps and TCP Selective Acknowledge (SACK) are good tools to handle and diagnose network issues. However, feedback from the field suggests to disable time stamps and SACK. Users with notebooks and workstations with old kernels might have faulty SACK implementations. Enabling SACK for connections to those clients can make problems worse.

2.6.3 IBM Spectrum Scale

Apply the settings described in this section on each compute node to optimize it for IBM ESS and Broad Institute GATK3. Details about ESS are provided in Chapter 4. For now, it is sufficient to know that the storage cluster includes ESS.

Because the storage backend is ESS, the `gssClientConfig.sh` script needs to be executed. The script is located on the ESS Management Server (EMS) in the `/usr/lpp/mmfs/samples/gss` directory). Configure each compute node with 16 GiB page pool for IBM Spectrum Scale, so that the syntax of the command looks as follows:

```
gssClientConfig.sh -P 16384 <compute_node_class>
```

Apply the following settings in IBM Spectrum Scale, if InfiniBand is used for the high-speed data network:

```
mmchconfig maxFilesToCache=32K -N <compute_node_class>
mmchconfig maxMBps=20000 -N <compute_node_class>
mmchconfig socketMaxListenConnections=8192 -N <compute_node_class>
mmchconfig envVar="MLX4_USE_MUTEX=1 MLX5_SHUT_UP_BF=1 MLX5_USE_MUTEX=1" -N <compute_node_class>
```

Certain genomics application benefit from the caching the entire file. For example, the average file size for `bc12fastq` is 3-7 MB. Setting the IBM Spectrum Scale file-caching thresholds to 8 MB improves the application performance.

```
mmchconfig seqDiscardThreshold=8M -N <compute_node_class>
mmchconfig writebehindThreshold=8M -N <compute_node_class>
```

After applying all changes a snippet of `mm1sconfig` looks like:

```
[compute]
pagepool 16384M
numaMemoryInterleave yes
maxFilesToCache 32k
maxStatCache 0
maxMBps 20000
workerThreads 1024
ioHistorySize 4k
verbsRdma enable
verbsRdmaSend yes
verbsRdmasPerConnection 256
verbsSendBufferMemoryMB 1024

ignorePrefetchLUNCount yes
scatterBufferSize 256k
nsdClientCksumTypeLocal ck64
nsdClientCksumTypeRemote ck64
```

```

socketMaxListenConnections 8192
envVar MLX4_USE_MUTEX=1 MLX5_SHUT_UP_BF=1 MLX5_USE_MUTEX=1
verbsPorts <active_verbs_ports>
seqDiscardThreshold 8M
writebehindThreshold 8M

```

```

[common]
cipherList AUTHONLY
adminMode central

```

Settings in **bold** font are changed for tuning. All other lines are the IBM Spectrum Scale default settings.

2.7 Monitoring

For monitoring we lean on the built-in tools provided by IBM Spectrum Scale and IBM Spectrum LSF Application Center. Figure 2-7 depicts the user view for submitting and manage batch jobs and Figure 2-8 depicts the administrator view to monitor cluster status and utilization. The monitoring of IBM Spectrum Scale is discussed in context of the storage services (Section 3.8).

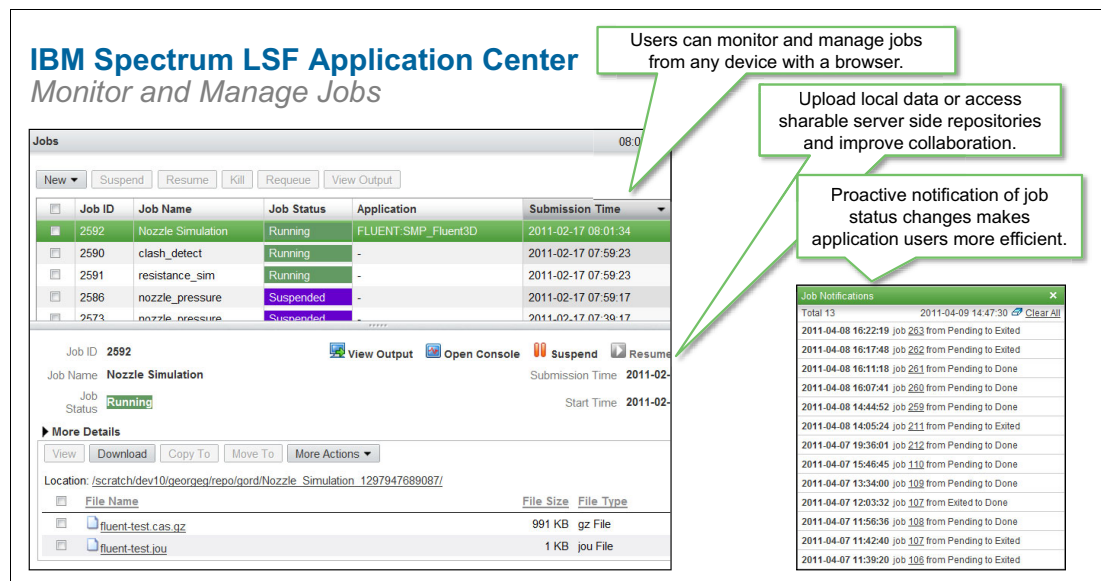
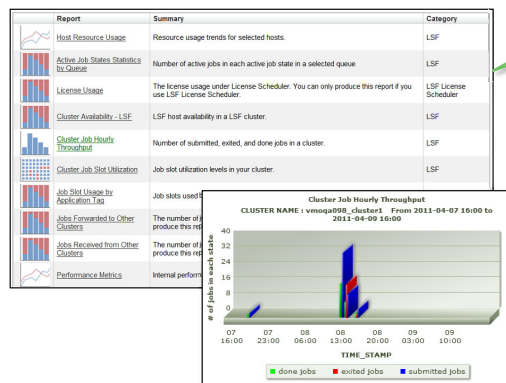


Figure 2-7 IBM Spectrum LSF Application Center to monitor and manage batch jobs

IBM Spectrum LSF Application Center Integrated Reporting



Extensive library of built-in, relevant reports related to resource usage and jobs.

Access reporting and analysis functions directly through IBM Spectrum LSF Application Center.

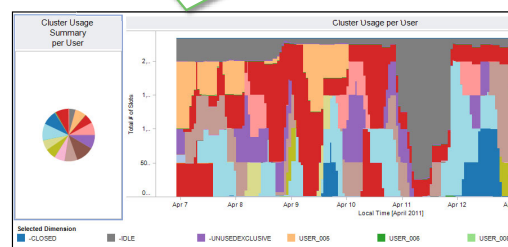


Figure 2-8 IBM Spectrum LSF Application Center to monitor cluster status and use

The storage services

The storage services provide rapid ingestion of genomics data that is acquired from devices, such as genome sequencers and cryo-electron microscopes, access of genomics data by data scientists and physicians, sharing of genomics data across sites and institutions, provisioning of genomics data to the compute services for analysis, and reliable and cost-effective storing and management of genomics data.

The design of this piece of the composable infrastructure provides scalable, high-performance storage that can be easily used and managed. The storage services design is based on seven expertly engineered building blocks that enable IT architects to compose solutions that meet customers varying performance and functional needs (Figure 3-1). This chapter discuss each of these building blocks.

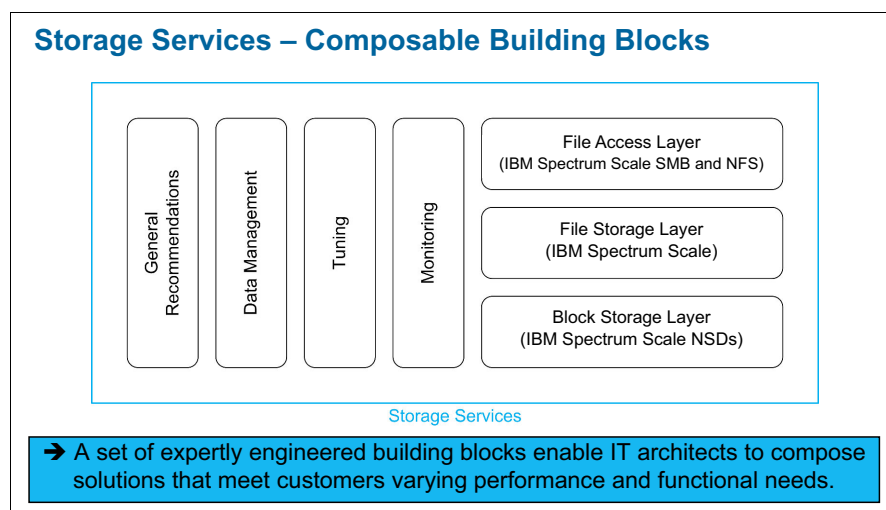


Figure 3-1 The seven expertly engineered building blocks of the storage services

This chapter includes the following topics:

- ▶ Overview
- ▶ File storage layer
- ▶ Block storage layer
- ▶ File access layer
- ▶ General recommendations
- ▶ Data management
- ▶ Tuning
- ▶ Monitoring

3.1 Overview

This section reiterates the capabilities of the storage services and maps them to solution elements. It also documents the respective software levels that were current when this paper was written.

3.1.1 Capabilities and solution elements

To enable access to genomics data the storage services provides the following capabilities:

- ▶ *Data transfer nodes* for secure high-speed external access via NFS and SMB to ingest data from genomics sequencers, microscopes, and so on for access by data scientists and physicians and for sharing across sites and institutions
- ▶ *Secure high-speed internal access* for analysis on the compute cluster

To effectively store and manage genomics data the storage services provide the following components:

- ▶ *Scale-out architecture* that is capable to store from a few 100 TB to tens of PB of genomics data
- ▶ *End-to-end checksum* to ensure the data integrity all the way from the application to the disks
- ▶ *Data management GUI* to configure and monitor storage resources
- ▶ *Optional professional services* ranging from management of daily operation to consultancy for major configuration changes

Table 3-1 on page 29 shows the mapping of solution elements to capabilities. Like for the compute services, the capabilities are written in a product neutral language to emphasize user requirements. The mapping of capabilities to solution elements shows how each selected solution element supports at least one solution capability. As explained earlier, this method assures that the selected solution elements are kept at the absolute minimum that is required to support the capabilities and that over engineering is avoided.

Storage device: The IBM Spectrum Scale Blueprint for Genomics Medicine Workloads blueprint uses IBM Elastic Storage Server (ESS) as storage device. IBM Spectrum Scale supports a broad variety of storage devices, but different tuning settings are required to optimize the infrastructure end to end.

Table 3-1 Mapping of solution elements to the storage services capabilities

Storage services: Solution elements	
Capability	Provided by
Scale-out architecture that is capable to store genomics data from a few 100 TB to tens of PB of file data	IBM Spectrum Scale
Data transfer nodes for secure high-speed external access via NFS and SMB to ingest data, user access, and sharing	IBM Spectrum Scale Cluster Export Services (CES)
Secure high-speed internal access for analysis on the compute cluster	IBM Spectrum Scale Remote Cluster Mount
End-to-end checksum to ensure the data integrity all the way from the application to the disks	IBM Elastic Storage Server (ESS)
Data Management GUI to configure and monitor storage resources	IBM Spectrum Scale GUI
Optional professional services ranging from management of daily operation to consultancy for major configuration changes	IBM Lab Based Services

3.1.2 Software levels

The following software levels were current when this paper was written:

- ▶ IBM ESS 5.2.0 (includes IBM Spectrum Scale 4.2.3.4)
- ▶ IBM Spectrum Scale 4.2.3.4 also on CES nodes
- ▶ Red Hat Enterprise Linux (RHEL) 7.3 Little Endian

3.2 File storage layer

At the heart of the storage services are the IBM Spectrum Scale *file systems*. The blueprint requires multiple file systems. Although multiple file systems can increase administrative overhead, the benefits outweigh the additional overhead. This section provides general preferred practices for the design of IBM Spectrum Scale file systems and then applies those practices to provide recommendations for genomics workload. It also introduces IBM Spectrum Scale filesets that are a great tool to efficiently manage large amounts of genomics data.

