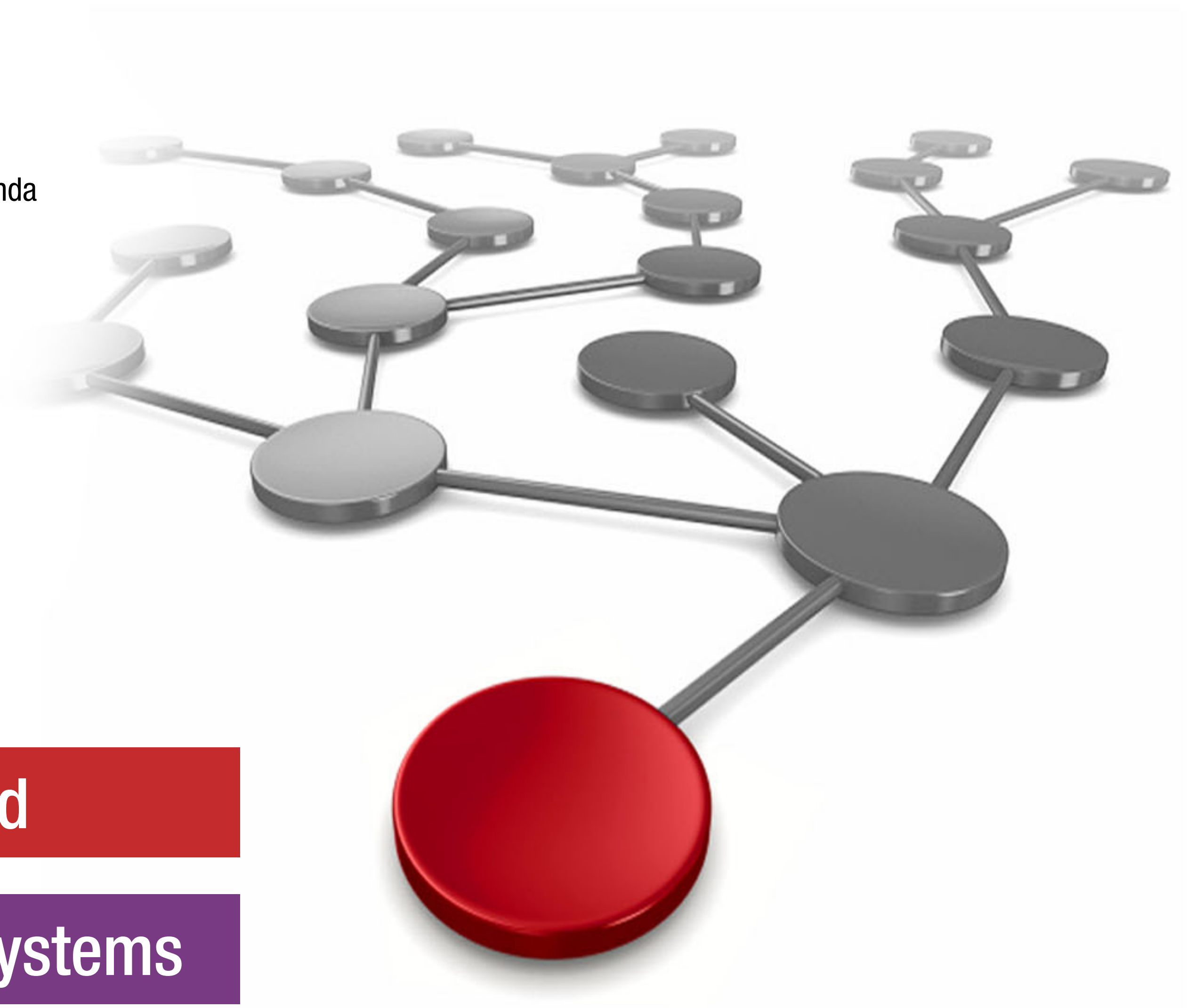
Redbooks
ibm.com/redbooks

Networking Design for HPC and AI on IBM Power Systems

Tobias Elpelt

Rico Franke

Yanil Zeledón Miranda



Cloud

Power Systems

IBM

Redpaper

International Technical Support Organization

Networking Design for HPC and AI on IBM Power Systems

April 2018

REDP-5478-00

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

First Edition (April 2018)

This edition applies to high performance computing and artificial intelligence networking running on IBM Power Systems servers.

Contents

Notices v
Trademarks vi

Preface vii
Authors. vii
Now you can become a published author, too! viii
Comments welcome. viii
Stay connected to IBM Redbooks viii

Chapter 1. Understanding the requirements 1
1.1 Performance 2
1.1.1 Network performance 2
1.1.2 Computing performance 2
1.1.3 Storage performance 2
1.1.4 Application and workload performance. 2
1.2 Scalability 4
1.3 Simplicity. 4
1.4 Reliability and availability 5
1.5 Power consumption. 5
1.6 Congestion 6

Chapter 2. Infrastructure for high-performance computing and artificial intelligence systems 7
2.1 IBM Power platform advantages 8
2.1.1 NVIDIA Tesla General Purpose Graphics Processing Unit. 9
2.1.2 NVLINK. 9
2.1.3 Memory bandwidth and PCI Express 9
2.1.4 Coherent Accelerator Processor Interface 9
2.2 InfiniBand 10
2.2.1 Technology 11
2.2.2 Topology 15
2.2.3 Hardware components 20
2.2.4 Tools 20
2.3 Ethernet. 22
2.3.1 Administration network 22
2.3.2 Out-of-band management network 22
2.3.3 Ethernet on the Fabric with RDMA over Converged Ethernet 23
2.4 Storage 23
2.4.1 IBM Spectrum Scale 23
2.4.2 IBM Elastic Storage Server. 24
2.5 Applications. 25
2.5.1 Message Passing Interface and other RDMA libraries 26
2.5.2 IBM PowerAI 26

Chapter 3. Designing a solution 29
3.1 Interconnect and data networks 30
3.1.1 Small solution 30
3.1.2 Medium solution 33
3.1.3 Large solution 37

3.1.4 Large solution upgrade 43
3.1.5 Extra-large solutions / summary 46
3.2 Administration and out-of-band management network 46

Related publications 49
IBM Redbooks 49
Online resources 49
Help from IBM 50

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.

GPFS™
IBM®
IBM Elastic Storage™
IBM Spectrum™
IBM Spectrum Archive™
IBM Spectrum Protect™
IBM Spectrum Scale™
OpenCAPI™
POWER®
Power Systems™
Redbooks®
Redbooks (logo) ®

The following terms are trademarks of other companies:

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

Microsoft, 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

This publication provides information about networking design for high-performance computing (HPC) and artificial intelligence (AI) for IBM® Power Systems™.

This paper helps you understand the basic requirements of designing a solution, the components in an infrastructure for HPC and AI systems, the designing of interconnect and data networks with use cases based in real life scenarios, and the administration and the out-of-band (OOB) management networks.

This paper covers all the requirements, provides a good understanding of the technology, and includes examples for small, medium, and large cluster environments. This paper is intended for IT architects, system designers, data center planners, and system administrators who must design or provide a solution for the infrastructure of an HPC cluster.

Authors

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

Tobias Elpelt is an IT Specialist for HPC in IBM Germany. He joined IBM in 2010 and has implemented and operated several small and large HPC environments. He holds a degree in Applied Computer Science. His background includes work with Linux and networks. He is Mellanox Professional Certified.

Rico Franke works as an IT Specialist for large Linux and HPC environments in Germany. He has more than 15 years of experience in supporting IBM product, and provides account support in the context of open source solutions. He leads the IBM operational support team at the Leibniz Supercomputing Centre, which services the warm-watercooled HPC cluster SuperMUC. He holds a degree of engineering in information technology.

Yanil Zeledón Miranda is a Solutions Architect at IBM Costa Rica. He joined IBM in 2016. His background includes network and data center architecture with over 15 years of experience. He holds certifications for Cisco CCNA, CCDA, CCAI, CCNP, CCDP, CCIP, Alcatel-Lucent AQPS and ACPS, Juniper JNSS, EMC SE, BlueCat SE, CMNA, and Certified Trainer.

The project that produced this publication was managed by:

Scott Vetter, PMP

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

Henry Brandt, Klaus Gottschalk, Markus Hilger, Florin Manaila, Dino Quintero
IBM

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 found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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

1

Understanding the requirements

Understanding the requirements of an application means that you must understand the application, the memory requirements, the CPU requirements, the time to process the information, how it scales, and how it must be modified to scale. But, understanding the requirements of a high-performance computing (HPC) cluster involves more than storage devices and the ideal combination of nodes that are connected through a high-bandwidth network.

This chapter describes the conventional design requirements for any HPC cluster. With a clear overview of these topics, you have a better understanding of why the interconnect network is important in the cluster and how each part fits in the solution.

Here are the basic requirements:

- Performance
- Scalability
- Simplicity
- Reliability and availability
- Power consumption
- Congestion

A conventional design requires many adjustments to its components until the technical requirements are met. Integration with other teams is essential to reach such criteria and integration. You must have a common understanding of the design and the viability of the solution.

1.1 Performance

Performance is the capabilities of a machine when it is observed under particular conditions. In an HPC, system nodes, switches, and storage devices must be evaluated.

The following sections describe the basic kinds of performance.

1.1.1 Network performance

Network performance and its relationship to the fabric is primarily measured from the throughput of a single host in the cluster to the access switch. This performance is measured by using two metrics that are based on reviewing the statistics and metrics from bandwidth and latency (delay). The bandwidth in the fabric describes the size of the pipe that is used to transmit the data. InfiniBand switches support values of over 100 Gbps, enabling an InfiniBand link to transmit data faster, and compared to Ethernet, it has low latency.

1.1.2 Computing performance

Computing performance is related to parallel processing for running advanced applications efficiently across the cluster. It is often measured in floating point operations per second (FLOPS). HPC supercomputers are ranked based on their performance by using the LINPACK Benchmark, which is used to measure a cluster's floating point computing power. This test does not reflect the overall performance of an HPC system, but reflects the performance of a system to solve dense linear equations. There are several other benchmarks, and some of them apply to real-world applications.

1.1.3 Storage performance

Storage performance has many factors, but they can be summarized as the speed of the source to read the information on a disk, the speed of the destination to write information on a disk, and the actions that occur between these two functions. Storage is commonly benchmarked in gigabytes per second (GBps), and is measured as aggregated throughput.

1.1.4 Application and workload performance

Application and workload performance must be deployed with cost-efficiency, high performance, reliability, and an infrastructure that is customized for the workload. State-of-the-art big data workloads are data-intensive with higher fidelity models and more interdisciplinary analysis, so processor performance depends on other system attributes, such as memory, networks, and storage. End-to-end HPC workflow performance is enhanced by scalable networks of servers and storage.

Figure 1-1 shows the performance characteristics for Analytics (Descriptive, Predictive, Prescriptive, and Deep Learning) mapped to the same seven key cluster system features that influence performance: FLOPS/core, number of cores, node memory capacity, memory bandwidth at each node, I/O performance, interconnect latency, and interconnect bandwidth.

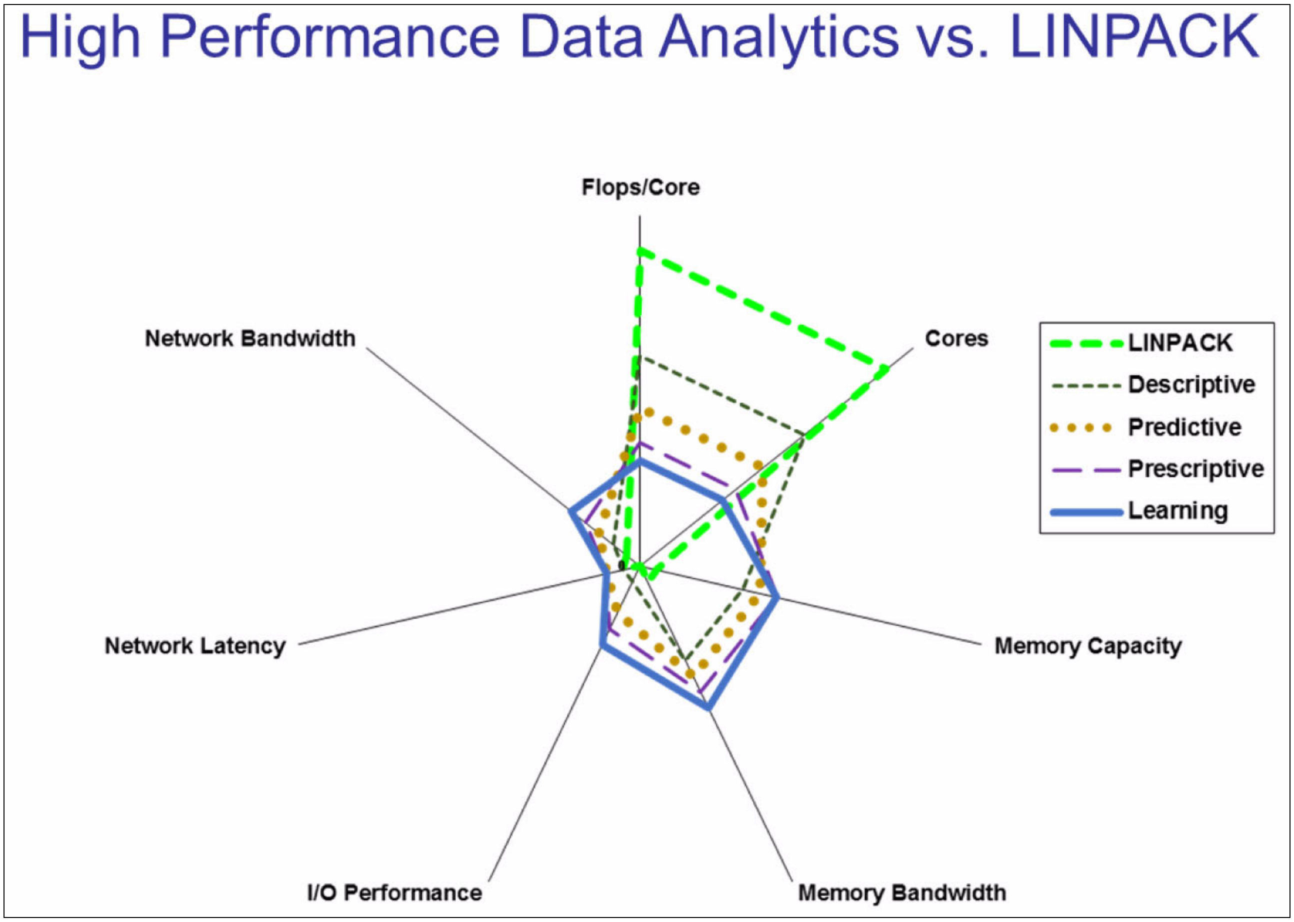


Figure 1-1 Performance characteristics for analytics

However, analytics and cognitive computing requires balanced, data-centric HPC systems. The emphasis on the capabilities of memory, network, and I/O performance relative to processor performance increases from descriptive to predictive to prescriptive to learning.

For more information, see the case study at HPC and HPDA for the Cognitive Journey with OpenPOWER.

The performance of the fabric affects the performance of the entire solution because the solution conforms to the backbone of the cluster.

1.2 Scalability

Systems (hardware and application) must grow in size and number to meet new requirements without affecting production, which is why it is important to select the correct topology from the beginning so that you can scale without affecting the cluster.

In general, there are two ways to scale:

- Scale up: You grow the computing node's power. For example, if you have a node with 2 CPUs/GPUs and you want to grow, you replace that node with another one that has 4 CPUs/GPUs. This growth is limited by the CPU/GPU platform.

  Figure 1-2 shows a scale-up situation.

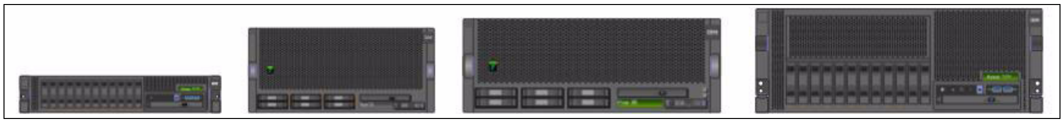


Figure 1-2 Scale up

- Scale out: You add more nodes and switches to the infrastructure, which is limited to the number of ports that are available from the network topology. Scaling out leads to an increase in latency and the requirement for more ports.

  Figure 1-3 shows a scale-out solution.

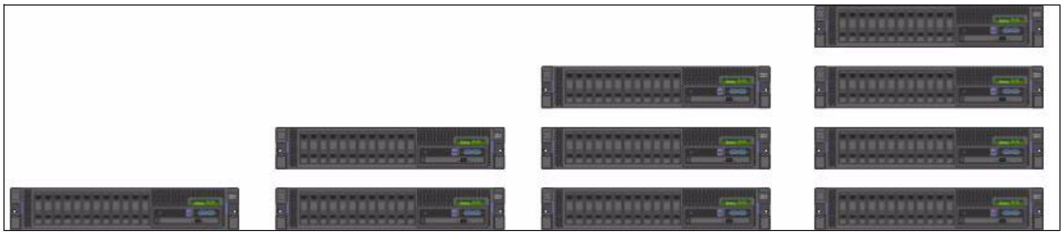


Figure 1-3 Scale out

In general, HPC systems can only scale out.

1.3 Simplicity

A solution should be simple to deploy, which shows a clear understanding of the technical requirements, and simple to operate and integrate with the tools that are used to manage it.

Consider the simplicity of the toaster. It has not changed much in the past 100 years, is easy to deploy, easy to operate, and integrates well into most kitchens. Over time, the user demands have not changed the basic requirements.

Imagine a cluster with 648 nodes on a fully non-blocking topology that uses a basic description. There are two general approaches to creating a system:

Approach 1 Use 54 one-rack unit (RU) fix switches with 36 ports each on a 2-layer fat-tree topology. You need a total of 648 cables between the switches for a non-blocking solution.

Approach 2 Use one 648-port director switch (for example, a Mellanox InfiniBand SX6536).

The most simple approach is the director switch that is populated with all the required hardware. Implementing a fat-tree topology requires more cabling, more equipment, more cooling, and more equipment to manage.

1.4 Reliability and availability

With many nodes, using high-speed connections might introduce more errors into the network than a common architecture or workload does. Ignoring these errors might impact the performance of the parallel programming.

High-performance applications can run for hours, days, weeks, or months. If there is a failure, such as a transmission failure (bit error), the software retransmits the missing information. If there is a hardware failure (for example, bad node, broken cable, or broken switch), the process stops, no retransmission is possible, and the entire data set might be lost.

HPC applications tend to be CPU-intensive, handle large quantities of data in fractions of seconds, and come from multiple servers across the cluster. Therefore, hardware failure and information going across multiple servers should immediately trigger the idea or need of *redundancy*.

There are many factors that can affect the availability of the network, such as the mean time between failures (MTBF), which is a measure of how reliable the hardware is. An HPC network has several nodes that are interconnected through several switches, and the MTBF is reduced inversely proportional to the amount of equipment. When one variable increases, other variables decrease in proportion. Again, redundancy is needed, and in this case, failover.

Defining a good redundancy strategy when you design the solution impacts the design and purchase costs. Redundancy has benefits such as avoiding a single point of failure and an increase in bandwidth. A balance between redundancy and cost efficiency can be achieved if important switches are redundant and not the entire cluster, or by connecting nodes by using two cables instead of one.

When you design a solution, it should be simple, cost-effective, and redundant where possible (for example, on the top of the fat-tree topology). Consider factors such as MTBF for reliability, and be proactive by using tools for monitoring and troubleshooting. Tools, such as the Open Fabrics Enterprise Distribution (OFED) package, are described later in this paper.

1.5 Power consumption

The power consumption of an HPC infrastructure is a large part the solution’s total cost of operation (TCO).

To calculate the energy efficiency of a system, the primary option is performance per watt, although for this discussion it should be *FLOPS Per Watt*, which is used to measure the rate of processing that can be delivered by a node for every watt of power that is consumed. The systems have lower consumption when idle and higher consumption when they reach peak performance. Power consumption directly affects the cooling system because lower performance means less cooling.

The interconnect is different. Even though the nodes might be idle, the ports that are used for the connectivity in the cluster are always active. There are approaches to create a low-power mode, but the time it takes to bring the port to an operational mode are fairly high, which creates a loss in performance and affects latency-sensitive applications running in the cluster. For a production environment, such an approach is not acceptable.

Cables affect power consumption because the increase in distance requires more power. There are two types of cables:

Passive cables Consume less power because they do not have electronic transceivers.

Active cables Consume more power because they require different types of transceivers to operate.

On the out-of-band (OOB) management network (the dedicated infrastructure for managing all the devices that are connected to the HPC system), you can use Gigabit Ethernet Switches that use the Energy-Efficient Ethernet (EEE) standard (IEEE 802.3az), which enables less power consumption during periods of low data activity. When the node decides that no data needs to be sent, it sends low-power idle (LPI) messages to the switch. Then, the node and switch periodically exchange messages to maintain the circuit as active even when the transmit path is in sleep mode.

Your design focus always is on performance, even if there are some savings on power consumption.

In comparison to compute node power consumption, network power consumption is small. In typical HPC systems, the share of the total power consumption is below 5%. Therefore, it is negligible.

1.6 Congestion

The performance of the HPC cluster is limited to the interconnect fabric when the fabric becomes saturated, which significantly degrades the performance across the HPC cluster. In larger fabrics, the degradation is reflected in the workload not having the expected performance. This situation might be an indication of a congestion problem.

A few factors that might cause congestion:

Blocking factor If blocking factor is not sized or configured properly, it can cause congestion in the topology. The most common or preferred topology in HPC is fat-tree, although there are other topologies that you can use, such as 2D and 3D mesh, 2D/3D torus, and Dragonfly. To avoid congestion, configure fat-tree to be fully Non-Blocking.

Unbalanced routing Constant link flapping (changing states to up or down) might trigger misconfiguration or miscalculations in the routing algorithm, which causes congestion.

Hardware problems Some of the common hardware problems that might cause congestion are broken switches, broken cables, and failing nodes. Any of these items might fail and cause congestion.

2

Infrastructure for high-performance computing and artificial intelligence systems

One of the requirements to design an effective network is to understand the available and upcoming infrastructure technology. This chapter describes the different parts of a high-performance computing (HPC) and artificial intelligence (AI) system with a focus on the network.

First, a short overview of the IBM Power platform is provided.

Then, this chapter provides a description of the interconnect that acts as the backbone of the system. There are two basic technologies that can be used to provide the backbone interconnect: InfiniBand and Ethernet with RDMA over Converged Ethernet (RoCE). For management and out-of-band (OOB) management, the Ethernet network is important too.

Additionally, storage is a factor for computing with a data-centric workload.

The last topic is an overview of the applications and libraries that are part of the solution.

2.1 IBM Power platform advantages

IBM Power Systems servers are designed for the most demanding and data-intensive computing. These servers unleash insight from your data pipeline, from managing mission-critical data, managing your operational data stores and data lakes, to delivering the best server for cognitive computing. With industry-leading reliability and security, the Power Systems infrastructure can manage the most data-intensive workloads.

Power Systems servers are deployed in many of the largest HPC clusters in the world. Configured into highly scalable Linux clusters, IBM Power Systems servers offer extreme performance for demanding workload, such as genomics, finance, computational chemistry, oil and gas exploration, and high-performance data analytics. An HPC cluster is a combination of high-performance compute nodes, a low-latency interconnect fabric with high bandwidth, high-performance parallel storage, and system software, which addresses the most challenging requirements for HPC and high-performance data analytics.

Although HPC and big data analytics are converging, traditional HPC clusters are built for another era, meaning that they are designed for data or computation throughput, but not both. These clusters cannot deliver adequate performance and scalability, and along with I/O bottlenecks and network latency when moving large data sets, the clusters slow down real-time insights. IBM HPC clusters deliver more when they are built with Power Systems because you can easily handle demanding workloads and high-performance data analytics.

The IBM and NVIDIA partnership was announced in November 2013 for integrating IBM POWER® systems with NVIDIA GPUs and the enablement of GPU-accelerated applications and workloads. The goal of this partnership is to deliver higher performance and better energy efficiency to companies and data centers.

The computational capability that is provided by the combination of NVIDIA Tesla GPUs and IBM Power Systems servers enables workloads from scientific, technical, and HPC to run on data center hardware. (In most cases, these workloads run on supercomputing hardware.) This computational capability is built on top of massively parallel and multithreaded cores with NVIDIA Tesla GPUs and IBM POWER architecture processors, where processor-intensive operations are offloaded to GPUs and coupled with the system's high memory-hierarchy bandwidth and I/O throughput.

In summary, IBM Power Systems servers with NVIDIA GPUs provide a computational powerhouse for running applications and workloads from several scientific domains, and for processing massive amounts of data. This data is sent across the network and shared with the HPC cluster.

For more information, see the following resources:

- NVIDIA Tesla Supercomputing
- OpenPOWER Foundation
- HPC and HPDA for the Cognitive Journey with OpenPOWER

This section describes the different parts and interconnects from the NVIDIA Tesla GPU through IBM Power Systems servers.

2.1.1 NVIDIA Tesla General Purpose Graphics Processing Unit

The NVIDIA Tesla General Purpose Graphics Processing Unit (GPGPU) is one of the most powerful and the most architecturally complex GPU accelerator ever built. It has a class-leading GPU, a high-performance interconnect that is called NVLINK that greatly accelerates GPU peer-to-peer and GPU-to-CPU communications, technologies to simplify GPU programming, Tensor cores, and exceptional power efficiency.

A powerful interconnect is valuable in multiprocessing systems. NVIDIA Tesla cards rely on traditional PCI Express (PCIe) for data transfers, but they also use the NVLINK bus that creates an interconnect for GPUs that offer higher bandwidth than PCIe can offer today.

2.1.2 NVLINK

NVLINK is the NVIDIA advanced interconnect technology for GPU-accelerated computing. It increases performance for both GPU-to-GPU communications and GPU access to system memory.

Although traditional NVLINK implementation primarily focuses on interconnecting multiple NVIDIA Tesla cards, with the IBM Power platform it also connects GPUs with CPUs, enabling direct system memory access and providing GPUs with extended memory orders of magnitude larger than the internal GPU memory.

2.1.3 Memory bandwidth and PCI Express

NVLINK is attached to the system memory, and has a bandwidth of hundreds of gigabits per second (Gbps). PCIe interfaces that are used to attach the network adapters also have access to memory. Here are sample bandwidths to illustrate the NVLINK advantage:

- 1 GPU =150 GBps bidirectional -> RAM
- 1 CPU = 120 GBps (per socket) -> RAM

PCIe uses a serial interface and enables point-to-point interconnections between devices by using a directly wired interface between these connection points. A single PCIe serial link is a dual-simplex connection that uses two pairs of wires, one pair for transmit and one pair for receive, and can transmit only 1 bit per cycle. These two pairs of wires are called a *lane*. A PCIe link uses multiple lanes.

2.1.4 Coherent Accelerator Processor Interface

Coherent Accelerator Processor Interface (CAPI) defines a coherent interface structure for attaching special processing devices to the IBM POWER processor bus. It attaches adapters that have coherent shared memory access to the processors in the server and shares full virtual address translation with these adapters by using standard PCIe buses.

The benefits of using CAPI include the ability to access shared memory blocks directly from the adapter, perform memory transfers directly between the adapter and processor cache, and reduce the code path length between the adapter and the processors. This reduction in the path length might occur because the adapter is not operating as a traditional I/O device, and there is no device driver layer to perform processing. CAPI also presents a simpler programming model.

The implementation of CAPI on the IBM POWER processor enables hardware companies to develop solutions for specific application demands. Companies use the performance of the IBM POWER processor and NVIDIA Tesla GPU for computing, and the adapter for a network interconnection such as InfiniBand by using a CAPI interface with a simplified programming model and efficient communication with the processor and memory resources.

Although CAPI is part of IBM POWER processors and is IBM intellectual property (the Processor Service Layer (PSL)), several industry solutions benefit from the mechanism of connecting different devices to the processor with low latency, including memory attachment. The PCIe standard is pervasive in processor technology, but its design characteristics and latency do not allow the attachment of memory for load/store operations.

Therefore, the IBM OpenCAPI™ Consortium was created, with the goal of defining a device attachment interface to open the CAPI interface to other hardware developers and extending its capabilities. OpenCAPI aims to allow memory, accelerators, network, storage, and other devices to be connected to the processor through a high-bandwidth, low-latency interface to become the interface of choice for connecting high-performance devices.

The design of OpenCAPI enables low latency when accessing attached devices (nearly in the same range of system memory access), which enables memory to be connected through OpenCAPI and serve as main memory for load/store operations. In contrast, PCIe latency is 10 times bigger. Therefore, OpenCAPI is a significant enhancement compared to traditional PCIe interconnects.

2.2 InfiniBand

InfiniBand is an open set of interconnect standards and specifications. The main InfiniBand specification is published by the InfiniBand Trade Association (IBTA).

The InfiniBand Architecture (IBA) is an industry-standard architecture for server I/O and inter-server communication. It was developed by the IBTA to provide the level of reliability, availability, performance, and scalability that is necessary for present and future server systems with levels better than can be achieved by using bus-oriented I/O structures.

InfiniBand is based on a switched fabric architecture of serial point-to-point links, where InfiniBand links can be connected to either host channel adapters (HCAs), which are used in servers, or switches. It can scale to meet the increasing bandwidth demands of today’s workload performance.

InfiniBand is a solution that ranges from the hardware to the application layer. It is developed by the OpenFabrics Open Alliance. It is an open industry-standard specification and independent of the host operating system and hardware platform. Here are some of its benefits:

- Low latency
- Simplified management
- High bandwidth
- Quality of service (QoS) enabled
- Scalable
- Supports CPU offloading
- Remote Direct Memory Access (RDMA)
- Lossless link level flow control

The next sections describe the technology, topologies, hardware components, and tools of InfiniBand.

2.2.1 Technology

Performance is one of the important metrics of the InfiniBand technology, and it depends on bandwidth and latency. The numbers that are used in this section are examples.

The bandwidth of an InfiniBand connection is measured in bits per second, and is known as the *link rate*, which is calculated by the following equation:

Link speed x Link width x Encoding factor = Link rate

The link speed of InfiniBand has increased over the years with each new generation of the technology that is released. A summary of this progress is shown in Table 2-1.

Table 2-1 InfiniBand link speed

Generation	Links speed	Year
Quad Data Rate (QDR)	10 Gbps	2007
Fourteen Data Rate 10 (FDR10)	~ 10 Gbps	2011
Fourteen Data Rate (FDR)	~ 14 Gbps	2011
Enhanced Data Rate (EDR)	~ 25 Gbps	2014
High Data Rate (HDR)	~ 50 Gbps	2017
Next Data Rate (NDR)	To be determined, probably ~ 100 Gbps	Not yet available
Extended Date Rate (XDR)	To be determined, probably ~ 250 Gbps	Not yet available

The link width of a InfiniBand connection is the number of lanes (2 for send, 2 for receive) inside a cable. The number of possible lanes vary, such as 1, 4, or 12, but currently 4 is used in production.

Figure 2-1 shows a schema of the InfiniBand cable width.