3.2.1 IBM Spectrum Scale file systems

There are many attributes of IBM Spectrum Scale file systems to consider when developing a solution for genomics. Multiple file systems increase administrative overhead. So, first consider how many IBM Spectrum Scale file systems you need to configure. Do you need more than one? IBM Spectrum Scale stripes data across all available resources. Multiple file systems might isolate resources, but for most workloads it is better if IBM Spectrum Scale can stripe across all available resources.

IBM Spectrum Scale has a nice feature to partition an IBM Spectrum Scale file system into multiple IBM Spectrum Scale filesets. IBM Spectrum Scale filesets are discussed later (Section 3.2.3). For now, it is sufficient to know that IBM Spectrum Scale filesets exist.

When designing IBM Spectrum Scale file systems, always think whether a new file system can be replaced by an IBM Spectrum Scale fileset of an already existing IBM Spectrum Scale file system.

You might have several reasons for creating more than just one file system:

- ▶ Each IBM Spectrum Scale file system can be configured with one data block size only. Based on the actual workload, it is required to configure the file system with the correct data block size to get optimal performance. Sometimes it is required to configure multiple file systems with different data block sizes to tune performance for different workloads.
- ▶ Multiple IBM Spectrum Scale file systems increase resiliency. Maintenance tasks, such as `mmfsck`, run longer on large file systems, which in turn implies longer downtime. If there is more than one file system, data ingest and analysis can continue on the remaining file systems.
- ▶ Quotas can have an impact to write workloads. With quota enabled, the clients and the quota manager must communicate to ensure that the quota is within the hard limits. This back-and-forth communication might degrade write performance with extreme workloads, for example when in large clusters many nodes write huge amounts of data to many disks at the same time.
- ▶ Snapshots can be created on the fileset level, although snapshots flush the data of the entire file system. Thus, each IBM Spectrum Scale node that mounts that file system needs to flush its buffer. The slowest node determines the execution time to complete a snapshot. The analysis workload running on the compute cluster highly use the compute nodes (as described in Section 2.2.1). This use can impact the execution time for snapshots, which implicitly impacts the execution time of all batch jobs that are accessing data in the same file system.
- ▶ File systems are the granularity for exposing data to remote IBM Spectrum Scale clusters using IBM Spectrum Scale multi-cluster remote cluster mount. This blueprint exports genomics and other data to the compute cluster.

Choosing the file system *block size* and *inode size* impacts space allocation and performance. IBM Spectrum Scale can store files smaller than ~3.5 KiB (without extended attributes) in an inode, if the inode size is configured to 4 KiB. For those files, the file system block size does not matter. All other files are stored in file system blocks or file system subblocks, the minimal allocatable space. IBM Spectrum Scale (up to 4.2.3) can divide a file system block in up to 32 subblocks of equal size.

If the file size is not a multiple of the subblock size, storage capacity is wasted. IBM Spectrum Scale 5.0 will have a different mechanism to allocate space. Therefore, the recommendation for file system block sizes will change with IBM Spectrum Scale 5.0.

IBM Elastic Storage Server (ESS) 5.2 offers overall better performance (considering client I/O, rebuilds, and so on) when configured with a file system block size of at least 4 MiB for data storage pools. For ESS-based deployments, 4 MiB block size is the best choice for small and mixed file workloads, while still giving a lot of sequential performance for larger files in the same file system. If you use ESS and know your exact workload (I/O pattern), a file system block size that matches the workload I/O size might offer better client I/O performance, as long as the block size is greater than or equal to 4 MiB.

IBM Spectrum Scale 4.2.3 supports the following methods to allocated space that are referred as *block allocation map*: cluster and scatter. The scatter method provides a more consistent file system performance on large clusters and large file systems by averaging out performance variations due to block location. It is also appropriate in most cases and is the default for IBM Spectrum Scale clusters with more than eight nodes or file systems with more than eight disks.

The *log file size* specifies the size of the internal log files. An increased log file size is useful for file systems that have a large amount of metadata activity, such as creating and deleting many small files or performing extensive block allocation and deallocation of large files, which is typical with genomics application I/O workloads.

Another important factor that impacts performance is the number of IBM Spectrum Scale nodes that mount a file system. Certain internal data structures of a IBM Spectrum Scale file system are optimized for the number of nodes where the file system are mounted. The number of nodes includes all nodes in the local IBM Spectrum Scale cluster and all nodes of all remote IBM Spectrum Scale clusters where the file system is mounted with multi-cluster remote cluster mount.

The `-n <number>` option of the `mmcrfs` command gives IBM Spectrum Scale a hint to optimize these data structures for the given number of nodes. The data structures are initialized when a file system is created. This value can be changed later, but a migration of files to a new storage pool is required to make the value effective. When creating a new file system, it is better to over estimate the number of nodes by a factor of two rather than making it too small.

For example, when you plan to create an IBM Spectrum Scale cluster with 80 nodes, use following command:

```
mmcrfs -n 128 ...
```

IBM Spectrum Scale supports the replication of data or metadata on the file system level. This replication is different from the replication in IBM Spectrum Scale RAID (discussed in Section 3.3.1). Replication of data or metadata on the file system level reduces the overall usable capacity but increases the data durability. Enabled replication protects against IBM Spectrum Scale NSD or underlying block storage errors. It is unlikely than an entire IBM Spectrum Scale NSD can get lost.

However, sometimes things happen that should not happen. The loss of an IBM Spectrum Scale NSD that stores metadata implies the loss of the entire file system. The restoration of a peta-scale file system can take a long time. Therefore, be sure to replicate at least the metadata to increase the resilience of the file system.

A final parameter to consider for the design of IBM Spectrum Scale file systems are access control lists (ACLs). IBM Spectrum Scale supports POSIX ACLs and NFSv4 ACLs. IBM Spectrum Scale CES requires you to configure the respective IBM Spectrum Scale file system with NFSv4 ACLs. Otherwise, you can decide which ACL type to use.

3.2.2 Recommendations for genomics medicine workloads

Based on general preferred practices for the design of IBM Spectrum Scale file systems, the blueprint requires the following separate file systems and one optional file system:

<code>/gpfs/data</code>	Genomics data and analysis results
<code>/gpfs/app</code>	Application binaries, configuration files, and log files
<code>/gpfs/user</code>	User data for execution of batch jobs (<i>optional</i>)
<code>/gpfs/ces</code>	Helper file system for CES to provide NFS and SMB

Acquired genomics data are ingested from genome sequencers and similar devices via NFS and SMB into the IBM Spectrum Scale `/gpfs/data` file system. The batch jobs that are running on the compute cluster to analyze genomics data pick up the data from there and write the results into the same file system. The application binaries, including the IBM Spectrum LSF binaries, are stored in the `/gpfs/app` file system to make them available on all compute nodes.

IBM Spectrum LSF is configured to write log files for the batch jobs into the same file system. The optional `/gpfs/user` file system can contain additional scripts and configuration files that users might need to run batch jobs. The `/gpfs/ces` file system is an internal file system that is used by IBM Spectrum Scale to store metadata for NFS and SMB. The `/gpfs/ces` file system is not accessed by users.

These file systems are optimized for the different workloads. Table 3-2 contrasts key settings and Figure 3-2 on page 33 through Figure 3-5 on page 34 provide details for each file system. Setting `relatime` on the `/gpfs/data` file system reduces metadata traffic and, thus, accelerates the analysis of genomics data. Separating the file system metadata and file data into different storage pools also enhances performance. See Section 3.3.1 for details about the storage pool configuration.

Note: IBM Spectrum Scale 5.0 will have a different mechanism to allocate space. Therefore, the advice for file system block sizes will change with IBM Spectrum Scale 5.0.

The `/gpfs/user` file system is optional. The execution of batch jobs on the compute cluster requires a shared home directory. This genomics blueprint suggests two possible methods to provide shared home directories. Use an existing shared home export from existing NFS server to mount it onto the compute cluster nodes to avoid data silos. Alternatively, create a separate IBM Spectrum Scale file system (`/gpfs/user`) on the storage cluster to avoid dependency to an external NFS service.

If you choose to store the shared home directories on IBM Spectrum Scale, store them in the separate `/gpfs/user` file system, because we observed that some genomics customers have a business policy for hourly snapshots of user data. In general, IBM Spectrum Scale snapshots quiesce the entire file system for a moment. Isolating the `/gpfs/user` file system from the `/gpfs/data` file system accelerates the batch jobs that analyze genomics data. Do not export the `/gpfs/user` file system via CES. NFS or SMB access can have an impact on performance of the running batch jobs.

You need to configure the `/gpfs/data` file system with NFSv4 ACLs, because it is exported via NFS and SMB. IBM Spectrum Scale CES requires that underlying IBM Spectrum Scale file systems are configured with NFSv4 ACLs. For reasons of simplicity of management of all IBM Spectrum Scale file systems, it is better to keep the ACL management consistent. Therefore, configure all four IBM Spectrum Scale file systems with NFSv4 ACLs.

Table 3-2 Key settings for the file systems

File system	Settings
<code>/gpfs /data</code>	<code>-j scatter</code> <code>-B 8 MiB</code> <code>-n <customer_specific></code> <code>--metadata-block-size 1 MiB</code> <code>-L 32 MiB</code> <code>-S relatime</code>
<code>/gpfs/app</code>	<code>-j scatter</code> <code>-B 4 MiB</code> <code>-n <customer_specific></code> <code>--metadata-block-size 1 MiB</code> <code>-L 32 MiB</code>

File system	Settings
/gpfs/user (optional)	-j scatter -B 4 MiB -n <customer_specific> --metadata-block-size 1 MiB -L 32 MiB
/gpfs/ces	-j scatter -B 1 MiB -n <customer_specific>

IBM Spectrum Scale File Systems – Guidelines for Genomics Workload	
Name	/gpfs/data
Purpose	Store genomics data and analysis result
Why separate file system?	This file system is the workhorse to store most of the data
Size	Depends on customer requirements: Few TiB up to Hundreds of PiB
Metadata	1 MiB block size on SSD
Data	8 MiB block size on NL-SAS, no replication
Log File Size	32 MiB (-L 32M)
Block Allocation Map	Scatter
Replication	Replicate metadata only (-M 2 -R 2 -m 2 -r 1)
ACL Type	NFSv4 only
Filesets	Multiple independent filesets (details follow later)
Relatime	Suppress the periodic updating of the value of atime (-S relatime)
Quota	Enable quota (-Q yes) (avoids remount when we enable quota later)
Exported to Compute Cluster	Yes (via IBM Spectrum Scale multi-cluster remote cluster mount)
Exported via CES	Yes (SMB and NFS)
Number of Nodes	Customer specific

Figure 3-2 Details of the /gpfs/data file system

IBM Spectrum Scale File Systems – Guidelines for Genomics Workload	
Name	/gpfs/app
Purpose	Stores all applications binaries, scheduler binaries, configuration files and log files needed on the compute nodes
Why separate file system?	Maintenance on /gpfs/data (e.g. file system check) must not impact availability of applications on compute nodes
Size	Depends on customer requirements. Rule of thumb: ~50TiB at least
Metadata	1 MiB block size on SSD
Data	4 MiB block size on NL-SAS, no replication
Log File Size	32 MiB (-L 32M)
Block Allocation Map	Scatter
Replication	Replicate metadata only (-M 2 -R 2 -m 2 -r 1)
ACL Type	NFSv4 only
Filesets	Root fileset only
Relatime	Use default
Quota	Enable quota (-Q yes) (avoids remount when we enable quota later)
Exported to Compute Cluster	Yes (via IBM Spectrum Scale multi-cluster remote cluster mount)
Exported via CES	No
Number of Nodes	Customer specific