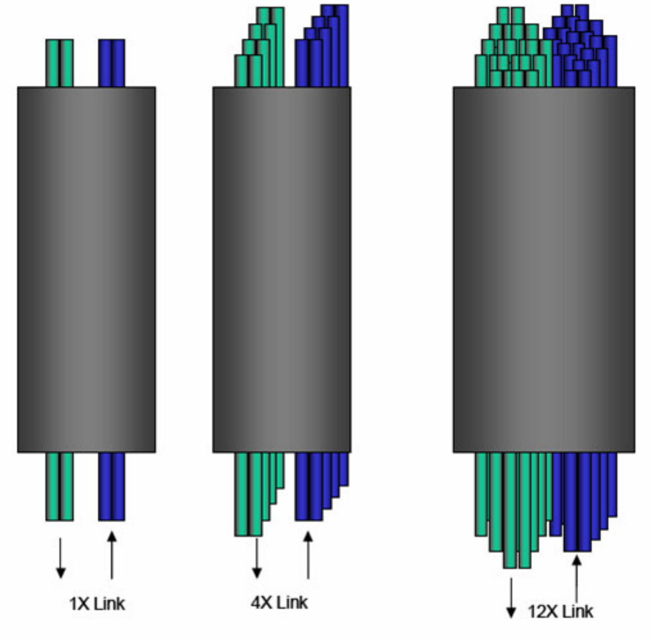


Figure 2-1 InfiniBand cable widths

The encoding factor describes the relationship between the size of the whole InfiniBand package and the actual size of the payload data. It is calculated by the following equation:

$$\frac{PayloadSize}{PackageSize}$$

For example, if you use the InfiniBand technology generation FDR10 (2011), the encoding factor is calculated as follows:

$$\frac{64Bits}{66Bits} = 0,\overline{96}$$

The result is close to 0,97, and in some scenarios it can be disregarded for performance calculation. In newer technology generations, such as EDR, the link speed is increased so that the link rate reamained the same.

Table 2-2 provides the available and future link rates for InfiniBand connections.

Table 2-2 Generations of InfiniBand and key metrics

Generation	Link speed	Link width	Encoding factor	Link rate
QDR	10 Gbps	4	8/10	32 Gbps
FDR10	~ 10 Gbps	4	64/66	40
FDR	~ 14 Gbps	4	64/66	~ 54 Gbps

Generation	Link speed	Link width	Encoding factor	Link rate
EDR	~ 25 Gbps	4	64/66	100 Gbps
HDR100	~ 50 Gbps	2	64/66	100 Gbps
HDR	~ 50 Gbps	4	64/66	200 Gbps
NDR (TBD)	Probably ~ 100 Gbps	4	prob. ~ 64/66	~ 400 Gbps
XDR (TBD)	Probably ~ 250 Gbps	4	prob. ~ 64/66	~ 1 Tbit/s

Starting with FDR10, there is an increase in the Link rate in relation to the link speed. Another thing to mention is the HDR100. It is an HDR link with a width of 4 divided, by a Y-cable, into two links of two link width each.

Latency is the time that it takes for a packet to get from source to destination. It is also improved from each generation to the next. Table 2-3 shows the latency decreasing over the years with each new technology.

Table 2-3 Latency versus the advance of technology

Generation	Latency	Year
QDR	1.3 µs	2007
FDR10	0.7 µs	2011
FDR	0.7 µs	2011
EDR	0.61 µs	2014
HDR	0.6 µs	2017

Eventually, latency cannot be reduced any further because nothing is faster than light. The length of the cable is at least one of these factors.

Other aspects of InfiniBand are as follows:

Remote Direct Memory Access RDMA helps reduce processor impact by directly transferring data from sender memory to receiver memory without involving host processors.

CPU impact InfiniBand requires a minimal impact on CPU load while transferring data because it uses a protocol that is implemented in hardware, kernel-bypassing features, and RDMA support.

Scalability Scalability is an advantage of InfiniBand because it enables the system to scale out and add more nodes and switches to the infrastructure. The infrastructure is limited to 48.000 nodes in a single subnet.

Identifiers

The globally unique identifier (GUID) is a 64-bit unique persistent hardware address for nodes, ports, and systems. In switches, all switches in a chassis have the same system GUID. The local identifier (LID) is a 16-bit layer 2 non-persistent address, which is assigned by the subnet manager. In HCA, each port has its own LID. In switches, all ports of a switch have the same LID. The LID is needed for the routing within a subnet.

Package forwarding

Switches use a linear forwarding Table (LFT), which contains a Destination LID to Exit Port mapping for routing the packages. Service level (SL) to virtual lane (VL) mapping is used to have several VLs on the same port, and is necessary for QoS.

Link layer protocol

This layer is responsible for the package format and protocols for operations within a subnet. It contains information about the payload size, addressing, forwarding, flow control, and SL to VL.There are two types of packages: link management packets (link operation) and data packets (send, read, write, and ack). The maximum size of data in a package, the Maximum Transfer Unit (MTU), that can be transmitted is 256 - 4096 bytes (headers and CRCs are not included). With a large payload, less CPU impact and higher bandwidth are used for storage workloads. With a small payload, lower latency and no CPU impact are used for HPC workloads.

Transport layer protocol

This layer is responsible for the end-to-end connection between two applications. Their endpoints are called queer pairs (QPs). Before the transmission, the QP segments data into packages and reassembles it afterward. A QP consists of a send and a receive queue. A QP has a queue pair number. An application has direct access to its QPs. They are assigned to the applications virtual address space. For each connection, a QP is created. A QP is assigned to a service type. There are five different types of services. This layer determines the behavior of the connection, which can be Reliable Connection (RC), Unreliable Connection (UC), Reliable Datagram (RD), Unreliable Datagram (UD), or Dynamically Connected (DC).

Upper layer

This layer is responsible for various protocol support and application interfaces, such as the Message Passing Interface (MPI) protocol, RDMA protocol, IP over InfiniBand protocol, client interface for the transport layer, application interface for InfiniBand HCA / Fabric (Verbs), General Service Interface (GSI), and Subnet Management Interface (SMI). The InfiniBand Verbs can interact with the HCA, manage QPs, and work with sending and receiving requests.

Quality of service

QoS is available with InfiniBand, and is accomplished through the VL feature. There are up to 15 virtual links that are available within each physical link, and each is used for different bandwidth and QoS. Each VL has a different buffer and can be assigned to a priority.

Partitions

A partition is a set of nodes in the fabric that are separated from the other nodes. Nodes in different partitions cannot reach each other. One port of a node can be a member of several partitions concurrently. The PKEY identifies a partition.

Subnet manager

The subnet manager is software that is responsible for the InfiniBand fabric, including initialization, configuration, and routing. It is used to manage communication. The architecture defines a communication management scheme that is responsible for configuring and maintaining each of the InfiniBand fabric elements. Management schemes are defined for error reporting, link failover, chassis management, and other services to ensure a cohesive communication environment.

2.2.2 Topology

There are several different types of topologies that you can use as part of a network, but there are only a few that are useful. For common workloads, it is important that the topology is homogeneous and balanced. In this section, the following five topologies are mentioned:

- Fat-tree
- Dragonfly
- Mash (two, three, and so on dimensions)
- Torus (two, three, and so on dimensions)
- Hypercube

Fat-tree

The fat-tree topology is the most common one for InfiniBand. One of the advantages is limitless scalability. Fat-tree is separated into levels. The switches on the upper levels are called *spine switches*. The switches on the lowest level are called *leaf switches*, and the nodes are always connected to them. The smallest fat-tree topology has at least two levels.

Figure 2-2 describes this topology.

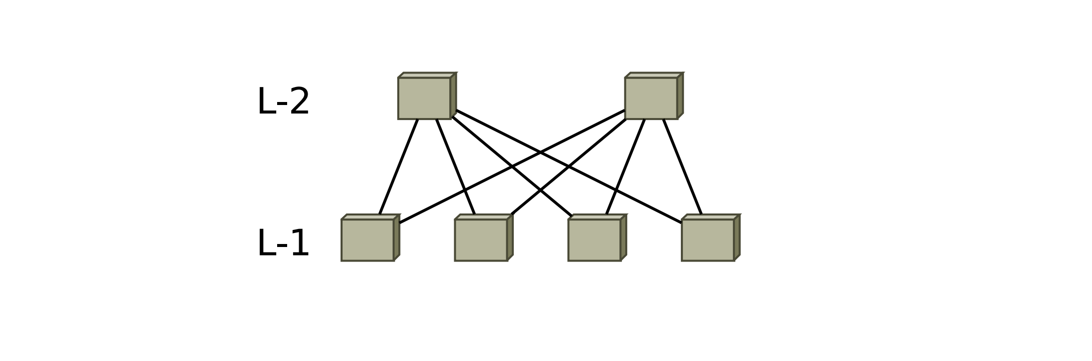


Figure 2-2 The fat-tree topology

As shown, each node from the L-1 is connected to each node in L-2. This topology makes your networks homogeneous. This topology provides a fully non-blocking network if the uplinks have the same throughput as the links that are going to the nodes.

Figure 2-3 shows a three-level fat-tree topology that is a hierarchical architecture. The nodes on the L-1 are divided into two groups. A group is defined as an island.

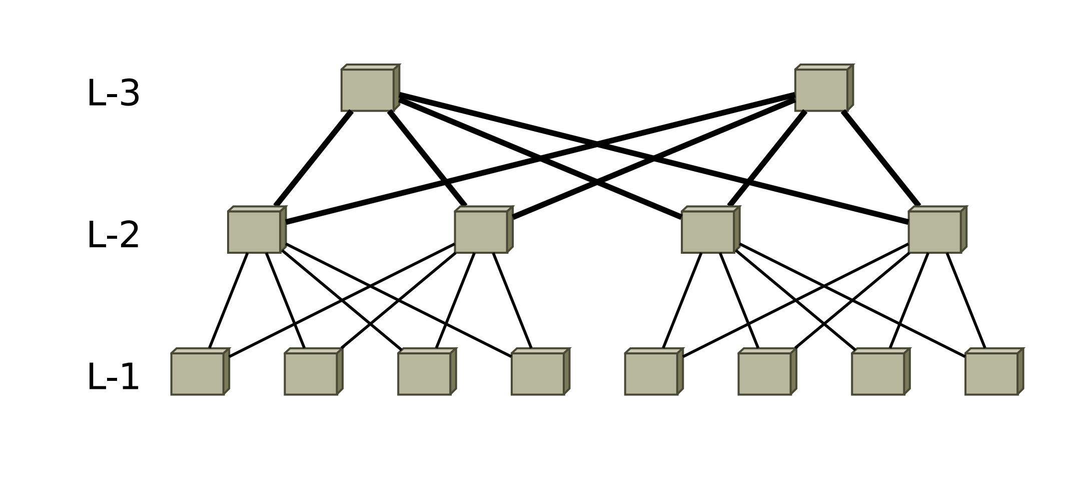


Figure 2-3 A three-level fat-tree topology

Up/Down (UpDn) and fat-tree are the most commonly used InfiniBand routing algorithms. They provide a level of routing to adapt the load according to fabric necessities. UpDn uses the shortest path available between access switches. In the fabric, all switches participate and loop-free routing is ensured.

With the UpDn routing algorithm, the latency does not increase in the L-2 subtree; if L-3 is needed to reach the destination, an extra hop increases the latency.

Fat-tree also offers the best performance when it is configured as *non-blocking*, which means that there are the same number of cables coming in from the nodes to the access switches as are going to the spine switches. For example, on a 40-port 1U access switch, you might use 20 cables coming from either 20 nodes or 10 nodes (two per node), which enables 20 cables to two different spine switches (10 to each one for redundancy and performance). The blocking factor in this case is 1:1.

Sometimes, this configuration is not possible, and the blocking factor changes, which leads to over-subscription. Over-subscription occurs when the applications on the cluster allow it to happen or when technical requirements can be met. Normally, inside an island, which is a grouping of nodes that serves a purpose or are in the same rack or line of racks, a fat-tree topology is always non-blocking. When you interconnect the islands, over-subscription comes in different forms, sometimes (for example) 1:2, 1:3, 1:4, and so on, which means that, using the previous example of the 40-port 1U access switch, that for every 30 ports coming from the nodes to the access switches, only 10 go to the spine switches.

Dragonfly

The Dragonfly topology is almost the same as the fat-tree topology, but it differs at the higher level. Instead of adding another row of switches, there are only cable connections.

Figure 2-4 shows a Dragonfly topology.

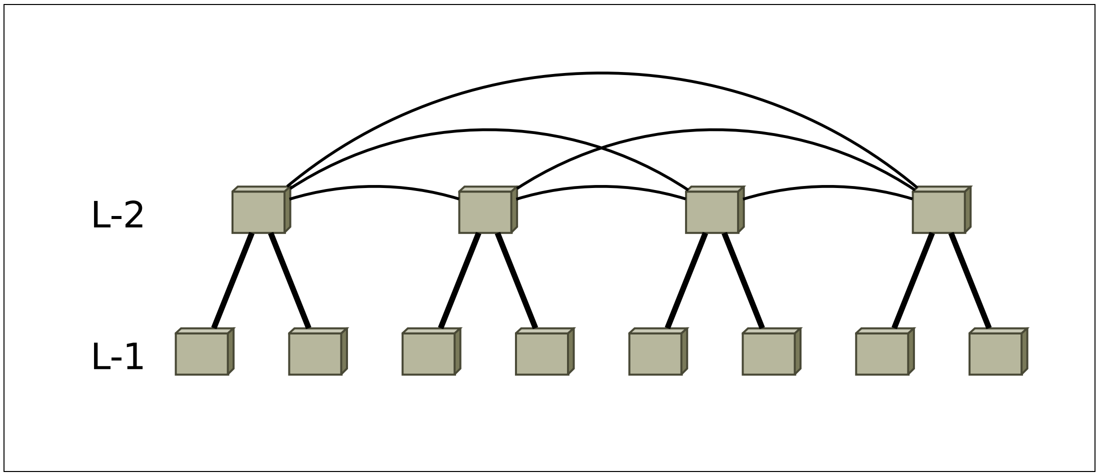


Figure 2-4 The Dragonfly topology

If you have more than three top-level switches in the Dragonfly topology, it is not fully non-blocking because the number of uplinks to a specific subtree is smaller than the links that that are going down.

Mesh (two or three dimensions)

Figure 2-5 shows 2- and 3-dimensional mesh topologies. Each switch is connected to its direct neighbor. On a 2-dimensional mesh, a switch needs up to four connections, and with a 3-dimensional mesh, the connections increase up to six.

A multidimensional mesh is possible, but with each dimension the uplink is increased by 2. This topology makes sense only for some appropriated applications, and is not recommended for larger scales because the number of hops increases.

.

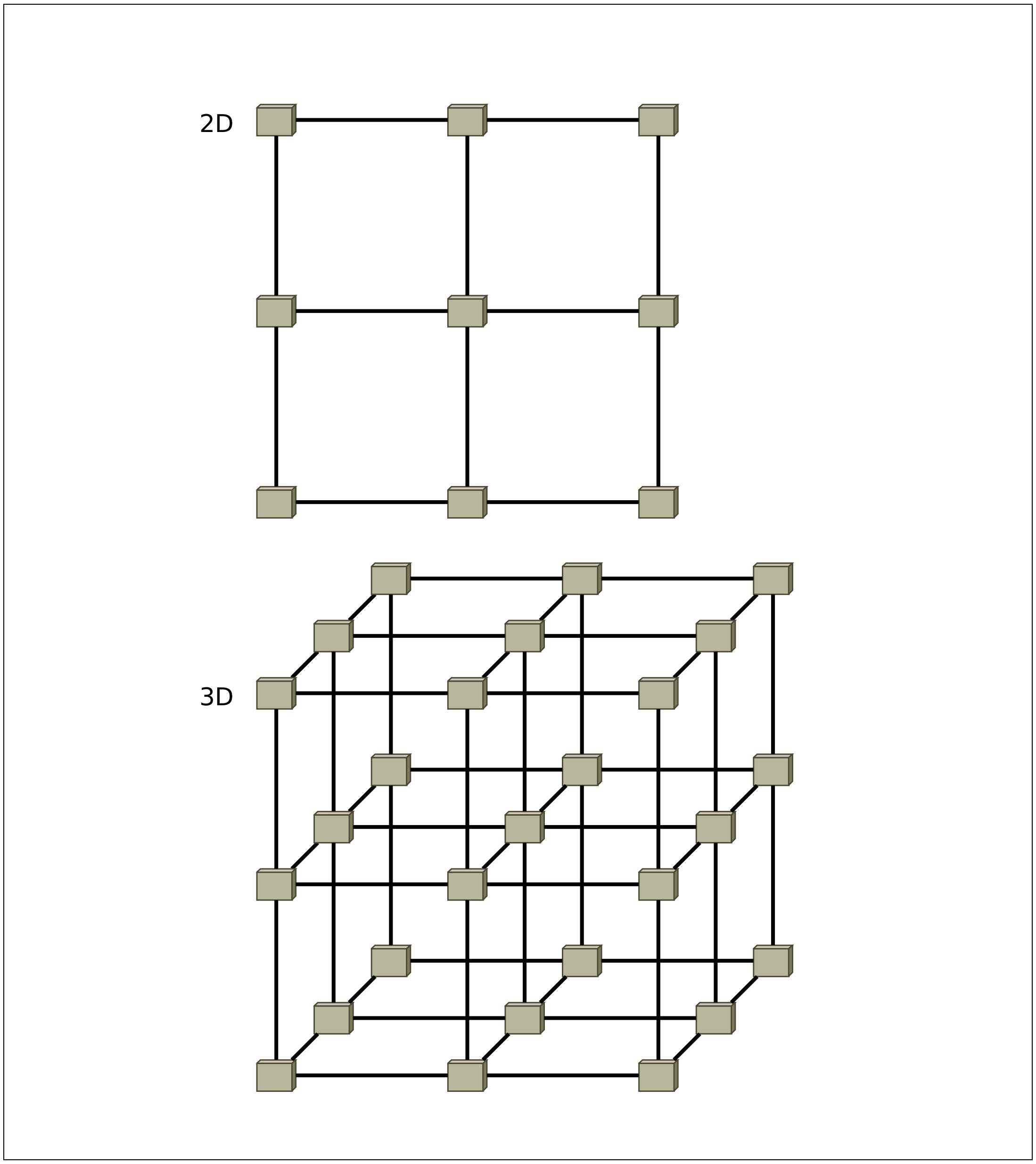


Figure 2-5 Mesh topologies

Torus (two or three dimensions)

A torus topology is similar to the mesh topology, with the addition that the ends of a corner are connected to the corner at the other end.

Figure 2-6 shows 2- and 3-dimensional Torus topologies. Like the mesh topology, it is preferable for only specific applications, and there are some concerns about scalability.

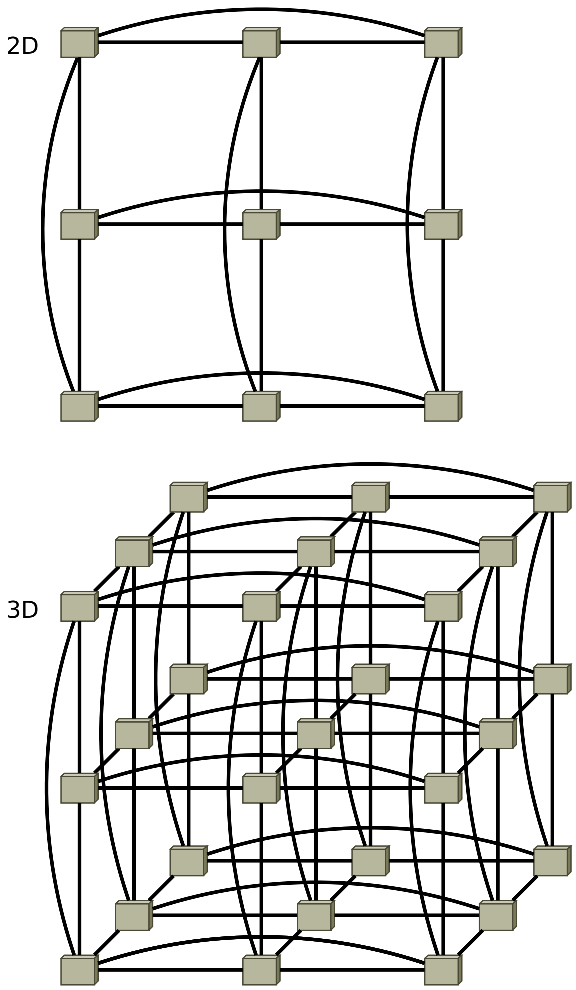


Figure 2-6 Torus topologies

2.2.3 Hardware components

There are three common network hardware types for InfiniBand: switches, HCAs, and cables.

Switches

Switches are connected inside a network to forward packages from one node to another node. InfiniBand and Ethernet switches are based on application-specific integrated circuit (ASIC) technology. Depending on the generation of the technology, each ASIC has a specific number of connection ports with the same link rate. There are no special uplink ports, as on Ethernet, with a different link speed. There are two different types of switches that are available for InfiniBand:

Edge switches Edge switches are small device, for example, one unit of a standard rack. The internal topology of a switch is non-blocking and has bidirectional bandwidth. There are unmanaged or managed switches.

Director switches Director switches are a non-blocking modular chassis consisting of several ASICs that are also used in edge switches. They offer high port density with high bandwidth.

Host channel adapters

InfiniBand adapters are called HCAs and connect to an PCIe slot. The newer generations together with IBM POWER are CAPI-enabled and lower latency is possible. There are HCAs with one and two ports that are available. The supported link speed relies on the technological generation: All adapters should provide compatibility with earlier adapters. For each generation, there might be several adapters with different functions, such as the support of extra Ethernet or exotic InfiniBand functions (lower ring-latency).

Cables

There are two different types of cables for each generation of the technology.

Passive copper cables Passive copper cables use Quad Small Form-factor Pluggable (QSFP)-like connectors that are 2 - 7 meters, depending on the link rate.

Active optical cables Active optical cables use QSFP connectors that are 100 - 300 meters, depending on the link rate.

2.2.4 Tools

Several versions of the InfiniBand hardware and software stack have been made available. The need for a standardized stack led to the creation of the OpenIB Alliance, now known as the OpenFabrics Alliance, which created the Open Fabrics Enterprise Distribution (OFED). It is the most common software distribution for InfiniBand. The OFED stack also includes several higher-level protocols that are not part of the initial IBTA specification.

The following list describes the most common command-line tools of the OFED.

`ibstat` Shows information about the local InfiniBand interface.

`ibstatus` Almost the same as **`ibstat`**, but with other metrics, for example, the LID is shown in hexadecimal.

`ibping` Like the **`ping`** command, but must be run on both the server and client. On the server, run **`ibping -S`**, and on the client, run **`ibping -L <LID>`**.

`ibtracert` Traces the path between 2 ports that are identified by their LIDs.

ibportstat Shows the logical link and physical port states. Can change the port status.

ibv_devices Lists InfiniBand devices and the node GUIDs of them.

ibaddr Displays GID and LID information about the local node/ports.

ibswitches Shows all the switches in the fabric and their GUIDs, names, ports, and LIDs.

ibhosts Shows all the hosts in the fabric and their GUIDs, names, and ports.

ib_write_lat Runs an RDMA write latency test.

ib_read_lat Runs an RDMA read latency test.

ib_send_lat Runs an RDMA read and write (send) latency test.

ib_write_bw Runs an RDMA write bandwidth test.

ib_read_bw Runs an RDMA read bandwidth test.

ib_send_bw Runs an RDMA read and write (send) bandwidth.

ibdump Dumps the traffic of an InfiniBand host, much like **tcpdump**, in the **pcap** format.

sminfo Shows information about the subnet manager.

smpquery A tool for querying the subnet manager for some attributes.

saquery Shows information about the nodes in the fabric.

ibroute Shows all LIDs that are reachable from a switch LID.

ibtracert Shows a path between two LIDs.

ibportstate Shows information about and configures an InfiniBand port.

iblinkinfo Shows connectivity information about all nodes in the fabric.

ibnetdiscover Similar to **iblinkinfo**.

ibdiagnet Runs a diagnostic on the complete subnet.

topodiff Records the subnet topology in a topology file.

2.3 Ethernet

Ethernet is the preferred method of interconnecting devices in the network because it is accessible, there are several providers available, it is easy to configure, easy to troubleshoot, easy to operate, and it performs well. The focus of this section is to review the use of this technology in an HPC environment for:

- Administration network
- OOB management
- Ethernet on the fabric with RoCE

2.3.1 Administration network

The costs for setting up the fabric are high in an HPC environment because the entire performance of the cluster depends on it. For the administration network, you do not need 100 Gbps ports because most of the traffic will be, for example, to update or install the operating system in the nodes, troubleshoot problems, start or restart nodes, and monitor the switches. Every device in the HPC cluster is connected to this administration network, so larger clusters have much equipment to administer. Each IBM POWER 9 server comes with two built-in 1 Gbps RJ45 ports for management, and so do some of the InfiniBand switches that are connected to this network (other switches have only one 1 Gbps port).

The administration network is not limited to management if it has the correct level of security. If so, you can use it for other operations, such as wide area network (WAN) access, bridging to the production network, and other functions.

2.3.2 Out-of-band management network

The idea behind the OOB network is a dedicated infrastructure that performs certain tasks that are important on every cluster for both the nodes and the fabric, such as:

- Perform a system start, stop, or restart.
- Diagnose the level of utilization in the cluster.
- Manage system performance.
- Monitor every node in the network.
- Upgrade the firmware when needed.
- Monitor the power supplies and fans.
- Monitor interface utilization.
- Monitor the status of the ports and other interfaces on the switches.
- Administer the management logs of the InfiniBand switches.

Most of the tools operate in this network and can give you information about problems in nodes. Through this network, you access the cluster for troubleshooting. If one of the switches fails, you can recover it through the management network if it is not because of a hardware problem.

2.3.3 Ethernet on the Fabric with RDMA over Converged Ethernet

Connectivity inside the cluster, specifically in the fabric, must perform at high speeds. When you choose the technology that you use in the fabric, there are different scenarios or approaches, including InfiniBand, Ethernet, Aries, and Omni-Path. There are HPC systems running over Ethernet networks, but the most common fabrics for HPC are InfiniBand and Ethernet because they both account for a significant percentage of the world's HPC systems.

Regarding the speeds that are supported for the fabric, many vendors offer 100 Gbps switches with a lower price than other proprietary solutions, which makes Ethernet an attractive solution because it works on almost any environment. For example, on storage solutions, there is a transition from Fibre Channel only networks to Fibre Channel over Ethernet (FCoE) solutions that converge the infrastructure.

Inside an HPC system that uses Ethernet for the fabric, the most common protocol for communication is RDMA, which provides direct memory access from the memory of one node to the memory of another node without going through either the CPU or the kernel layer stack. The CPU does not become as involved as with common network workload, and can be used for other tasks. There is little latency in the communication and performance is improved.

RoCE was developed so that nodes can communicate on an Ethernet fabric. There are two versions:

- RoCEv1 is a link layer protocol (L2) that has the limitation that the protocol works only on the same broadcast domain.
- RoCEv2 has more focus on Layer 3 (L3) because it runs on UDP port 4791 for IPv4 and IPv6, which means that it can work on any low-latency, high-bandwidth switch and be routable across the network. RoCEv2 is not recommended if there are no low-latency routers.

2.4 Storage

Shared storage architectures and their performance rely on the interconnect network. The following storage solutions are commonly used for HPC clusters that use IBM Power Systems servers.

2.4.1 IBM Spectrum Scale

IBM Spectrum™ Scale, formerly known as IBM General Parallel File System (IBM GPFS™), is a high-performance, shared-disk file management solution that provides fast, and reliable access to data from multiple servers. Applications can readily access files by using standard file system interfaces, and the same file can be accessed concurrently from multiple servers and protocols.

It provides high availability through advanced clustering technologies, dynamic file system management, and data replication. If there are server or cluster malfunctions, it can continue to provide data access. Its scalability and performance are designed for data-intensive applications, such as data mining, data analytics, seismic data processing, scientific research, and scalable technical computing.

It is supported on IBM Power Systems servers and other systems that use Linux, and several other operating systems. The software package provides simplified data management and integrated information lifecycle tools that can manage petabytes of data and billions of files to address the growing cost of managing growing amounts of data.

Some of the key benefits of IBM Spectrum Scale™ are:

- Provides a modern scale-out architecture.
- Access file data directly by using NFS, SMB, Object, and Hadoop.
- Has virtually limitless performance and capacity scaling.
- Has automated data placement and data migration.
- Uses less expensive commodity storage.
- Integrates with IBM Spectrum Archive™ and IBM Spectrum Protect™.
- Has a long history of proven reliability across multiple industries.

Here are some key parameters of IBM Spectrum Scale:

Strengths IBM Spectrum Scale provides a global namespace, shared file system access among IBM Spectrum Scale clusters, simultaneous file access from multiple nodes, high data protection and availability through replication, the ability to make changes while a file system is mounted, and simplified administration even in large environments. It also supports RDMA for copying files with a low latency over the interconnect network.

Basic structure IBM Spectrum Scale is a clustered file system that is defined over one or more nodes. Each node in the cluster consists of three basic components: administration commands, a kernel extension, and a multithreaded daemon. On the Linux operating system, there is an extra portability layer that is needed.

Scalability IBM Spectrum Scale has many features beyond common data access, including data replication, policy-based storage management, and multi-site operations. Multiple clusters can share data within a location or across WAN connections.

For more information, see An Introduction to IBM Spectrum Scale.

2.4.2 IBM Elastic Storage Server

IBM Elastic Storage™ Server is a modern implementation of software-defined storage (SDS), which combines IBM Spectrum Scale software with IBM POWER processor-based I/O-intensive servers and dual-ported storage enclosures. IBM Spectrum Scale is the parallel file system at the heart of IBM Elastic Storage Server. IBM Spectrum Scale scales system throughput as it grows while still providing a single namespace, which eliminates data silos, simplifies storage management, and delivers high performance. By consolidating storage requirements across your organization onto IBM Elastic Storage Server, you can reduce inefficiency, lower acquisition costs, and support demanding workloads.

IBM Elastic Storage Server supports high-speed Ethernet and InfiniBand I/O adapters to interconnect to the HPC fabric.

The following server features are provided:

IBM Spectrum Scale RAID
Enables the rebuilding of failed disks in minutes instead of hours by using declustered RAID. Requires less raw capacity by using erasure coding instead of replication for data protection.

Unified storage
Consolidates storage for file, object, and big data workloads into a single storage pool. Enables new applications by using object or HDFS interfaces to benefit from the simplicity of file data management.

True scale-out capability
Scales performance, capacity, files, and objects linearly with automatic load balancing. Avoids network-attached storage (NAS) performance bottlenecks by using a parallel file system.

Hadoop connector
Hadoop and Spark applications run natively by using HDFS application programming interfaces (APIs). IBM Elastic Storage Server enables faster, more consistent data analytics by eliminating data movement and conversion into dedicated Hadoop storage. It combines multiple data sources into a single HDFS view as needed by the business.

Policy-based tiering and compression QoS
Tiers data automatically between various tiers of storage, including tape and cloud, based on policies and reduces TCO by eliminating the need to purchase excess, high-performance storage.

Integrated & modular
Enables quick deployment of the initial solution and extra blocks of storage as demands increase. IBM Elastic Storage Server is designed for high availability and balanced to maximize throughput. Choosing IBM Elastic Storage Server avoids the risks and complexity of other options and provides a single source of global support for hardware and software.

2.5 Applications

HPC and AI applications rely on the complete infrastructure. An interconnect network is a critical performance factor because of its usage of massive parallel input/output transactions. Although bandwidth is the most common metric, latency is in most use cases the critical part. Also, the simplicity of the programming model is worth describing. This section points out two different application basics for HPC and AI.

2.5.1 Message Passing Interface and other RDMA libraries

MPI is a communication protocol for technical computing. It is an industry standard for parallel communications in the HPC community. Its implementation covers peer-to-peer and broadcast messages, and is typically provided as a library. One of the advantages is portability because the definition is generic. There are implementations in different popular programming languages that are available, for example, C, C++, FORTRAN, Java, and Python. To send data simple and fast through the interconnect network, MPI uses RDMA.