Figure 3-3 Details of the /gpfs/apps file system

IBM Spectrum Scale File Systems – Guidelines for Genomics Workload	
Name	/gpfs/user
Purpose	User data for execution of batch jobs (optional file system)
Why separate file system?	Isolate activity from other Separate Scale file systems
Size	Depends on customer requirements. Rule of thumb: ~50GiB per user
Metadata	1 MiB block size on SSD
Data	4 MiB block size on NL-SAS, no replication
Log File Size	32 MiB (-L 32M)
Block Allocation Map	Scatter
Replication	Replicate metadata only (-M 2 -R 2 -m 2 -r 1)
ACL Type	NFSv4 only
Filesets	Root fileset only
Relatime	Use default
Quota	Enable quota (-Q yes) (avoids remount when we enable quota later)
Exported to Compute Cluster	Yes (via IBM Spectrum Scale multi-cluster remote cluster mount)
Exported via CES	No
Number of Nodes	Customer specific

Figure 3-4 Details of the /gpfs/user file system

IBM Spectrum Scale File Systems – Guidelines for Genomics Workload	
Name	/gpfs/ces
Purpose	Metadata for Cluster Export Services (CES)
Why separate file system?	Isolation from all other file systems to increase resiliency of NFS and SMB
Size	64 GiB
Metadata + Data	1 MiB block size on SSD, System Pool only
Log File Size	32 MiB (-L 32M)
Block Allocation Map	Scatter
Replication	Replicate data and metadata (-M 2 -R 2 -m 2 -r 2)
ACL Type	NFSv4 only
Filesets	Root fileset only
Relatime	Use default
Quota	No
Exported to Compute Cluster	No
Exported via CES	No
Number of Nodes	Customer specific, typically 32 or 64

Figure 3-5 Details of the /gpfs/ces system

3.2.3 IBM Spectrum Scale filesets

Before configuration of the file systems, the use of IBM Spectrum Scale filesets should be considered. A *fileset* is a subtree of a file system namespace that provides a means of partitioning the file system to allow administrative operations. From a user point-of-view, a fileset looks like a directory.

There are two types of filesets:

- An *independent* fileset has its own inode space.
- A *dependent* fileset shares its inode space with an associated independent fileset.

Some data management functions have a dependency to the fileset type. Consider dependent filesets to use advanced placement policies and ILM tiering. Independent filesets should be considered if you want to use project-level quotas, snapshot, and AFM in addition to advanced placement policies and ILM tiering. There is a hard limit of 1,000 independent filesets and 10,000 dependent filesets. Keep these limits in mind when designing the use of filesets.

Filesets are not required, but filesets are a great tool to effectively automate data and capacity management. You need to configure filesets from the beginning, because introducing filesets or changing fileset boundaries later might trigger expensive copy or move operations. For this reason, it is advised to create at least one independent fileset for each file system. This method allows for the configuration of data management later, even if it is not required at the beginning.

The fileset design is customer specific and depends on how you organize data. An example of where independent filesets might be useful is in the `/gpfs/data` file system. Establishing independent filesets for each project in the `/gpfs/data` file system (for example, `/gpfs/data/project1`, `/gpfs/data/project2`, and so on) can improve automated data management (as described in Section 3.6).

Table 3-3 depicts the filesets and directories that recommended for this blueprint.

Table 3-3 Recommended directories and IBM Spectrum Scale filesets

Path	Description
<code>/gpfs</code>	► Directory under Linux root file system ("/")
<code>/gpfs/data</code>	► IBM Spectrum Scale File System under <code>/gpfs</code>
<code>/gpfs/data/project1</code> <code>/gpfs/data/project2</code> and so on	► Use independent filesets under <code>/gpfs/data</code> , if you do not hit the limit of 1,000 independent filesets. ► You might want to choose a different naming convention.
<code>/gpfs/app</code>	► IBM Spectrum Scale File System under <code>/gpfs</code> ► Directory structure for workload scheduler needs special consideration. See Section 2.3.1 for details.
<code>/gpfs/user</code>	► IBM Spectrum Scale File System under <code>/gpfs</code>
<code>/gpfs/ces</code>	► IBM Spectrum Scale File System under <code>/gpfs</code>

3.3 Block storage layer

The *block storage layer* provides the storage capacity for the IBM Spectrum Scale file systems. IBM Spectrum Scale supports a broad variety of different storage devices. You need to optimize the end-to-end tuning of a IBM Spectrum Scale environment for the workload and the underlying storage devices. This blueprint uses IBM ESS as storage device.

3.3.1 IBM Elastic Storage Server

IBM ESS includes IBM Spectrum Scale RAID (also known as *GPFS Native RAID* or GNR). IBM Spectrum Scale RAID has the following capabilities that makes ESS an excellent choice for performance and resiliency:

- Fast disk rebuilds: Disks rebuild in minutes versus hours or days of traditional RAID 5 and RAID 6.
- End-to-end data integrity: IBM Spectrum Scale RAID maintains checksum of data blocks from the client to the blocks on the disk and validates at every point, thus eliminating the chances of silent data corruption or data loss.
- Higher storage resiliency: The erasure coding is with up to three parity blocks and can survive three disk failures with only 27% overhead in capacity compared to 200% overhead with three-way replication. It uses fault domains to layout disks in such a way that it can survive entire disk shelf (enclosure) failures. It also uses a disk hospital to pro-actively identify sick drives (disks with bad sectors or media errors) and either a) replace the disk or b) fix any bad data from parity.

ESS provides two encodings for data protection, replication (mirroring) and erasure encoding (distributed RAID). Store the metadata on the SSDs with 4-way replication to provide fast metadata access and updates. Fast metadata access and updates are important for many automated data management functions. 4-way replication protects against three disk failures and enclosure failure. Note that this replication on the storage layer is different to the replication within the IBM Spectrum Scale file system (Section 3.2.1).

Store data with average performance requirements, such as genomics data, on NL-SAS, because it is cheaper than SSDs. We configured the ESS GL6S with 8+3P erasure encoding to protect against three disk failures and enclosure failure.

The example environment for this paper used the ESS GL2S with 4-way replication for metadata replication and the ESS GL6S with 8+3P erasure encoding for the data for the three file systems that store genomics data, application files and user data (as listed in Table 3-4). For the small CES file system, we configured one storage pool and kept everything on SSD with 4-way replication.

Table 3-4 ESS and IBM Spectrum Scale RAID configuration

File system	Metadata	Data
/gpfs/data	4W on SSD	8+3P on NL-SAS
/gpfs/app	4W on SSD	8+3P on NL-SAS
/gpfs/user	4W on SSD	8+3P on NL-SAS
/gpfs/ces	4W on SSD	N/A (System Pool only)

3.4 File access layer

The data transfer nodes of the file access layer enable data ingest from genome sequencers, microscopes, and so on, access by data scientists and physicians, and for sharing across sites and institutions. This blueprint covers secure access via NFS and SMB.

3.4.1 NFS and SMB

IBM Spectrum Scale has built-in support for NFS and SMB via IBM Spectrum Scale CES. Access by devices, such as for data ingest by sequencers and microscopes, is determined by the interfaces that the devices provide. Most devices provide a capability to write acquired data to an SMB share or NFS export. Field experiences shows that SMB provides effective access for notebooks and workstations running Windows, Linux, and macOS.

CES depends on an external authentication and an ID mapping source for user identification and user authentication, such as LDAP or Microsoft Active Directory, and on an external network to connect external devices and users. See IBM Knowledge Center IBM Spectrum Scale documentation [Planning for protocols](#) and [Deploying protocols](#) for more information, or engage an IBM professional services for assistance.

It is a general preferred practice to connect workstations and notebooks of user's like data scientists via SMB. Physicians typically access results via a download via portal. The download portal is typically a custom development that is outside the scope of the blueprint.

For genomics workloads, export only the `/gpfs/data` file system via CES. Devices such as sequencers and microscopes typically support data acquisition via SMB or NFS. The `/gpfs/app` and `/gpfs/user` file systems are not exported via CES to avoid potential performance impact of running batch jobs by concurrent NFS or SMB access. The `/gpfs/ces` file system is not exported via CES, because that is an internal file system for CES metadata only.

3.5 General recommendations

Preferred practices increase the operational efficiency for managing the entire storage infrastructure. This section provides recommendations for IBM Spectrum Scale, external dependencies, and some communication and security aspects.

3.5.1 Recommendations for IBM Spectrum Scale

All nodes of the storage cluster are configured as IBM Spectrum Scale nodes. Each storage node can belong to several possible node designations. Figure 3-6 summarizes the node designation of the example configuration that was introduced in Section 1.4.

	Node Type	Memory	IBM Spectrum Scale Node	IBM Spectrum Scale Quorum Node	IBM Spectrum Scale Manager Node	IBM Spectrum Scale Admin Node	IBM Spectrum Scale Contact Node	IBM Spectrum Scale GUI Node
ESS EMS	ESS Mgmt	32 GB (*)	X			X		X
ESS GS2S I/O 1	ESS I/O	256 GB (**)	X				X	
ESS GS2S I/O 2	ESS I/O	256 GB (**)	X				X	
ESS GL6S I/O 1	ESS I/O	256 GB (**)	X				X	
ESS GL6S I/O 2	ESS I/O	256 GB (**)	X				X	
CES Protocol 1	CES	128 GB	X	X	X	X		
CES Protocol 2	CES	128 GB	X	X	X	X		
CES Protocol 3	CES	128 GB	X	X	X	X		
(*) ESS EMS Nodes are always configured with 32 GB memory. (**) ESS I/O Nodes are always configured with 256 GB memory.								

Figure 3-6 Designation of the compute nodes

ESS I/O nodes and ESS EMS are always Power servers. We choose Power servers for the CES protocol nodes to reuse the tooling for monitoring and installation. This configuration improves operational efficiency.

Regarding memory, ESS I/O nodes are available with 256 GB memory and ESS EMS nodes are available with 32 GB only. For the protocol nodes, the example configuration uses 128 GB, because NFS and SMB benefit from more memory, and they are also configured as quorum and manager nodes. Having 128 GB memory on the protocol nodes reduces the likelihood that they run out of memory. This configuration improves the resiliency of the entire IBM Spectrum Scale storage cluster, because out-of-memory conditions on quorum or manager nodes impacts cluster stability.

We have explained general advice for the IBM Spectrum Scale quorum nodes, manager nodes, admins nodes, and GUI nodes already for the compute cluster (Section 2.5.2). The same suggestions apply for the storage cluster, with the additional constraint that ESS I/O nodes must not be configured as either a quorum node or a manager node (as described in [IBM Knowledge Center](#)) and that only the ESS EMS can be configured as an IBM Spectrum Scale GUI node. Therefore, the example environment configures all three protocol nodes as quorum and manager nodes.

In addition, you need to configure contact nodes for IBM Spectrum Scale multi-cluster remote cluster mount. Contact nodes are required on IBM Spectrum Scale clusters that export IBM Spectrum Scale file systems to other IBM Spectrum Scale clusters via a multi-cluster remote cluster mount.

For the example environment, we configured the ESS I/O nodes as *contact nodes*, because remote clusters have dependencies to the storage but not to the protocol nodes. Configuring the ESS I/O nodes as contact nodes enables the compute cluster to continue to access the genomics data, even if the protocol nodes are down. This configuration improves resiliency of the compute cluster.

We recognize that the ESS I/O nodes must be configured as temporary quorum nodes and temporary manager nodes, when all protocol nodes are down. However, this type of configuration is a different discussion, and we do not cover that topic here.

Finally, we need to discuss the *autoload option*. The autoload option determines whether IBM Spectrum Scale is started automatically when a node is booted. This setting is a node setting. Autoload is different from the automount option. The automount option indicates whether a file system is mounted automatically on all nodes. It is a preferred practice that all quorum nodes and manager nodes are configured with `autoload=yes`.