IBM Spectrum MPI is a high-performance, production-quality implementation of MPI. It accelerates application performance in distributed computing environments. It provides a familiar, portable interface that is based on the open source MPI. It goes beyond Open MPI and adds some unique features of its own, such as advanced CPU affinity features, dynamic selection of interface libraries, superior workload manager integration, and better performance. It supports a broad range of industry-standard platforms, interconnects, and operating systems, ensuring that parallel applications can run almost anywhere.

A non-optimized environment can hinder competitiveness and slow time to results. Rather than have the application handle architectural differences in your infrastructure, IBM Spectrum MPI manages them. It eliminates the need to write multiple versions of the application to account for different interconnects. It also optimizes application performance by improving collective algorithms.

The software features an Open MPI implementation for HPC parallel applications with improved performance and scalability. It is supported on IBM Power Systems and brings a collective MPI library and point-to-point communications protocol (Parallel Active Messaging Interface (PAMI)) back end that provides improved network connectivity and enhancements to application developer usability. It maximizes network efficiency by dynamically selecting the optimal network connection between each node at run time.

It also delivers an improved, RDMA-capable PAMI by using InfiniBand with OFED on IBM POWER hardware. It also offers a superior collective library that supports the seamless use of GPU memory buffers for the application developer. The library features advanced logic to determine the fastest algorithm for any given collective operation.

2.5.2 IBM PowerAI

IBM PowerAI is a package of software distributions for many of the major deep learning software frameworks for model training, such as TensorFlow, Caffe, Chainer, Torch, Theano, and their associated libraries, such as cuDNN and nvCaffe. These are extensions that take advantage of accelerators, for example, nvCaffe is the NVIDIA extension to Caffe to work on GPUs. As with nvCaffe, IBM has an own extension to Caffe, which is called IBM Caffe.

The IBM PowerAI solution is optimized for performance by using the NVLink-based IBM POWER servers for HPC. The stack also comes with supporting libraries, such as DIGITS, OpenBLAS, Bazel, and, NCCL.

Why IBM PowerAI simplifies adoption of deep learning

Today, organizations use deep learning to develop powerful, new analytic capabilities that span multiple usage patterns, from computer vision and object detection, improved human computer interaction through natural language processing, to sophisticated anomaly detection capabilities. At the heart of any use case that is associated with deep learning are sophisticated pattern recognition and classification capabilities, which serve as the birthplace for revolutionary applications and insights of the future.

However, in situations where organizations try to expand their area for deep learning or to start working on the development of deep learning, there are enormous difficulties, especially the performance issues that are caused by hardware limitations and time-consuming processes in each framework for setup, tuning, upgrades, and so on.

In the fourth quarter of 2016, IBM announced a revamp of IBM PowerAI, seeking to address some of the bigger challenges facing developers and data scientists. The goals were to reduce the time that is required for AI system training, making and running a snap with an enterprise-ready software distribution for deep learning and AI, and simplifying the development experience. The idea behind IBM PowerAI is to embrace and extend the open source community, embrace and extend the capability and creativity that is happening there, and add IBM unique capabilities. This revamp manifests itself in a number of value differentiators for AI applications that IBM Power Systems brings to the table.

IBM PowerAI provides the following benefits:

- Fast time to deploy a deep learning environment so clients can get to work immediately:
  - Simplified installation in usually less than 1 hour
  - Precompiled deep learning libraries, including all required files
- Optimized performance so users can capture value sooner:
  - Built for IBM Power Systems servers with NVLink CPUs and NVIDIA GPUs, delivering performance unattainable elsewhere
  - IBM PowerAI Distributed Deep Learning (DDL), taking advantage of parallel processing
- Designed for enterprise deployments: Multitenancy supporting multiple users and lines of business (LOBs)
- Centralized management and monitoring by integrations with other software: IBM service and support for the entire solution, including the open source deep learning frameworks

IBM PowerAI Distributed Deep Learning

IBM PowerAI DDL is a software/hardware co-optimized DDL system that can achieve near-linear scaling up to 256 GPUs. This algorithm is encapsulated as a communication library called DDL that provides communication APIs for implementing deep learning communication patterns across multiple deep learning frameworks.

The inherent challenge with DDL systems is that as the number of learners increases, the amount of computation decreases at the same time the amount of communication remains constant, which results in unwanted communication ratios.

To mitigate the impact of this scaling problem, IBM PowerAI DDL uses an innovative multi-ring communication algorithm that balances the communication latency and communication impact. This algorithm adapts to the hierarchy of communication bandwidths, including intranode, internode, and interrack, within any system. This implementation enables IBM PowerAI DDL to deliver the optimal DDL solution for a given environment.

The current implementation of IBM PowerAI DDL is based on IBM Spectrum MPI because IBM Spectrum MPI itself provides many of the required functions, such as scheduling processes and communication primitives in a portable, efficient, and mature software infrastructure. IBM Spectrum MPI specifically provides functions and optimizations to IBM Power Systems and InfiniBand network technology.

IBM PowerAI DDL objects() are the foundation of IBM PowerAI DDL. Each object() instance combines both the data and relevant metadata. The data structure component itself can contain various payloads, such as a tensor of gradients. The metadata provides a definition of the host, type of device (GPU and others), device identifier, and the type of memory where the data structure is. All of the functions within IBM PowerAI DDL operate on these objects, which are in effect synonymous with variables.

In a 256 GPU environment, ~90 GB of data must be transmitted to perform a simple reduction operation, and the same amount of data must be transmitted to copy the result to all the GPUs. Even a fast network connection with a 10 GBps transfer rate, bringing this data to a single parameter server can take 9 seconds. IBM PowerAI DDL performs the entire reduction and distribution of 350 MB in less than 100 ms by using a communication scheme that is optimized for the specific network topology.

A key differentiator for IBM PowerAI DDL is its ability to optimize the communication scheme based on the bandwidth of each available link, topology, and latency for each segment. In a heterogeneous environment, with multiple link speeds, IBM PowerAI DDL adjusts its dimensions to use the fastest available link, thus preventing the slowest link from dominating the run time.

Using IBM PowerAI DDL on cluster of 64 IBM S822LC for High Performance Computing (HPC) systems, we demonstrate a validation accuracy of 33.8% for Resnet-101 on Imagenet 22k in ~7 hours. For comparison, Microsoft ADAM 1 and Google DistBelief 2 were not able to reach 30% validation accuracy for this data set.

For more information about IBM PowerAI, see *IBM PowerAI: Deep Learning Unleashed on IBM Power Systems Servers*, SG24-8409.

3

Designing a solution

High-performance computing (HPC) systems are available in various sizes. These solutions can start with a couple of nodes and scale out to hundreds or even thousands of nodes.

This chapter show examples of HPC systems in three different sizes to illustrate the process of their design and scaling. The examples describe solutions for small, medium, and large systems.

3.1 Interconnect and data networks

The main approach to designing an HPC or artificial intelligence (AI) system data network is to provide a high bandwidth or low latency path for nodes to interconnect, and a reliable and scalable solution. The following three examples contain information about the fabric design to support nodes that were sized based on typical application needs.

3.1.1 Small solution

This example has two stages:

- Implement a basic HPC or AI system.
- Scale the cluster.

Technical details about the nodes are not part of the description, although the necessary calculations were done to confirm that this is a viable solution.

Initially, a single InfiniBand switch is used to interconnect the nodes and create a small design. This first stage includes 10 nodes that are interconnected through a 40-port 1U InfiniBand switch.

Architecture overview

The design includes the following hardware components:

- Two management nodes
- Six compute nodes
- One login node
- One storage node
- One InfiniBand 40-port 1U switch
- One Ethernet switch for management of the components
- One Ethernet switch for the storage network

The overview shows that all compute nodes of the HPC/AI systems cluster are connected to the InfiniBand switch, the storage connection uses Ethernet for data access, and you use the management network for system administration and out-of-band (OOB) access.

This architecture is shown in Figure 3-1.

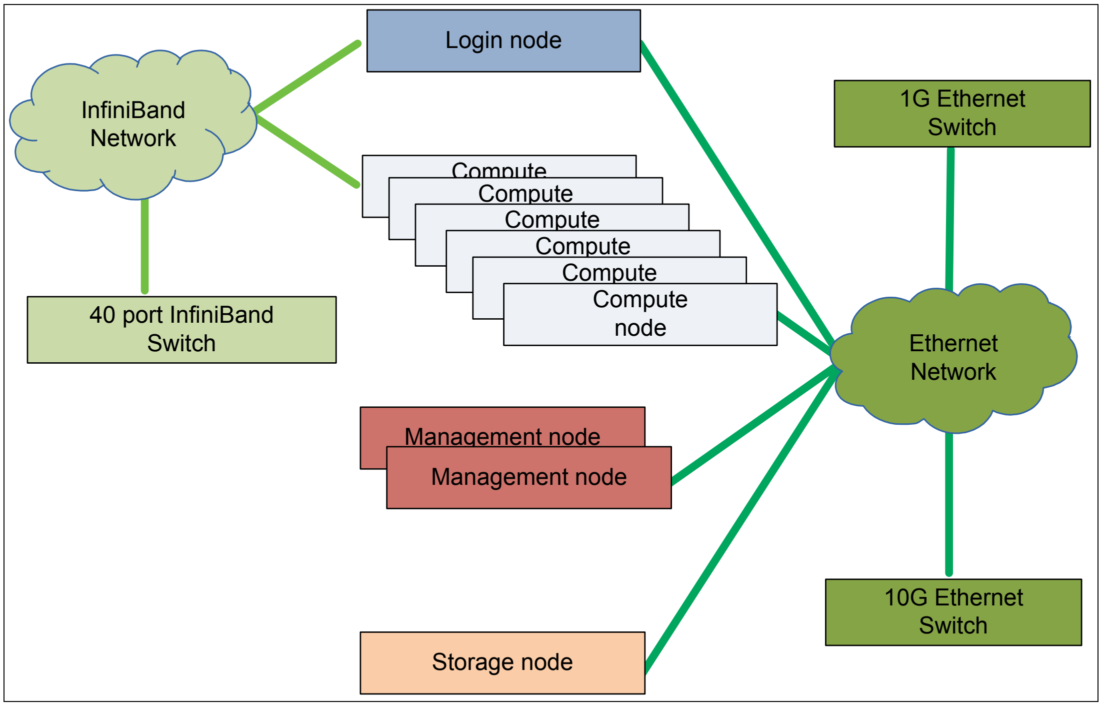


Figure 3-1 Small solution architecture

In this example, a 10 Gb Ethernet switch connects the storage node. An advantage is that an external storage system can be attached over that network to the cluster. This example demonstrates an option of how an existing enterprise storage solution can be combined with an InfiniBand based architecture.

Network overview

The network diagram in Figure 3-2 shows how the devices are interconnected:

- The different types of servers, which are in the middle of the drawing.
- The InfiniBand Switch is on the left side of the figure, and it is connected to the login node and the compute nodes only.
- On the right side of the figure are the Ethernet switches. Hardware management uses the 1 Gb switch, and storage traffic uses the 10 Gb switch.

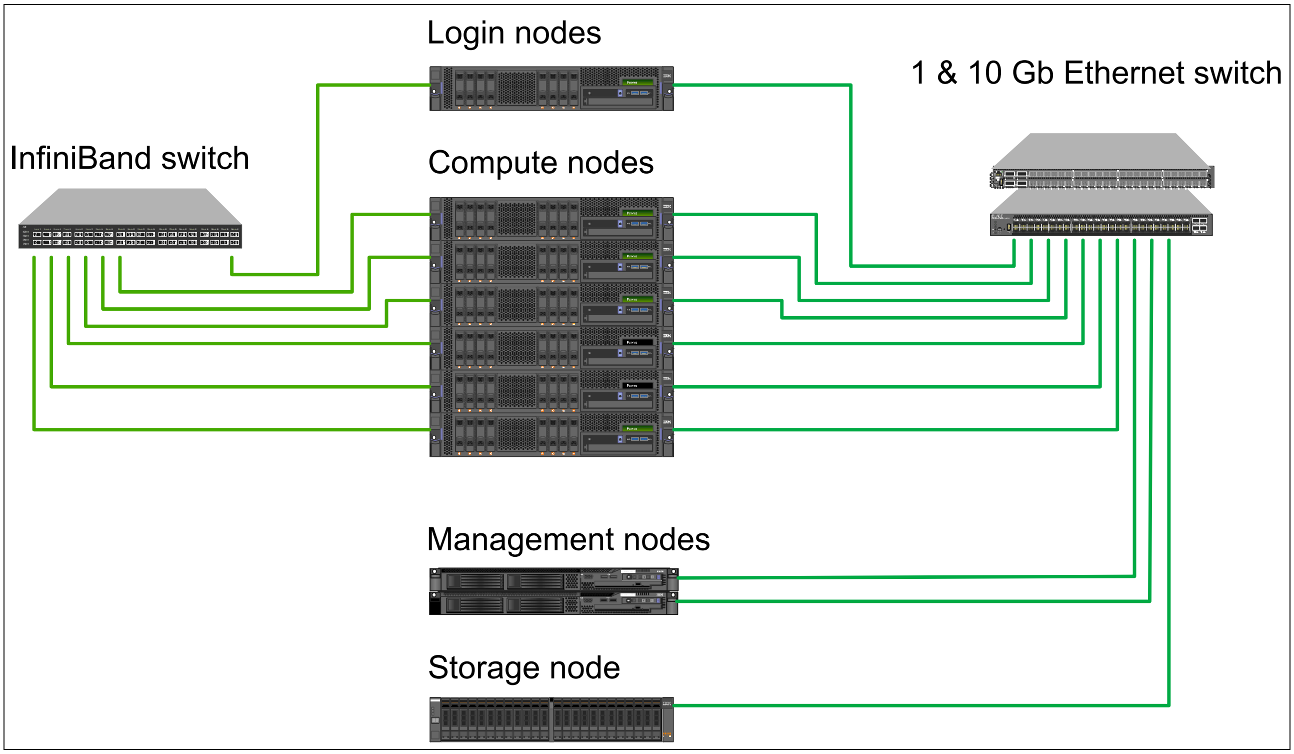


Figure 3-2 Small solution network diagram

Both images, the architectural overview and network diagram, should illustrate how different system connections are established.

Rack view

Figure 3-3 shows an option about how the devices can be installed inside the 42 Unit rack, which uses 21 rack units (RUs) of the available space. Note the location of the InfiniBand switch: It is not at the top or the bottom of the rack. To optimize the cable length, place the switch in the middle of the rack.