This setting increases the resiliency of the IBM Spectrum Scale cluster. It is also a preferred practice that all NSD client nodes are configured with `autoload=yes`. This setting simplifies the management of the IBM Spectrum Scale cluster. ESS I/O nodes and EMS nodes are configured with `autoload=no`. This setting is the default setting configured by the ESS installation scripts.

3.5.2 External dependencies

There are several external dependencies that are essential for the storage services. IBM Spectrum Scale depends on a highly available name resolution service (typically DNS) for name resolution and reverse name resolution. IBM Spectrum Scale depends on time services (typically NTP) or time synchronization. Certain user and administrative commands depend on proper ID mapping. The ID mapping can be configured either locally on each node or preferably on a central ID mapping service (such as LDAP or Microsoft Active Directory).

The ESS installation scripts configure a DNS service and an NTP service on the ESS EMS. Both services need to be connected to the customer-provided DNS and NTP services. All IBM Spectrum Scale nodes of the storage cluster connect to the DNS and NTP services running on the ESS EMS. This configuration is different to the compute nodes that connect to the customer-provided DNS and NTP services via the compute cluster management nodes.

In addition, IBM Spectrum Scale admin nodes need to map UID and GIDs to user and group names and vice versa. A preferred practice is to configure all IBM Spectrum Scale nodes with an ID mapping to keep the configuration of all nodes the same in order to standardize and, therefore, to simplify the configuration. Therefore, each IBM Spectrum Scale admin node needs to connect to the customer provided ID mapping service.

3.5.3 Communication and security aspects

All nodes of the storage cluster must be able to communicate with each other. All contact nodes of the storage cluster must be able to communicate to all nodes of the compute cluster. All nodes used for administering the IBM Spectrum Scale must be able to do password-less root ssh and scp into any other node of the storage cluster. Sudo wrappers are not used in this blueprint.

3.6 Data management

The blueprint uses two IBM Spectrum Scale storage pools for each of the following file systems:

- ▶ /gpfs/data
- ▶ /gpfs/app
- ▶ /gpfs/user

Storage pools are visible for administrators but not for the user. The *system pool* is for metadata only and is on SSD only. The *data pool* is for data and is on NL-SAS. The separation of data and metadata enables fast metadata access and updates.

IBM Spectrum Scale provides many more functions to automate data management, such as quotas, snapshots, ILM placement and migration policies, integrated backup, and fast restore. Those functions are not covered by this version of the blueprint but, nevertheless, can be configured. See IBM Knowledge Center [IBM Spectrum Scale](#) for more details or to engage IBM professional services for assistance.

3.7 Tuning

This blueprint is based on IBM ESS 5.2 and IBM Spectrum Scale 4.2.3.4 on the ESS I/O nodes and IBM Spectrum Scale 4.2.3.4 (or higher) on the CES protocol nodes and compute nodes. For more information, refer to the following resources:

- ▶ IBM Knowledge Center [Elastic Storage Server \(ESS\) 5.2 for Power](#)
- ▶ [Elastic Storage Server Version 5.2 Quick Deployment Guide](#)

The tuning recommendations optimized for the Broad Institute GATK [best practices](#), IBM Elastic Storage Server (ESS) as back-end storage, and InfiniBand. For the storage services, this section includes the following tuning recommendation topics:

- ▶ IBM Elastic Storage Server
- ▶ Protocol nodes

3.7.1 IBM Elastic Storage Server

The ESS installation scripts installs the recommended firmware and software packages on all ESS I/O nodes and creates the IBM Spectrum Scale storage cluster (as described in [Elastic Storage Server Version 5.2 Quick Deployment Guide](#)). The generic tuning applied by the ESS installation scripts is a solid baseline for genomics workloads. Only a few settings need to be adjusted.

Configure the following setting in the `/etc/sysctl.conf` file:

```
net.core.somaxconn = 8192
```

Apply the following settings in IBM Spectrum Scale if InfiniBand is used for the high-speed data network:

```
mmchconfig socketMaxListenConnections=8192 -N <essio_node_class>  
mmchconfig envVar="MLX4_USE_MUTEX=1 MLX5_SHUT_UP_BF=1 MLX5_USE_MUTEX=1" -N <essio_node_class>
```

After applying all changes, a snippet of the `mmfsconfig` output looks as follows:

```
[ESS I/O Nodes]  
nsdRAIDBufferPoolSizePct 80  
maxBufferDescs 2m  
nsdRAIDTracks 128k  
nsdRAIDSmallBufferSize 256k  
nsdMaxWorkerThreads 3k  
nsdMinWorkerThreads 3k  
nsdRAIDSmallThreadRatio 2  
nsdRAIDThreadsPerQueue 16  
nsdRAIDEventLogToConsole all  
nsdRAIDFastWriteFSDDataLimit 256k
```

```

nsdRAIDFastWriteFSMetadataLimit 1M
nsdRAIDReconstructAggressiveness 1
nsdRAIDFlusherBuffersLowWatermarkPct 20
nsdRAIDFlusherBuffersLimitPct 80
nsdRAIDFlusherTracksLowWatermarkPct 20
nsdRAIDFlusherTracksLimitPct 80
nsdRAIDFlusherFWLogHighWatermarkMB 1000
nsdRAIDFlusherFWLogLimitMB 5000
nsdRAIDFlusherThreadsLowWatermark 1
nsdRAIDFlusherThreadsHighWatermark 512
nsdRAIDBlockDeviceMaxSectorsKB 8192
nsdRAIDBlockDeviceNrRequests 32
nsdRAIDBlockDeviceQueueDepth 16
nsdRAIDBlockDeviceScheduler deadline
nsdRAIDMaxTransientStale2FT 1
nsdRAIDMaxTransientStale3FT 1
nsdMultiQueue 512
nspdQueues 64
numaMemoryInterleave yes
maxFilesToCache 128k
maxMBpS 16000
workerThreads 1024
ioHistorySize 64k
verbsRdma enable
verbsRdmaSend yes
verbsRdmPerConnection 128
verbsSendBufferMemoryMB 1024
scatterBufferSize 256K
nsdClientChecksumTypeLocal ck64
socketMaxListenConnections 8192
envVar MLX4_USE_MUTEX=1 MLX5_SHUT_UP_BF=1 MLX5_USE_MUTEX=1
maxStatCache 0
pagepool <60% of memory>
verbsPorts <active_verbs_ports>

```

Settings in **bold** font are changed for tuning. All other lines are the IBM Spectrum Scale default settings.

3.7.2 Protocol nodes

You need to configure the same settings on the *protocol nodes* as you did for the compute nodes. See Section 2.6.1 and Section 2.6.2 for details.

The tuning for IBM Spectrum Scale is slightly different because the NFS and SMB workload that is running on the protocol nodes is different from the GATK workload that is running on the compute nodes.

Because the storage backend is ESS, the `gssClientConfig.sh` script needs to be executed. The script is located on the ESS Management Server (EMS) in the `/usr/lpp/mmfs/samples/gss` directory. Configure each protocol node with 32 GiB page pool for IBM Spectrum Scale so that the syntax of the command looks as follows:

```
gssClientConfig.sh -P 32768 <protocol_node_class>
```

The page pool is increased to 32 GiB so that NFS and SMB can benefit from IBM Spectrum Scale caching.

Apply the following settings in IBM Spectrum Scale if InfiniBand is used for the high-speed data network:

```
mmchconfig maxFilesToCache=2M -N <protocol_node_class>
mmchconfig maxMBpS=20000 -N <protocol_node_class>
mmchconfig socketMaxListenConnections=8192 -N <protocol_node_class>
mmchconfig envVar="MLX4_USE_MUTEX=1 MLX5_SHUT_UP_BF=1 MLX5_USE_MUTEX=1" -N <protocol_node_class>
```

Increasing the **maxFilesToCache** value is a general preferred practice for protocol nodes to cache the file inodes for recently used files that have been closed and, thereby, can improve the NFS and SMB performance.

After applying all changes a snippet of the **mmfsconfig** output looks as follows:

```
[protocol]
pagepool 32768M
numaMemoryInterleave yes
maxFilesToCache 2M
maxStatCache 0
maxMBpS 20000
workerThreads 1024
ioHistorySize 4k
verbsRdma enable
verbsRdmaSend yes
verbsRdmasPerConnection 256
verbsSendBufferMemoryMB 1024
ignorePrefetchLUNCount yes
scatterBufferSize 256k
nsdClientCksumTypeLocal ck64
nsdClientCksumTypeRemote ck64
socketMaxListenConnections 8192
envVar MLX4_USE_MUTEX=1 MLX5_SHUT_UP_BF=1 MLX5_USE_MUTEX=1
verbsPorts <active_verbs_ports>

[common]
cipherList AUTHONLY
adminMode central
```

Settings in **bold** font are changed for tuning. All other lines are the IBM Spectrum Scale default settings.

3.8 Monitoring

IBM Spectrum Scale and ESS provide an integrated GUI to monitor and manage all IBM Spectrum Scale and ESS resources (Figure 3-7). For additional information refer to the following IBM Redpapers™ publications:

- ▶ *Monitoring Overview for IBM Spectrum Scale and IBM Elastic Storage Server*, [REDP-5418](#)
- ▶ *Monitoring and Managing IBM Spectrum Scale Using the GUI*, [REDP-5458](#)
- ▶ *Monitoring and Managing the IBM Elastic Storage Server Using the GUI*, [REDP-5471](#)

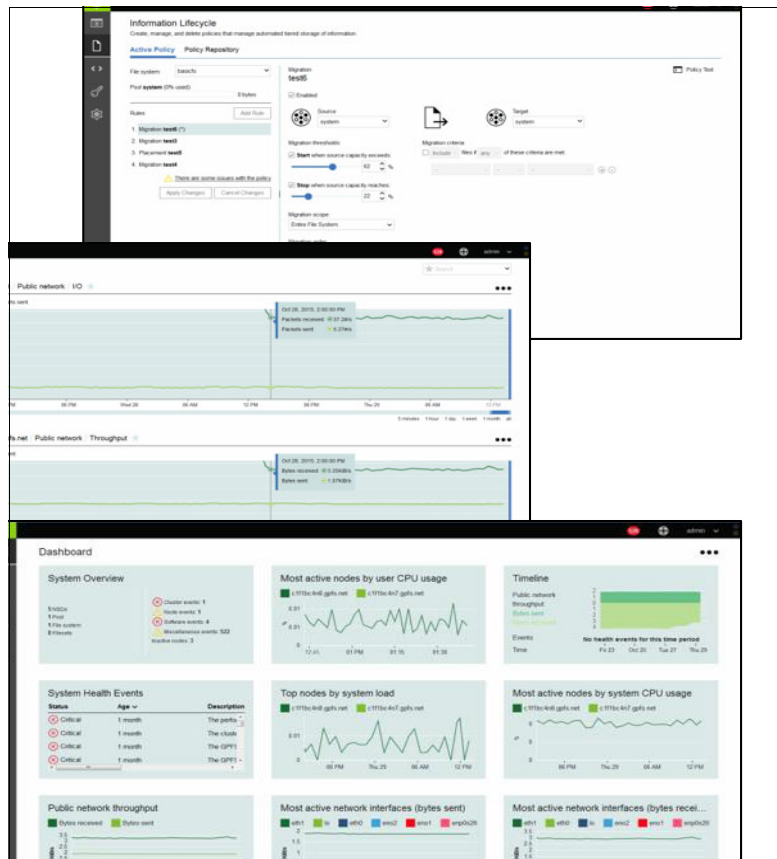


Figure 3-7 IBM Spectrum Scale and the IBM ESS GUI

The private network services

The purpose of the private network services is to integrate the compute services (Chapter 2) and the storage services (Chapter 3) into an IT infrastructure solution for genomics medicine workloads. The design of this piece of the composable infrastructure provides a high-speed data network for fast and secure access to genomics data and provisioning and service networks for the monitoring and management of all solution components.