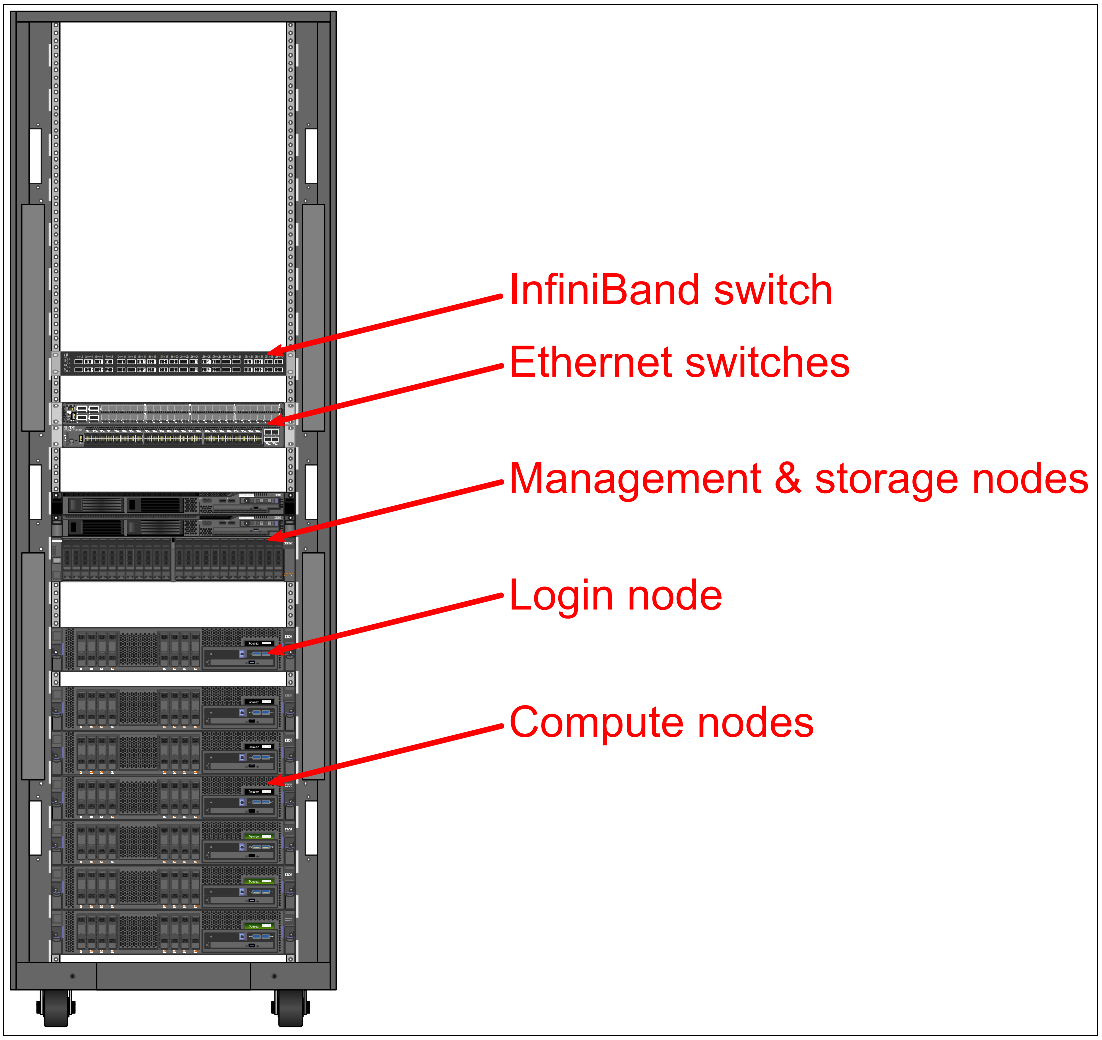


Figure 3-3 Small solution rack view

In this example, only 21 RU are used in the rack. The spaces between the nodes and switches are included in the figure for better visualization of the components.

3.1.2 Medium solution

Typically, over time the application requirements increase and the demand for more compute resources rises. Then, you must plan for the extension of the initial example. It is simple to add more compute nodes to the environment, but there are limits:

- Rack space capacity
- Floor space availability
- Cooling availability
- Power consumption

- InfiniBand topology
- Network resources
- Storage capacity and performance
- Application scaling capabilities

More servers can be placed into the available rack space if there are no environment limitations in terms of weight, power, and cooling. The network and InfiniBand switches are placed in the middle of the rack to optimize the cable length between the components within the rack. But, if one of these constraints cannot be met, then an extra rack is required. Then, it is necessary to plan carefully the cable routing between the racks to optimize the cable length and corresponding costs. Depending on the characteristics of the data center, cables can be managed above the rack in a Top Of Rack (ToR) tray, below the rack through a raised floor, or straight through the middle of the rack side wall if the mechanical design supports such a cable passage.

The next threshold in terms of growth is the number of switch ports. Typically, an InfiniBand switch has 36 or 40 (High Data Rate (HDR) and beyond) ports. After these ports are all used, a new topology must be established. In our example, a non-blocking fat-tree topology is used, which means that a leaf switch must provide the same number of ports (as uplinks) and compute nodes. So, up to 20 ports per leaf switch are available only for compute nodes. At least three leaf switches are required to increase the number of ports, compared to a single switch. In addition, two spine switches are necessary to interconnect the leaf switches.

Figure 3-4 illustrates a fat-tree fabric of five switches. That fabric provides 60 free ports on the lowermost leaf switches. To extent that fabric, a fourth leaf switch can be added to the 20 unused ports of the uppermost spine switches, which increases the number of compute ports to 80. Further extensions require more spine and leaf switches.

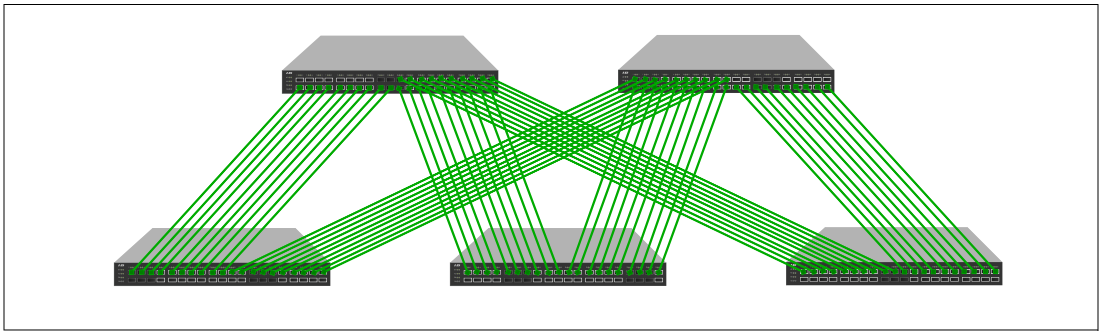


Figure 3-4 Five-switch networking fabric

This example shows that growing an HPC solution is not a linear effort, so you must be aware of the thresholds where extra costs appear.

In stage two of our small solution, we reengineered the network to support the new requirements. After we resized the new solution, we implemented the architecture that is described in “Architecture overview”.

Architecture overview

The enhanced design includes the following hardware components:

- Two management nodes for HA
- Fifty compute nodes
- Two login nodes
- Two storage nodes

- Five InfiniBand 40-port 1U switches
- Two shared storage servers
- Two switches for management

Figure 3-5 shows the architecture of a typical medium network architecture.

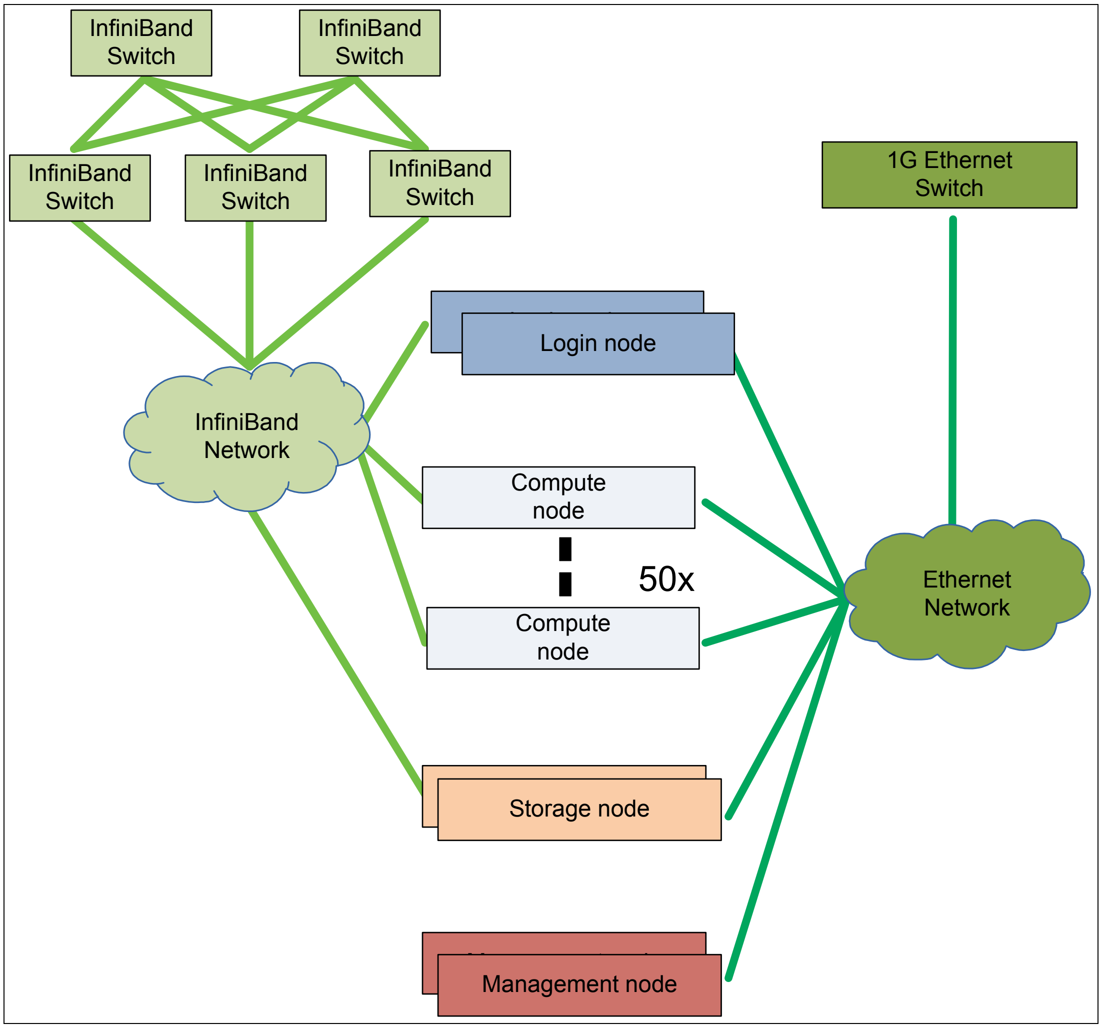


Figure 3-5 Medium solution architecture

The overview illustrates that all compute nodes are connected to the InfiniBand switches for the HPC/AI systems cluster and that the storage nodes are now using InfiniBand for internal cluster communication. To remove the costs of an extra 10-Gigabit Ethernet network, external storage is connected to the storage nodes only through Ethernet. InfiniBand now covers the application and storage traffic. With the increased number of compute nodes, the demand on storage performance increases. To support storage bandwidth and latency needs, a second storage node is added.

The management network is still in place for compute administration and OOB administration.

Network overview

Figure 3-6 is a network diagram that pertains to the InfiniBand infrastructure. It shows a fully non-blocking topology to support all 54 nodes. The management node is primarily using the Ethernet network for administration purposes. If more communication-intensive services such as job scheduling must run on the administration node, then add an InfiniBand link. The suggested fat-tree topology, which is built with five switches, can connect up to 60 nodes.

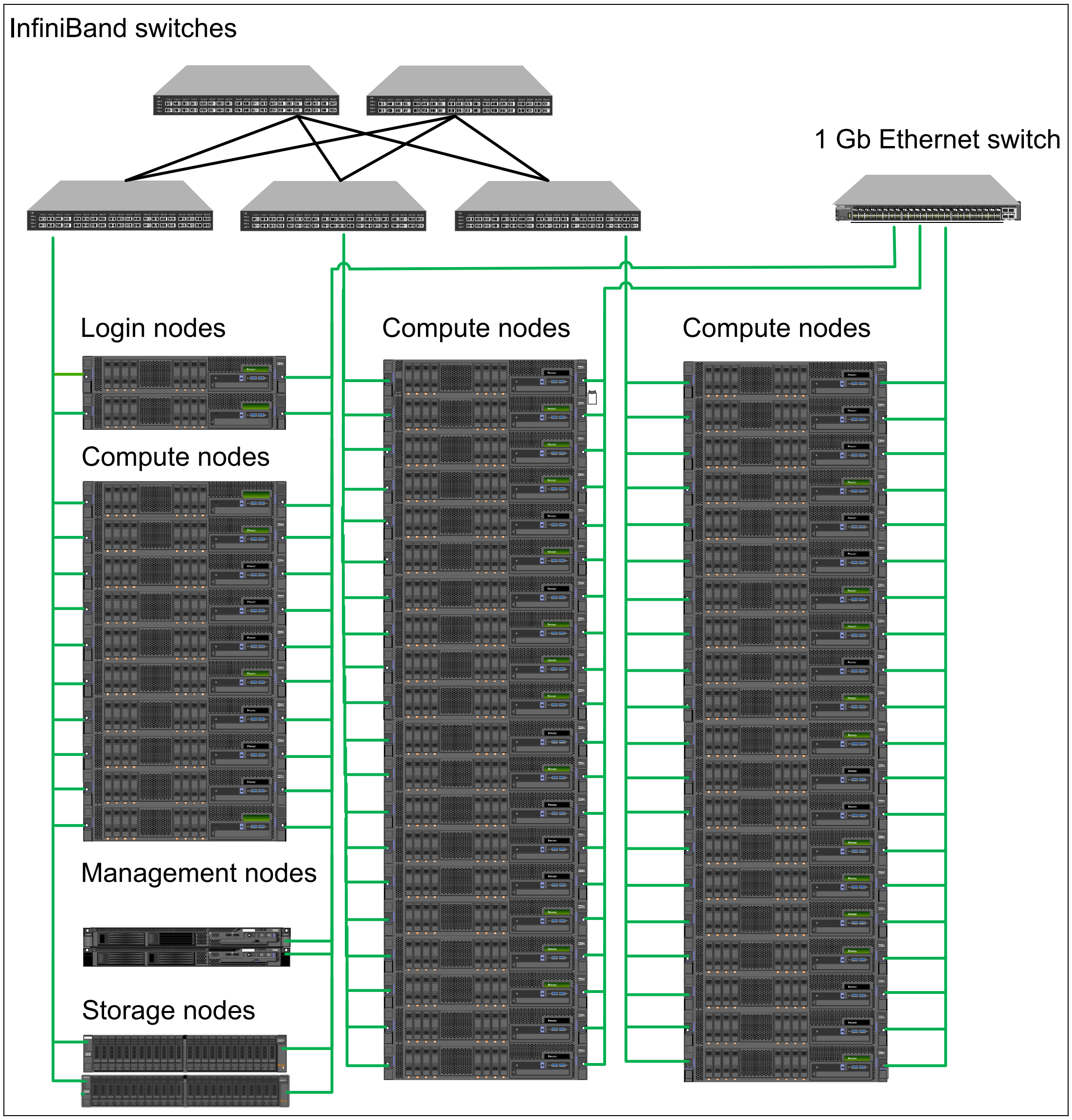


Figure 3-6 Medium solution non-blocking network

Rack view

Figure 3-7 shows an example of how devices can be installed within three 42 U racks. As in 3.1.2, “Medium solution” on page 33, the InfiniBand switches are in the center of each rack to support short cables. For the same reason, all racks should be placed side by side. As these racks are almost fully equipped, it is mandatory to verify the environmental support in the data center for power, cooling, and weight.

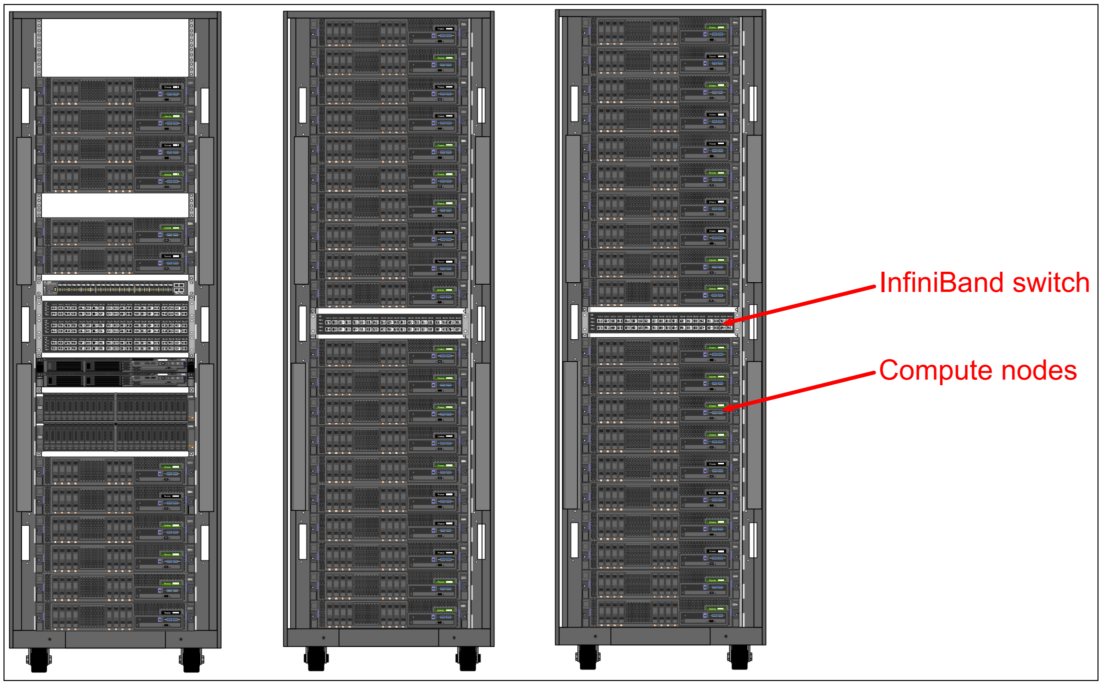


Figure 3-7 Medium solution rack diagram

3.1.3 Large solution

The final example describes a design for large clusters of hundreds to thousands of nodes. It introduces concepts of compute islands, hierarchical management, and blocking factors in the interconnect topology. Scalability is essential for the design and must scale out by adding more compute nodes, and scale up by using increasingly powerful, future hardware. This solution takes the lifecycle of the components into account and enables an update of the solution with next-generation hardware after it becomes available.

With the highly increased number of nodes, it becomes feasible to create a structure of multiple node groups, which are called *islands*. The islands concept enables subdivided management and maintenance because each island can be handled as own entity. From an application perspective, the island behaves as one large cluster, but each island can support special needs, such as more memory or acceleration hardware. If multiple software environments are needed, the solution can support different operating systems, scientific libraries, or software levels. A job scheduler should be able to take island boundaries into account for optimized job placement, and use the islands to offer environments for long and short running jobs, large memory, accelerators, or other special purposes.

To plan the InfiniBand topology and determine the optimal size of the islands, you must review the following application requirements:

- Number of nodes running one parallel application
- Interconnect bandwidth requirements of the applications
- Variation of hardware requirements (memory size, CPU/core counts, and accelerators)

If the applications scale to the maximum size of the cluster and require full interconnect bandwidth, then the InfiniBand topology must be fully non-blocking. For a fat-tree topology, fully non-blocking means many switch and cables, which requires a high investment. But, a cluster is used typically by multiple jobs of smaller sizes that fit into one island, so it is more cost-effective to design a non-blocking fabric within an island but use fewer connections between them. This setup introduces a blocking factor on the island level. Because of the size of the projected solution, you must use a new switch type, which is called a *director size switch*. It supports up to 800 ports and reduces the number of switch-to-switch cables.

In addition, even full cluster jobs often do not require the full interconnect bandwidth because they are limited by other limits, such as the CPU/GPU floating point operations per second (FLOPS) or memory bandwidth.

A fully non-blocking fat-tree topology greater than a specific size can be replaced by a director switch. Figure 3-8 shows a fat-tree non-blocking InfiniBand Architecture (IBA) that is equivalent to a director switch.

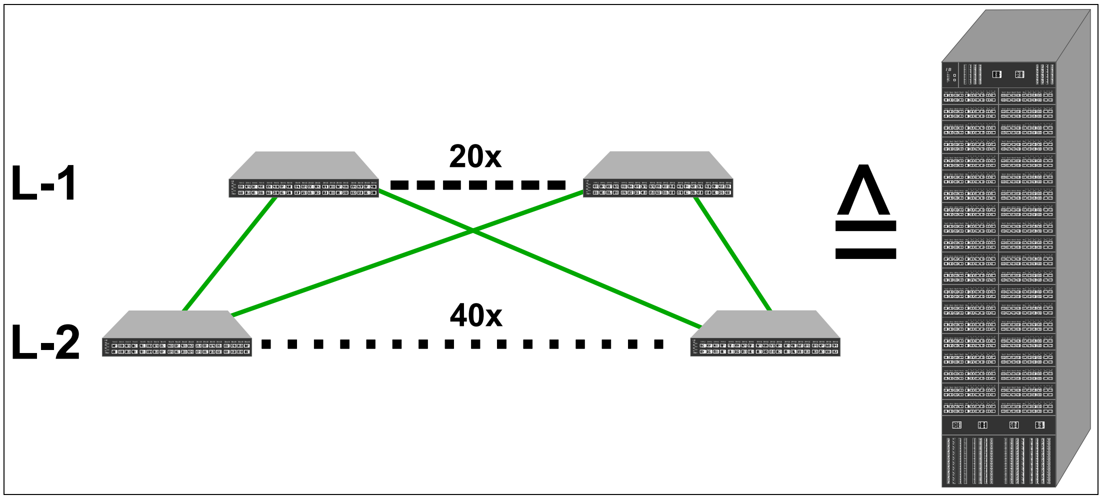


Figure 3-8 A fat-tree non-blocking network architecture

In this example, there is an InfiniBand network with 800 access ports. With a classic topology, 60 ToR switches and 800 switch-to-switch cables in between are required. In a director switch, these switch-to-switch cables are replaced by a hardwired midplane in the middle of the chassis. Furthermore, a switch is composed of four different module types:

Leaf-switching module This module is in the front of the chassis and contains the access ports. Half of the ports are externally connected, and the other half are internally connected.

Spine-switching module This module is in the back of the chassis and contains only internally connected ports. There is only half as many spine modules than leaf modules in the chassis.

Management module This module is responsible for the management of the switching modules in the chassis. A high availability capability creates redundancy, so there are usually two of them.

Power supply unit This module provides the electrical source of the chassis. Because of redundancy, there are several of them in the chassis.

Depending on the size of the cluster, a large HPC/AI solution requires more infrastructure components to manage and service the compute nodes. The following node types are included in the design:

- Login nodes
- Service nodes
- Storage nodes
- Compute nodes

A supplement to the nodes and the interconnect, there is an Ethernet network for OOB hardware management and administration purposes.

Designing an interconnect network for such an infrastructure is challenging. In this paper, the design is a simplified representation, and further customer-specific characteristics are not considered.

Architecture overview

The large solution example (Figure 3-9) starts with 1800 compute nodes, evenly distributed over three islands.

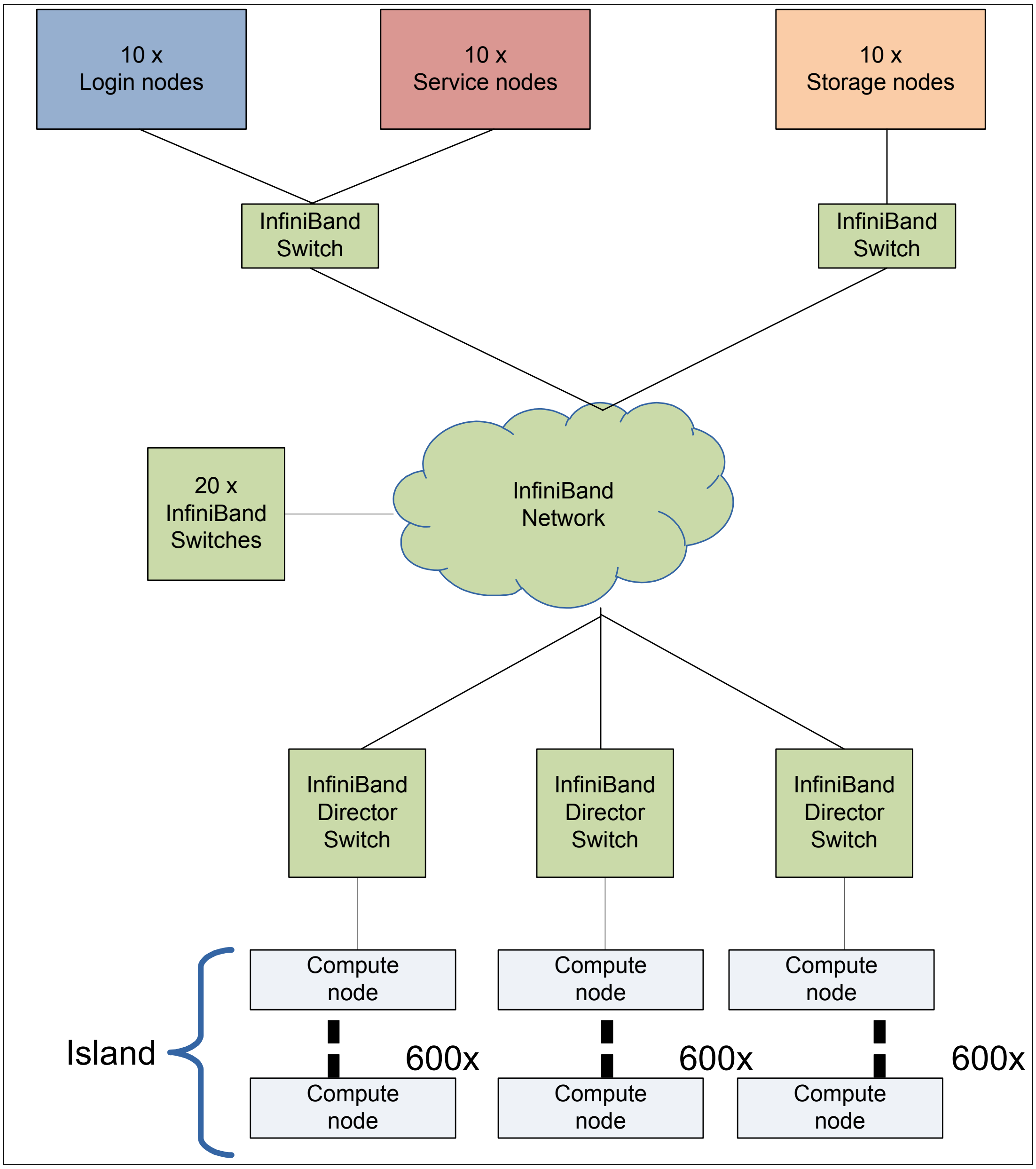


Figure 3-9 Large solution architecture

Note: The Ethernet network is not shown in Figure 3-9.

So, each island has 600 nodes that are managed by two service nodes. The management infrastructure is hierarchically organized with two service nodes acting as the master node, and two subordinate nodes that are responsible for each island. The nodes are set up in pairs for high availability. The hierarchical concept is necessary to manage the cluster efficiently and scale the service infrastructure in the same way as the compute resources.

The large design includes the following hardware components:

- Ten login nodes
- Ten service nodes (including management nodes)
- Twenty storage nodes (for example, IBM Elastic Storage Server)
- Eighteen hundred compute nodes
- Twenty-two InfiniBand 1RU Switches
- Three director switches

Network diagram

There are two approaches to building a fat-tree topology

- Use three hierarchy levels and group the nodes into islands. ToR switches are placed at the upper level of the fat-tree topology and director size switches at the lower level. This approach limits the node count of a single island because the size of the director switch and the blocking factor determines the maximum number of free ports and nodes.
- Place director switches at the upper level and ToR switches at the lower level to create larger island sizes, as described in 3.1.4, “Large solution upgrade” on page 43.

Figure 3-10 illustrates the fabric layout of the example. At the upper left of the picture, login and service nodes are connected directly to the same switch. The storage nodes at the upper right are also directly connected to a switch. Compute nodes are divided into three regions, which are called islands. An island consists of 30 racks with 20 compute nodes each (with water cooling), for a total of 600 nodes. Each compute node is connected to its island's director switch.

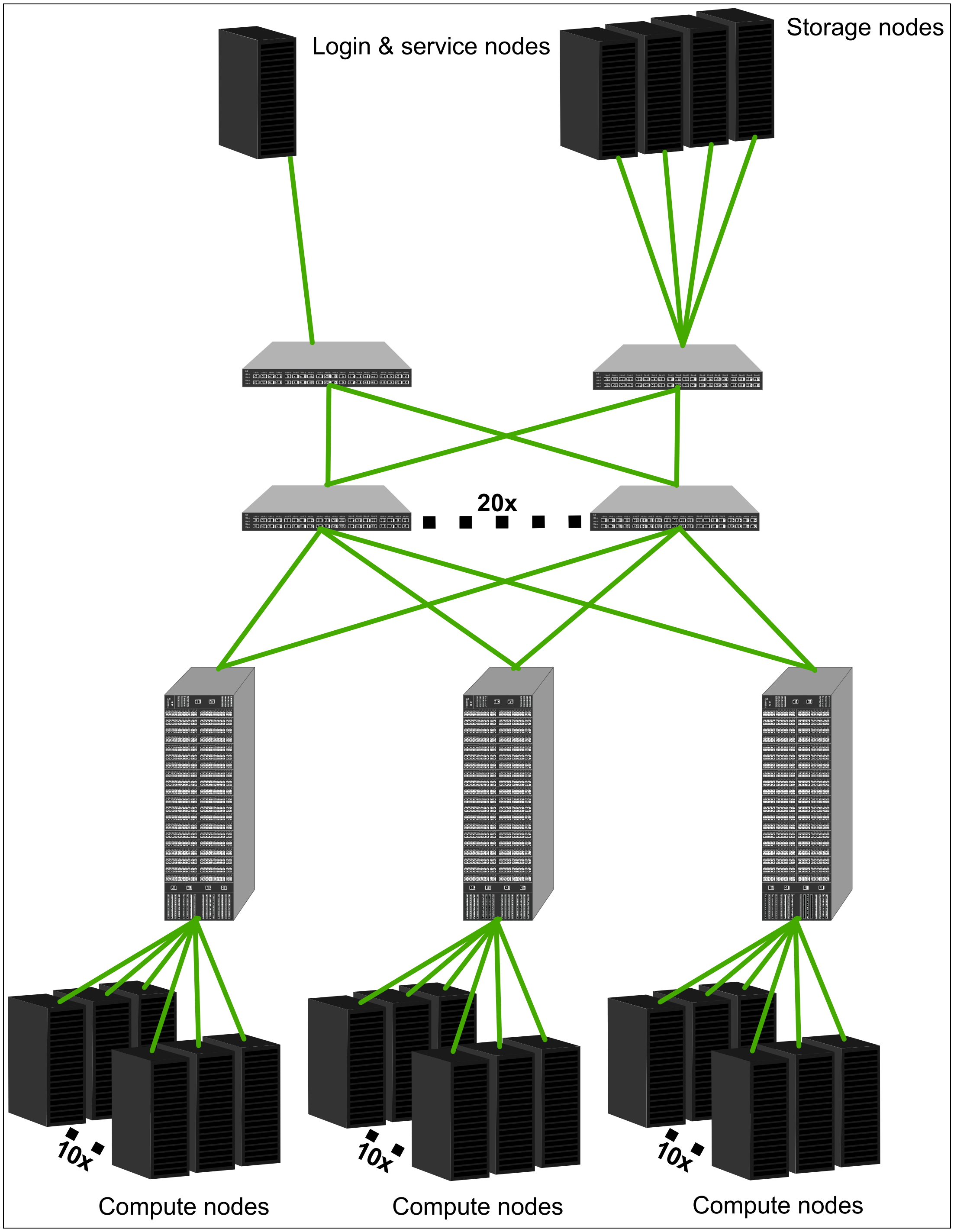


Figure 3-10 Large solution network diagram

In this example, every node is connected to the InfiniBand switch by only one cable, and the uplinks of the switches are connected as a fat-tree topology. The switches of the login, service, and storage nodes have 40 ports, where 20 of them are used as uplinks and 20 for connecting the nodes. Director switches have 800 ports, where 200 of them are used as uplinks and 600 for connecting the compute nodes of an island with a blocking factor of 1:3, which means that up to one-third of the islands nodes can communicate with full bandwidth to another island. If all nodes are transferring data in parallel, then the communication occurs with one-third of each nodes port bandwidth. Again, within an island, there is no such blocking, so all nodes can use the full bandwidth. The job scheduler must be aware of these limits and optimize job placement according to bandwidth needs.