The private network services design is based on five expertly engineered building blocks that enable IT architects to compose solutions that meet customers varying performance and functional needs (as illustrated in Figure 4-1). This chapter discusses each of these building blocks.

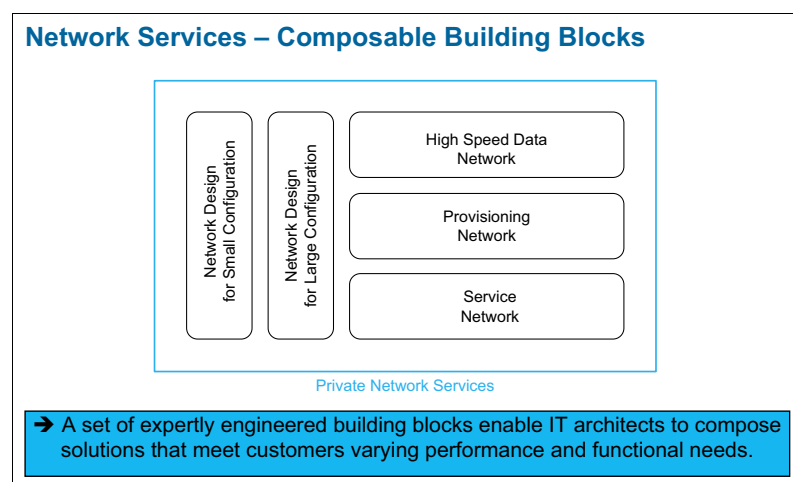


Figure 4-1 The building blocks of the private network services

This chapter includes the following topics:

- ▶ Overview
- ▶ High-speed data network
- ▶ Management networks
- ▶ Network designs

4.1 Overview

This section reiterates the capabilities of the private network services and maps them to solution elements.

4.1.1 Capabilities and solution elements

To integrate the compute services and the storage services into an IT infrastructure solution for genomics medicine workloads, the private network services provides the following capabilities:

- ▶ *A high-speed data network* for fast and secure access to genomics data:
 - Storage nodes are connected to the network with at least two links for high availability.
 - Compute nodes are connected to the network with one port or with two ports if you want high availability.
- ▶ *Provisioning networks* for provisioning and in-band management of the storage and compute components and for administrative login
- ▶ *Service networks* for out-band management and monitoring of all solution components
- ▶ *A scalable design* that can start small and grow to a large configuration that consists of hundreds of compute nodes and tens of PB of storage

Table 4-1 shows the mapping of solution elements to capabilities. The capabilities are written in a product neutral language to emphasize user requirements. The mapping of capabilities to solution elements shows how each selected solution element supports at least one solution capability. This method assures that the selected solution elements are kept at the absolute minimum that is required to support the above capabilities and that over engineering is avoided.

Note that the IBM Spectrum Scale Blueprint for Genomics Medicine Workloads uses InfiniBand for the high-speed data network. IBM Spectrum Scale provides other high-speed network technologies, such as 40 Gb Ethernet and 100 Gb Ethernet. Those network technologies are not covered by this version of the blueprint but, nevertheless, can be configured. See the [IBM Spectrum Scale documentation](#) for details or engage IBM professional services for assistance.

Table 4-1 Mapping of solution elements to the private network services capabilities

Private network services: Solution elements	
Capability	Provided by
A high speed data network for application communication and data access	InfiniBand
Provisioning networks for provisioning and in-band management and monitoring of all solution components	1 Gb Ethernet
Service networks for out-band management and monitoring of all solution components	1 Gb Ethernet
A scalable design that can start small and grow to a large configuration that consists of hundreds of compute nodes and tens of PB of storage.	Ready-to-use network layouts

4.1.2 Shared network

The private network services are different from the *shared network* (as illustrated in Figure 4-2, Figure 4-3 on page 48, and Figure 4-4 on page 49). The shared network connects the components of the IBM Spectrum Scale Blueprint with your environment and services. It is your responsibility to provide the shared networks.

The shared networks typically include a *campus* network and a *management* network (Figure 4-2). The campus network is usually a public network that is externally visible from the cluster. It is the primary path for users to access the system. Users access the workload management GUI over the campus network. The campus network is also the default path for movement of data into and out of the system via NFS and SMB provided by the storage services.

The management network is used by administrators or other privileged users to access elements that are not intended to be accessible to users. The management network is also used to connect to infrastructure services, such as NTP, DNS, and authentication services. Some customers deploy separate campus and management networks, whereas some customers combine the two types of networks. This blueprint can support either environment.

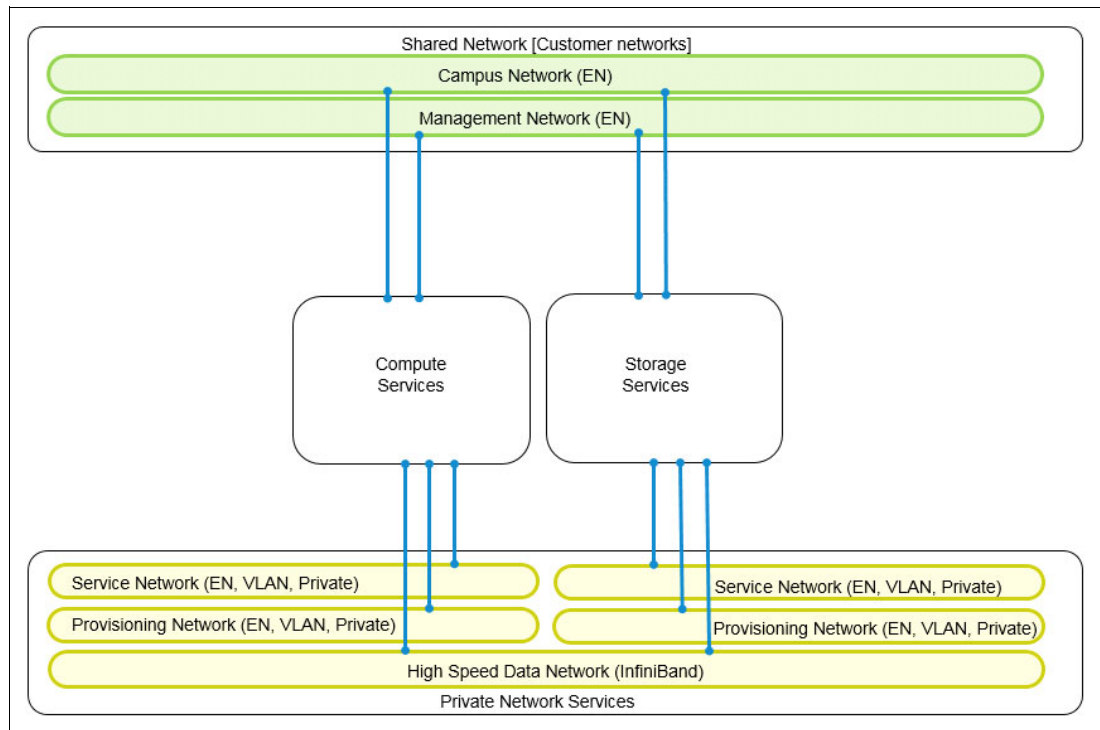


Figure 4-2 Shared networks integrate the solution with the customer environment

The private network services integrate the compute services and the storage services into an integrated IT infrastructure solution for genomics medicine workloads. The compute services and the storage services each have a *provisioning* network and a *service* network. The shared networks integrate the solution with the customer environment. The connectivity of each compute node and each storage node to the private and shared networks varies depending on each node's designations (as illustrated in Figure 4-3 and Figure 4-4 on page 49).

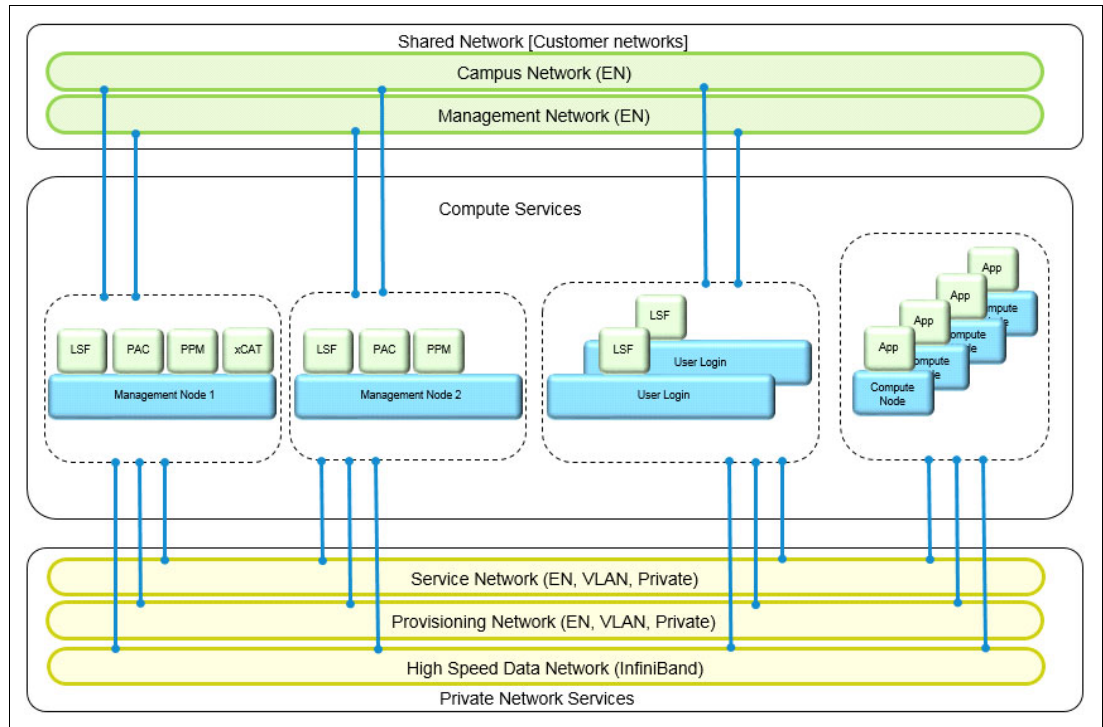


Figure 4-3 Compute node connectivity varies based on each node's designation

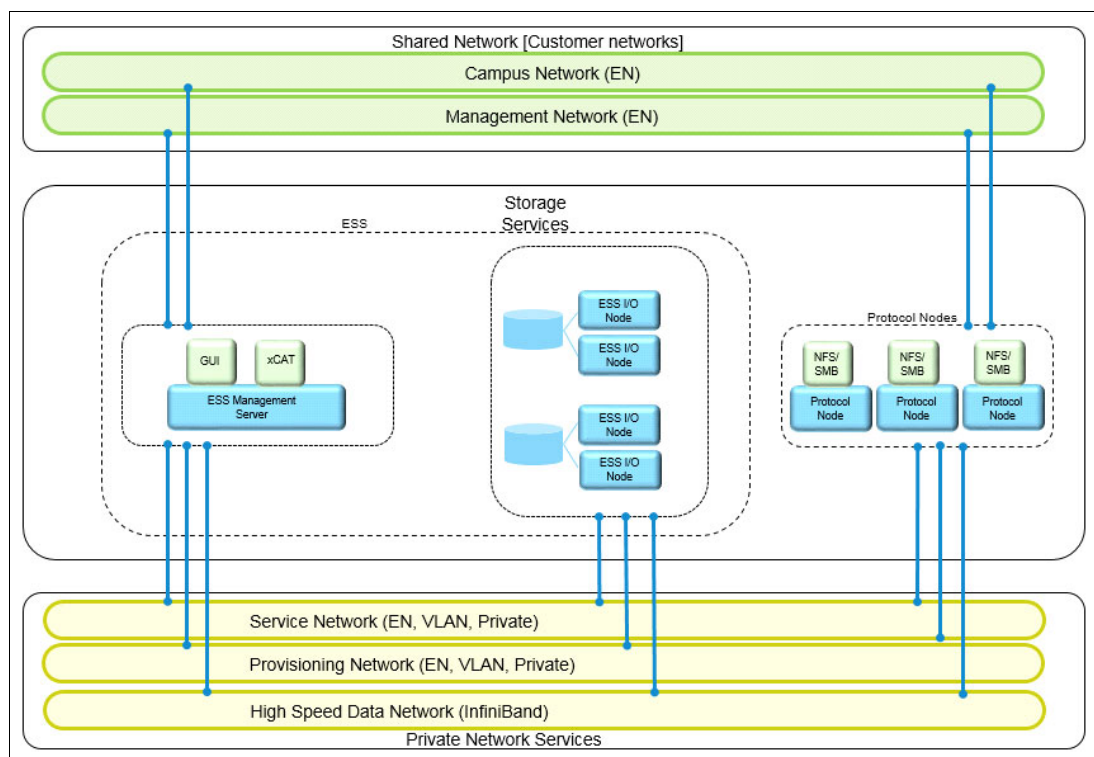


Figure 4-4 Storage node connectivity varies based on each node's designation

4.2 High-speed data network

The performance and reliability of IBM Spectrum Scale depends on the high-speed data network. A dedicated and stable high-speed data network connects all IBM Spectrum Scale nodes to provide high-speed data access. This section describes the following topics:

- ▶ IBM Spectrum Scale network requirements
- ▶ Recommendations for genomics medicine workloads
- ▶ Miscellaneous comments

4.2.1 IBM Spectrum Scale network requirements

IBM Spectrum Scale uses two networks:

- ▶ The IBM Spectrum Scale Daemon Network
- ▶ The IBM Spectrum Scale Admin Network

The daemon network is used for communication between the `mmfsd` daemon of all nodes. The daemon network requires TCP/IP. In addition to TCP/P, you can optionally configure IBM Spectrum Scale to use Remote Direct Memory Access (RDMA) for selected daemon communication. TCP/IP is still required if RDMA is enabled for daemon communication. The performance of IBM Spectrum Scale depends on the bandwidth, latency, and reliability of the IBM Spectrum Scale daemon network.

The IBM Spectrum Scale admin network is used for the execution of administrative commands. The admin network requires TCP/IP. The admin network can be either the same or a different network as the daemon network. The reliability of IBM Spectrum Scale depends on the admin network.

IBM Spectrum Scale is a clustered file system that depends on a high performance, low latency and stable network. The IBM Spectrum Scale `mmfsd` daemon runs on each node that participates in an IBM Spectrum Scale cluster. The `mmfsd` daemons of all cluster nodes need to communicate with each other to maintain a global cluster state, which includes distributed file and directory locks and a distributed cache.

This configuration requires low-latency Remote Procedure Call (RPC) communication and high throughput daemon communication between all IBM Spectrum Scale nodes. Non-blocking network fabrics meet IBM Spectrum Scale network requirements. Non-blocking network fabric means that the throughput between two nodes is not constrained by inter switch links.

It is a general preferred practice to connect all IBM Spectrum Scale nodes via a dedicated private high-speed data network for daemon communication and administrative communication. The private network services are not connected to external networks, such as the data center network or the Internet. You can connect IBM Spectrum Scale nodes to multiple networks in order to connect them to other servers and services (as shown in Figure 4-3 on page 48 and Figure 4-4 on page 49).

Experience in the field has proven that using the existing data center network can be problematic, because most shared networks are not designed for high throughput and low latency I/O. Other activity on the shared network can cause IBM Spectrum Scale performance, system health, stability, and other factors to degrade (for example, node failures can occur or commands can take a long time to complete).

Experience in the field has also proven that running this network over a shared infrastructure can be problematic. Carefully configure features such as VLAN and Quality of Service on shared links to support all protocols and ports that are used by IBM Spectrum Scale.

4.2.2 Recommendations for genomics medicine workloads

This blueprint recommends to use Mellanox InfiniBand EDR (100 GBps) switches for the high-speed data network. To meet the IBM Spectrum Scale network requirements (described in Section 4.2.1) and to fully use the bandwidth provided by the ESS systems, the storage cluster uses the following architectural configuration:

- ▶ All storage cluster nodes are connected for high availability and non blocking.
- ▶ All storage cluster nodes are connected with InfiniBand EDR (100 GBps).
- ▶ Both ESS GS2S I/O nodes are connected with four InfiniBand EDR links per node.
- ▶ Both ESS GL6S I/O nodes are connected with six InfiniBand EDR links per node.
- ▶ The ESS management node (EMS) is connected with two InfiniBand EDR links per node.
- ▶ All Cluster Export Services (CES) Protocol nodes are connected with two InfiniBand EDR links per node.

To meet the IBM Spectrum Scale network requirements (Section 4.2.1) and to balance the costs for the compute cluster, the storage cluster also takes into account the following architectural considerations:

- ▶ Field experience has proven that InfiniBand FDR (56 GBps) is sufficient for the compute nodes because most genomics workloads have CPU or memory constraints but not I/O constraints.
- ▶ Cluster management nodes are connected for high availability and, therefore, include at least two InfiniBand FDR or InfiniBand EDR links.
- ▶ Worker nodes can be connected with one InfiniBand link to reduce cost or with two InfiniBand links for high availability. Both links are either InfiniBand FDR or InfiniBand EDR.

The IBM Spectrum Scale daemon network is provided by InfiniBand. IPoIB is enabled to provide TCP/IP. Bonding (active/passive) is enabled on nodes that have more than one InfiniBand link, and RDMA is enabled on all nodes. The IBM Spectrum Scale admin network uses TCP/IP over the same IPoIB network.

4.2.3 Miscellaneous comments

Each InfiniBand fabric requires a *subnet manager*. See the Network Designs for details (Section 4.4).

Detailed monitoring of InfiniBand networks requires Mellanox Unified Fabric Management (UFM) software. UFM requires a software license and an x86-64 server. See [UFM documentation for details](#).

Some parallel applications require a high-speed network for interprocess communication (for example, MPI). Most genomics applications are single node jobs that do not need interprocess communication. The IBM Spectrum Scale daemon network and application traffic (for example, MPI) share the same InfiniBand network. A misbehaved application can impact the stability and performance of IBM Spectrum Scale. Administrators should monitor for such applications and limit their impact.

IBM Spectrum Scale depends on *static* IP addresses. The storage services provide static IP addresses for all storage nodes. You need to provide static IP addresses for all compute nodes.

4.3 Management networks

The private network services provide provisioning networks and service networks to monitor and manage all resources of the described solution. In total, four of those management networks are required (as illustrated in Figure 4-1 on page 45).

4.3.1 Provisioning network

There are separate provisioning networks for the storage services and the compute services. A *provisioning network* is a private network that is used by the cluster manager (for example, xCAT) to provision the compute and storage components of the solution and, subsequently, to manage and monitor those components. The provisioning network for IBM ESS is known as the *xCAT* network.

The storage cluster requires a dedicated private provisioning network. All nodes in the storage cluster need a single connection to the provisioning network. DHCP is used to assign static IP addresses for all interfaces on the provisioning network. SNMP monitoring of the provisioning network components is out of scope for this blueprint. HA for the provisioning network is out of scope for this blueprint. IPv6 is disabled on the provisioning network interfaces on all nodes.

Compute nodes cannot be connected to the provisioning network for the storage services. They must be separate as each management software includes its own DHCP server, and DHCP does not allow two DHCP servers on the same network. In most cases, you will already have a provisioning network in your data center. Otherwise, you need to configure a provisioning network for the compute nodes.

4.3.2 Service network

There are separate service networks for the storage services and the compute services. A *service network* is typically a private Ethernet network that is used to access the management processors of the servers within a server. A management processor can be a flexible service processor (FSP), which is typical for Power Systems servers, or a baseboard management controller (BMC), which is typical for OpenPower and x86-64 servers. A cluster manager can use a protocol, such as IPMI, to do hardware discovery, power control, and out-band management and to monitor the solution components.

The storage cluster requires a dedicated private service network. The EMS management node (EMS) needs a single connection to the service network. The flexible service processor (FSP) port of each storage node (ESS and CES) needs a single connection to the service network. The FSP port of the EMS is optionally connected to a customer provided service network. DHCP is used to assign dynamic IP addresses to the FSP ports that are part of the service network. SNMP monitoring of the service network components is out of scope for this blueprint. HA for the service network is also out of scope for this blueprint.

Compute nodes cannot be connected to the service network for the storage services. They must be separate, because each management software includes its own DHCP server and DHCP does not allow two DHCP servers on the same network. In most cases, you will already have a service network in your data center. Otherwise, you need to configure a service network for the compute nodes.

4.4 Network designs

This blueprint provides ready-to-use network designs for a small configuration and a large configuration as described in the following sections.

4.4.1 Network design for small configuration

The data network for the storage services and the compute services comprises a pair of InfiniBand EDR (8828-E36) switches to support a redundant high-speed data network. The switches can be ordered with an ESS. The InfiniBand subnet manager is configured on the CES protocol nodes.

The provisioning network and the service network for the storage services comprises a 1 Gb Ethernet (8831-S52) switch that is shared with the service network. The switch can be ordered with an ESS. The switch is shared for both networks. Use an untagged VLAN to separate the provisioning network and the service network. The switch is configured with spanning tree disabled.

In addition, you need to provide the provisioning network and the service network for the compute services.

The blueprint provides a ready-to-use network design for a small configuration (as illustrated in Figure 4-5 on page 53). The design limits the number of compute nodes by the amount of InfiniBand Ports of the 8828-E36 switch. Each 8828-E36 InfiniBand switch has 36 ports. The storage nodes require 28 InfiniBand switch ports in total (14 in each switch). The remaining switch ports can be used for the compute nodes.

The network design provides high availability of all compute nodes. Each compute node is connected with both InfiniBand switches and there are no inter switch links. Each InfiniBand switch has 22 free ports. Therefore, the design supports the compute cluster with up to 22 management nodes, user login nodes, and worker nodes.

The InfiniBand subnet managers will run on the first two protocol nodes.

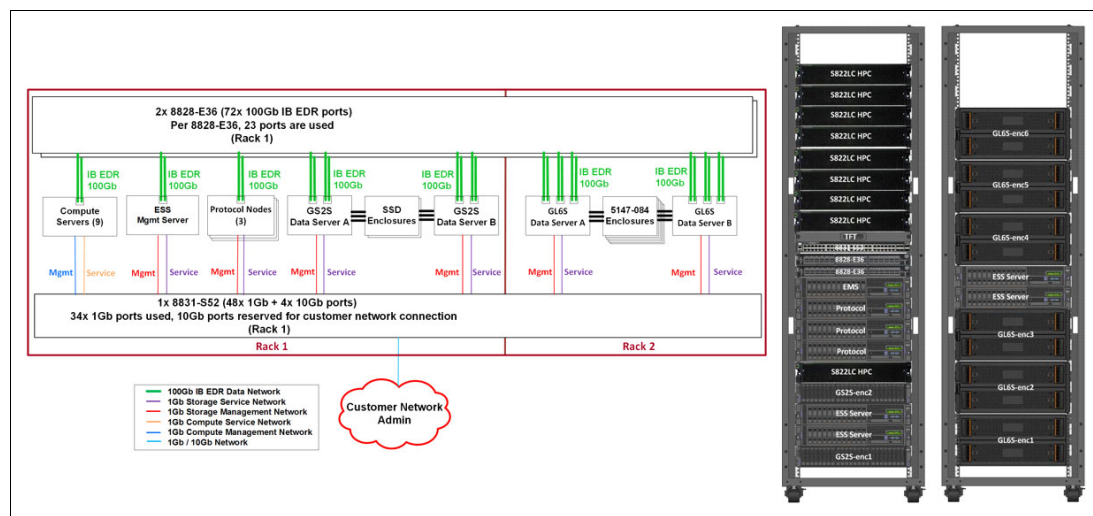


Figure 4-5 Network design for a small configuration

The network design for the small configuration supports up to 22 compute cluster nodes, including compute cluster management nodes, login nodes, and worker nodes.

4.4.2 Network design for large configuration

It is planned to add the network design for a large configuration in a future update of this paper. Use the large configuration from the beginning if you plan to grow the compute nodes and the storage nodes to exceed the number of available InfiniBand switch ports of the small configuration. Contact IBM for assistance with a large network configuration.



Profiling GATK

This appendix provides details about the profiling of the Genome Analysis Toolkit (GATK) workflow, which is described in Section 2.2.1.

Note: The product release levels indicated in this appendix were the ones used in the lab environment for this paper.

Figure A-1 shows step one of the Application Workflow, *Mapping to reference genome using BWA MEM*.

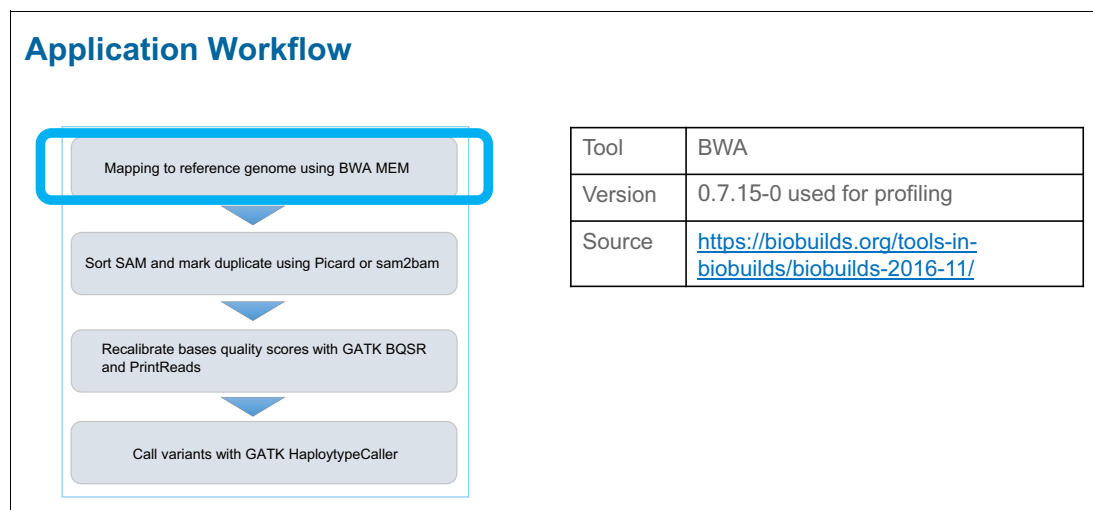


Figure A-1 Application Workflow, Mapping to reference genome using BWA MEM

Figure A-2 shows Application Profiling - BWA MEM.



Figure A-2 Application Profiling - BWA MEM

Figure A-3 shows step two of the Application Workflow, *Sort SAM and mark duplicate using Picard or sam2bam*.

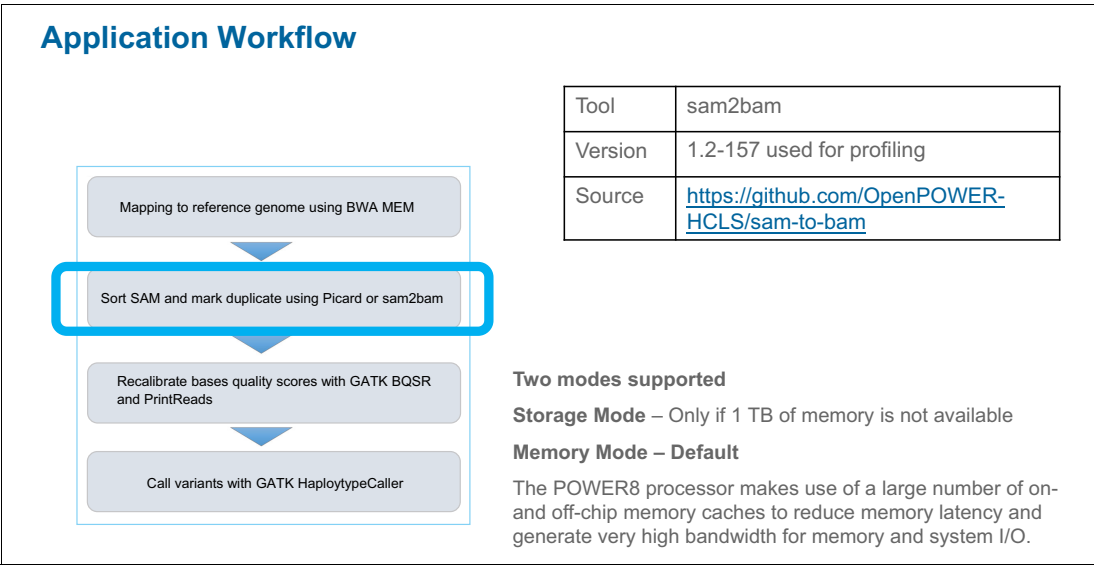


Figure A-3 Application Workflow, Sort SAM and mark duplicate using Picard or sam2bam

Figure A-4 shows Application Profiling - Sam2Bam (Storage Mode).

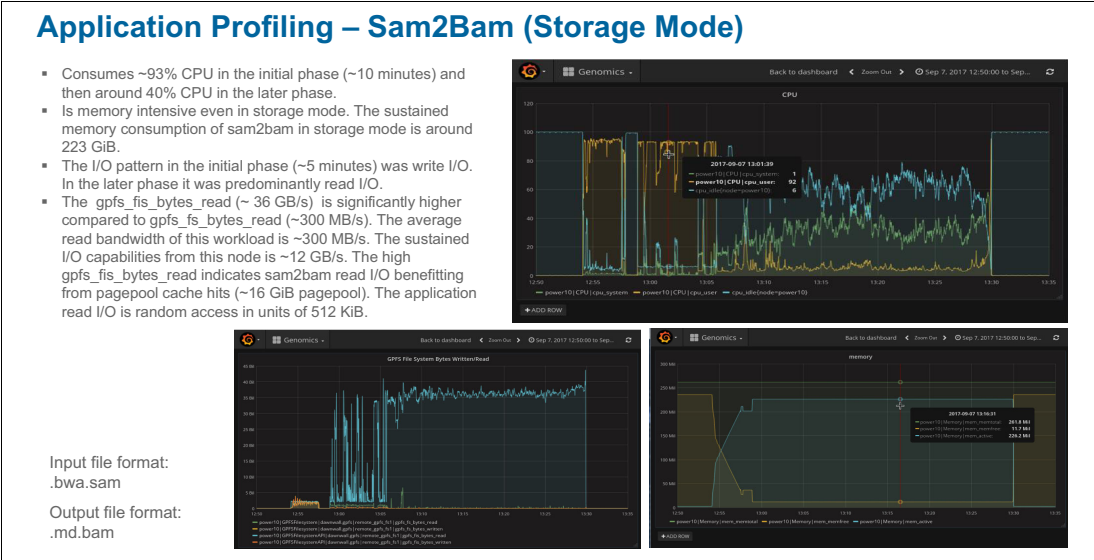


Figure A-4 Application Profiling - Sam2Bam (Storage Mode)

Figure A-5 shows step three of the Application Workflow, *Recalibrate bases quality scores with GATK BSQR and PrintReads*.

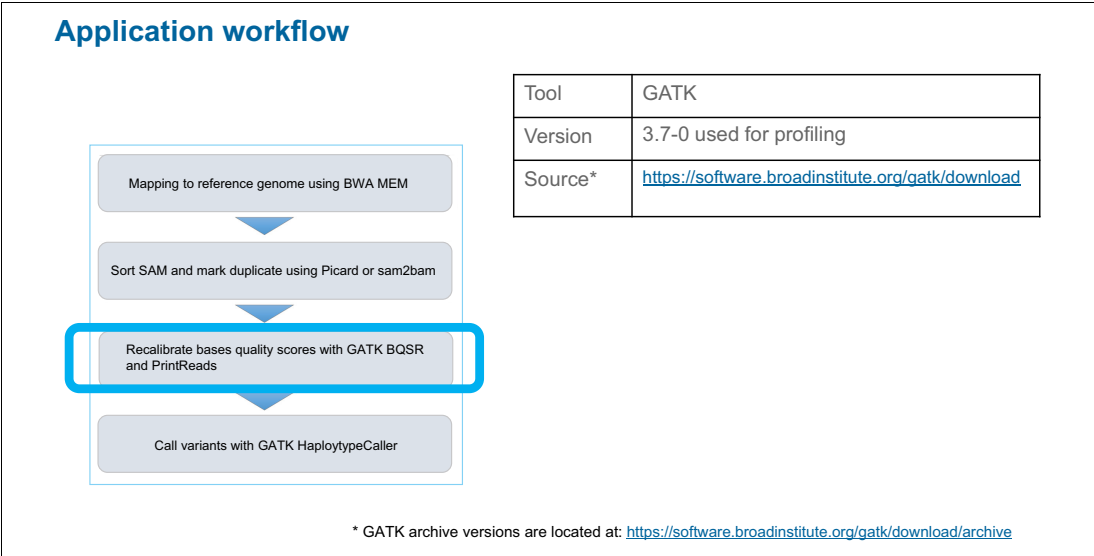


Figure A-5 Application Workflow, Recalibrate bases quality scores with GATK BSQR and PrintReads

Figure A-6 shows Application Profiling - GATK BQSR, and Figure A-7 shows GATK PrintRead.

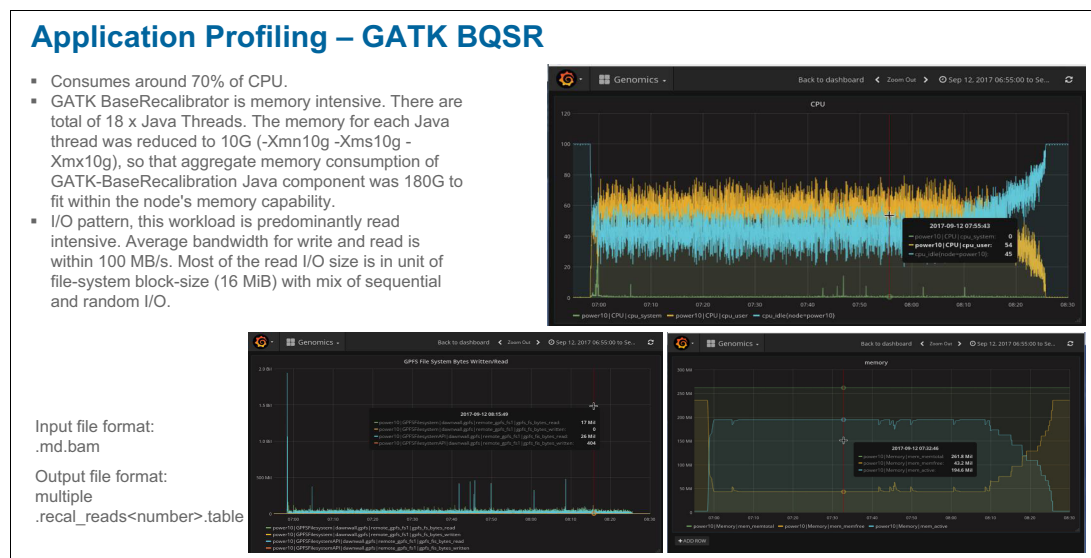


Figure A-6 Application Profiling - GATK BQR

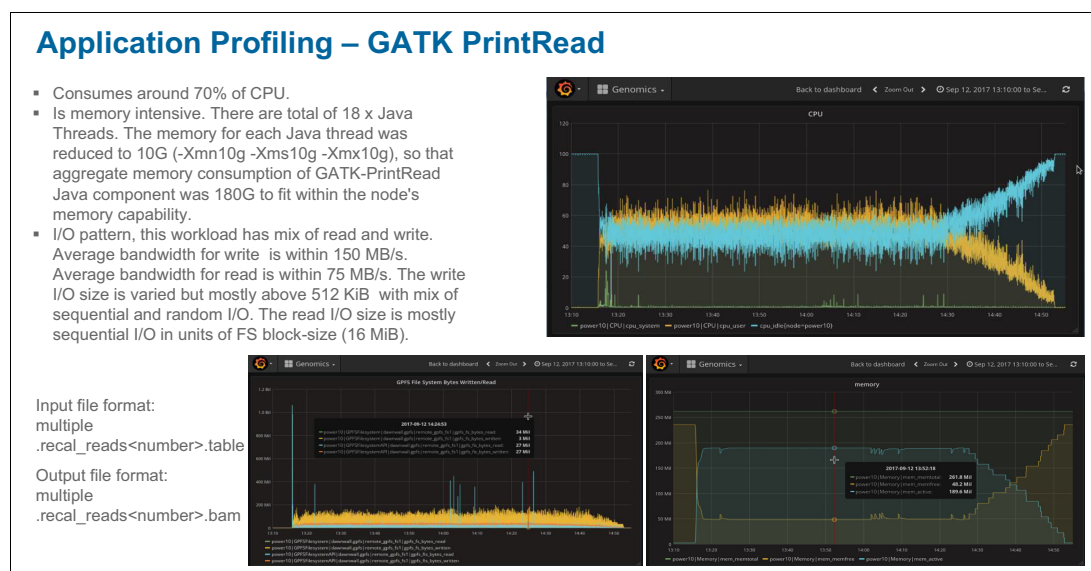


Figure A-7 Application Profiling - GATK PrintRead

Figure A-8 shows step four of the Application Workflow, *Call variants with GATK HaplotypeCaller*.

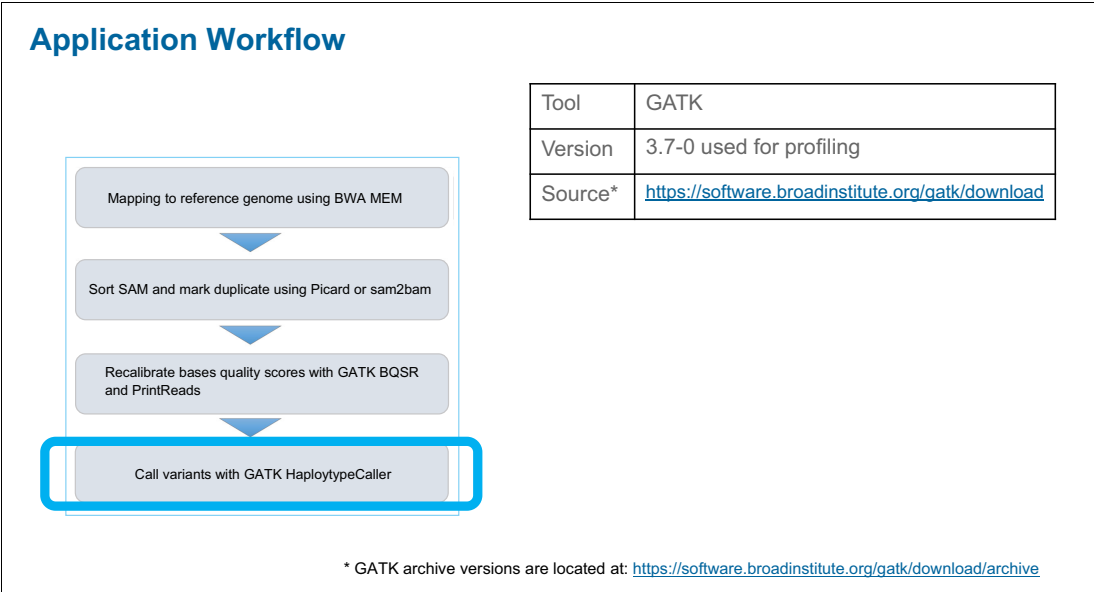
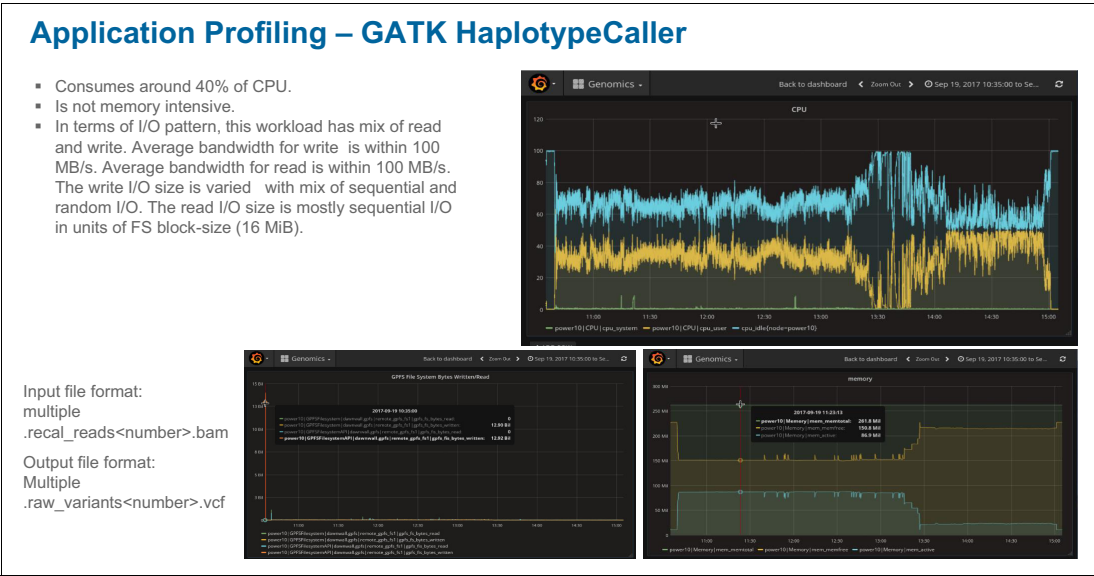


Figure A-8 Application Workflow, Call variants with GATK HaplotypeCaller

Figure A-9 shows Application Profiling - GATK HaplotypeCaller, and Figure A-10 on page 60 shows GATK MergeVCF.



Application Profiling – GATK MergeVCF

- Not CPU intensive.
- Is not memory intensive.
- I/O pattern, this workload has mix of read and write.
Average bandwidth for write is within 1.5 GB/s.
Average bandwidth for read is within 2 GB/s. The read I/O size is mostly sequential I/O in units of FS block-size (16 MiB). The write I/O size is mostly sequential I/O in units of FS block-size (16 MiB).

Input file format:
multiple
.raw_variants<number>.vcf

Output file format:
Single
.raw_variants.vcf file

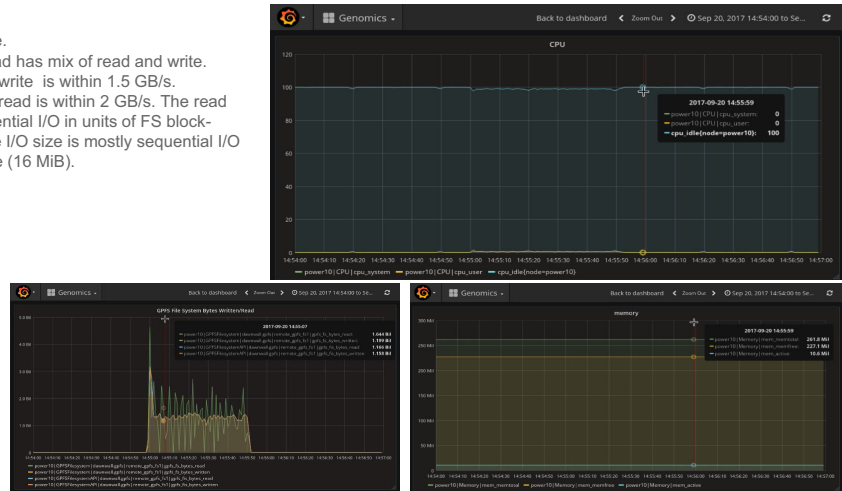


Figure A-10 Application Profiling – GATK MergeVCF

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

Online resources

These websites are also relevant as further information sources:

- ▶ Design Thinking Workshop Looks to Explore Growth Opportunities for Clients' Skills Development
<https://www.ibm.com/blogs/ibm-training/design-thinking-workshop-looks-explore-growth-opportunities-clients-skills-development>
- ▶ Getting started with agile principles
https://www.ibm.com/cloud/garage/content/culture/practice_agile_principles
- ▶ Broad Institute GATK
<https://software.broadinstitute.org/gatk>
- ▶ Performance optimization of Broad Institute GATK Best Practices on IBM reference architecture for healthcare and life sciences
<https://www.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=TSW03540USEN>
- ▶ IBM Spectrum Scale:
<https://www.ibm.com/support/knowledgecenter/STXKQY>
- ▶ IBM Spectrum Scale 4.2.3 - Planning for protocols - Authentication considerations:
https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.3/com.ibm.spectrum.scale.v4r23.doc/bllins_authconcept.htm
- ▶ IBM Spectrum Scale 4.2.3 - Planning for protocols - CES network configuration:
https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.3/com.ibm.spectrum.scale.v4r23.doc/blladv_cesnetworkconfig.htm
- ▶ IBM Spectrum Scale 4.2.3 - Planning for protocols - Planning for SMB:
https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.3/com.ibm.spectrum.scale.v4r23.doc/bllins_planningsmb.htm
- ▶ IBM Spectrum Scale 4.2.3 - Deploying protocols:
https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.3/com.ibm.spectrum.scale.v4r23.doc/bllins_deployingprotocoltasks.htm
- ▶ Elastic Storage Server (ESS) 5.2 for Power:
https://www.ibm.com/support/knowledgecenter/en/SSYSP8_5.2.0
- ▶ Elastic Storage Server (ESS) 5.2 for Power - Quick Deployment Guide:
https://www.ibm.com/support/knowledgecenter/SSYSP8_5.2.0/ess_qdg.pdf
- ▶ *Monitoring Overview for IBM Spectrum Scale and IBM Elastic Storage Server*, REDP-5418:
<https://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/redp5418.html>

- ▶ *Monitoring and Managing IBM Spectrum Scale Using the GUI*, REDP-5458:
<http://www.redbooks.ibm.com/Redbooks.nsf/RedpieceAbstracts/redp5458.html>
- ▶ *Monitoring and Managing the IBM Elastic Storage Server Using the GUI*, REDP-5471:
<http://www.redbooks.ibm.com/Redbooks.nsf/RedpieceAbstracts/redp5471.html>
- ▶ IBM Spectrum LSF:
<https://www.ibm.com/support/knowledgecenter/en/SSWRJV>
- ▶ IBM Spectrum LSF Application Center:
<https://www.ibm.com/support/knowledgecenter/en/SSZRJV>
- ▶ IBM Spectrum LSF Process Manager:
<https://www.ibm.com/support/knowledgecenter/en/SSZSHQ>
- ▶ IBM developerWorks: IBM Spectrum LSF and IBM Platform LSF best practices and tips:
<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/New%20IBM%20Platform%20LSF%20Wiki/page/LSF%20best%20practices%20&%20tips>
- ▶ Getting started with Performance Tuning of Mellanox adapters
<https://community.mellanox.com/docs/DOC-2490>
- ▶ Mellanox Unified Fabric Manager (UFM):
<https://www.mellanox.com/ufm>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



REDP-5479-01

ISBN 0738456578

Printed in U.S.A.

Get connected