3.1.4 Large solution upgrade

In this example, more compute resources are required and the solution that is described in 3.1.3, “Large solution” on page 37 must be extended. One option is to duplicate the configuration of the existing compute islands and add more of them, but also increase the number of compute nodes in a new island. In the solution that is described in 3.1.3, “Large solution” on page 37, the director switch limits the island size. To avoid that limitation, the fabric must provide more ports for compute nodes in a non-blocking setup. Therefore, a new layer of ToR switches is added, and the director switches now build a fat-tree topology at the upper level. As a side effect, more switches are required.

Architecture overview

The HPC/AI system is extended by using 1200 more nodes in one island. To support the increased number of compute resources, the service and storage infrastructure must be expanded as well. Therefore, the showcase doubles the number of login, service, and storage nodes. In a real scenario, these numbers must be validated by performance measurements.

The upgraded large design includes the following hardware components:

- Twenty login nodes
- Twenty service nodes (including management nodes)
- Forty storage nodes (for example, IBM Elastic Storage Server)
- Three thousand compute nodes
- Ninety-two InfiniBand 1 RU Switches
- Five director switches

Figure 3-11 shows the upgraded solution architecture.

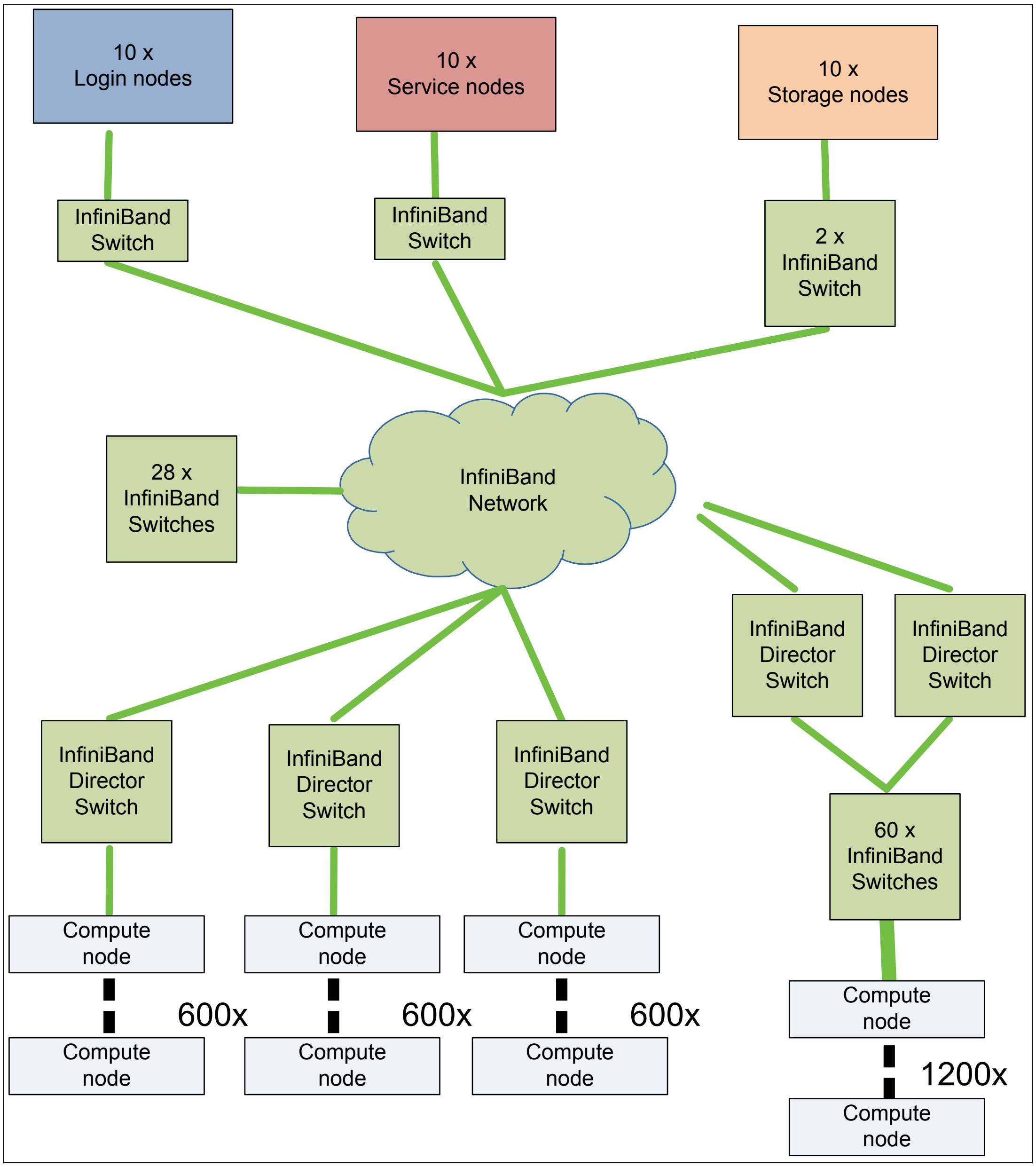


Figure 3-11 Large solution upgraded architecture

Network diagram

Figure 3-12 shows the expanded cluster. More login, service, storage, and compute nodes are added. All new compute nodes are in one large island at the lower right side. This new island is designed as an internal non-blocking architecture.

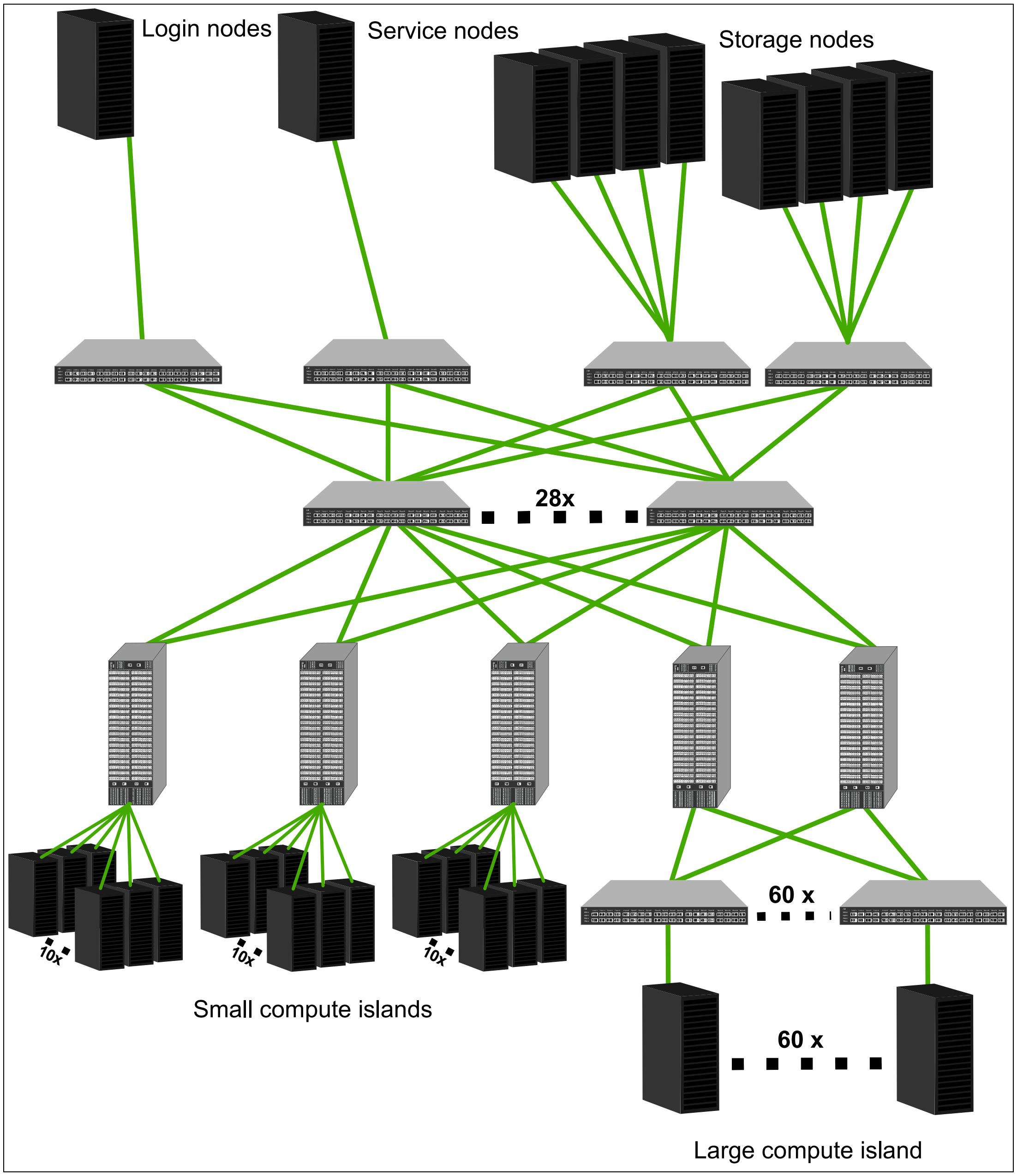


Figure 3-12 Large solution upgraded network diagram

Note the following details of the new large compute island:

- To enable a non-blocking architecture beyond the port number of a director switch, an extra switch layer is required. This layer is realized by using ToR switches, where one ToR switch per rack is used.
- The ToR switches are connected to the higher-level director switches through a fat-tree topology. Each ToR switch has 20 uplinks that are divided into 10 uplinks per director switch.
- The director switches, as in the other islands, are connected to the remaining cluster with a blocking factor.

The proposed design is based on an upgrade scenario where the new island must be integrated into the existing setup. Other variations are possible, for example, the new island can be used in a stand-alone approach.

3.1.5 Extra-large solutions / summary

An InfiniBand fabric theoretically can be scaled up to 40,000 nodes, and even that number can be exceeded by connecting multiple fabrics with InfiniBand routers. But, there are only a few examples in the world where large fabrics with more than 10,000 nodes are in use. Each of them is optimized for specific requirements, and there is no standard design to showcase here.

A HPC/AI solution must fulfill the application needs. CPU performance and core counts are steadily increasing and accelerators are boosting the compute performance. But, parallel HPC/AI applications must exchange the local computed results with other nodes, so the ratio between computation and interconnect performance becomes more important. There is no unique solution for all needs; you must carefully validate the bandwidth and latency needs to find the correct one. This chapter demonstrated some examples to start with, but modifications are necessary to meet the necessary characteristics.

Furthermore, the blocking factor is an important value, which should be chosen based on application characteristics.

3.2 Administration and out-of-band management network

The administration and OOB management networks are designed to ensure a dedicated path to access the operating system, whether a command-line or GUI interface, of all the devices in the HPC system without going through the production network.

As described in 2.3, “Ethernet” on page 22, LAN switches are used for this network because they provide an affordable, stable, and secure infrastructure for the management of the cluster.

Considerations

The amount of equipment that is involved in an HPC cluster might be low, medium, or high. Larger clusters might have a large amount of hardware that requires management. Providing a network that can meet all your requirements is challenging because most of the requirements are related to the switches that you select, the topology that you use, and how the cluster is managed.

For example, using low-cost access switches might provide an affordable network, but connectivity is not your only concern. Some of those switches support small MAC-Address tables that theoretically might be exceeded, and they provide little visibility or control over what you can or cannot do, which causes issues such as spanning tree misconfigurations, broadcast storms, and low CPU performance.

This network is the path to every device in your cluster, so correctly sizing the hardware and choosing the proper topology helps you avoid these and other similar problems on the management network.

Approaches

The following three approaches describe the design of an administration and OOB network. Each approach accounts for all of the points that are described in Chapter 1, “Understanding the requirements” on page 1, and uses different topologies and technologies. Other options are available, although the following ones are based on our expertise:

Approach 1 There are LAN switch providers that integrate fabric extenders into their portfolio, which enables individual switches to act as an extension of other more robust switches, which provide many benefits to the infrastructure.

For example, a network that uses two InfiniBand director switches with 500 nodes each for a single HPC solution is a good scenario to illustrate this approach.

Integrating fabric extenders into a high-performance switch in the OOB network means that there is no reliance on a spanning tree architecture (a single management point), so there will be few problems when you upgrade the OS on the switches.

Fabric extenders also offer great performance, low costs, redundancy on the top switches, and a working topology that is called End of Row. However, growth requires you to interconnect to other top switches. The number of ports that you can support in this solution might exceed 1100 Gb Ethernet ports.

Approach 2 The traditional star topology is where there are one or two core switches and some access switches where all nodes and InfiniBand equipment are connected.

Using the previous example of two InfiniBand director switches with 500 nodes each on a single solution, you have a port count of over 1000 ports. So, a configuration of 24 or 48-port access switches with two core switches for redundancy can support this solution and provide room for growth. You also can consider a single provider or mix of providers.

Switches with large MAC address tables, many CPUs, and a large amount of memory must be manageable. You can manage this network by using various tools, such as a single management dashboard that is provided by proprietary software, which enables you to monitor and configure correctly the equipment.

Approach 3 A traditional star topology that uses stacked access switches, which limits the number of cables that go up to the core (which can also be stacked). This topology reduces cabling costs and the number of devices to manage in the OOB topology because all switches in the stack provide a single management point.

In this example, you use two InfiniBand director switches with 500 nodes each on a single solution, with three stacks of eight switches of 48 ports each, with only two or four ports going up to the core from each stack. This solution requires fewer cables to the core, good performance on the stack, redundancy, less administration, and enough space to grow.

All three approaches provide a common OOB management network that is accessible, has proper performance and room for growth, and can include other services, such as bridging to the production network or access to the wide area network (WAN).

Recommendations

Designing the correct administration or OOB network is the first step. The second step is to provide the following minimum requirements:

- Servers where the support and management tools are going to be based should be accessible from any location inside and outside the OOB.
- Include failover for connectivity.
- Ensure that all tools have what they need to access the nodes, including separation through VLANs.
- Increase business continuity by creating the correct recovery plan for the fabric and the OOB.
- Keep inventory and equipment locations current.

These minimum requirements are only part of a broad range of options to consider.

Conclusion

Both OOB and HPC networks are completely, physically, and utterly separated regarding traffic and functions. Ensure that the administration or OOB network is a powerful management tool because it is available even if the entire HPC cluster fails.

Related publications

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

IBM Redbooks

The following IBM Redbooks publications provide more information about the topics in this document. Some publications that are referenced in this list might be available in softcopy only.

- *IBM PowerAI: Deep Learning Unleashed on IBM Power Systems Servers*, SG24-8409
- *IBM Power System AC922 Introduction and Technical Overview*, REDP-5472
- *HPC Clusters Using InfiniBand on IBM Power Systems Servers*, SG24-7767
- *Implementing InfiniBand on IBM System p*, SG24-7351

You can search for, view, download, or order these documents and other Redbooks, Redpapers, web docs, drafts, and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- IBM Power Systems, Storage, and Applications:
  - https://www-03.ibm.com/systems/power/solutions/bigdata-analytics/smartpaper/
  - https://www.ibm.com/us-en/marketplace/ibm-elastic-storage-server
  - https://www.ibm.com/us-en/marketplace/scale-out-file-and-object-storage
  - https://www.ibm.com/us-en/marketplace/ibm-elastic-storage-server
  - https://www.ibm.com/us-en/marketplace/spectrum-mpi
  - https://www.ibm.com/us-en/marketplace/deep-learning-platform
- InfiniBand, RDMA over Converged Ethernet (RoCE), and Ethernet:
  - https://en.wikipedia.org/wiki/InfiniBand
  - https://en.wikipedia.org/wiki/RDMA_over_Converged_Ethernet
  - https://en.wikipedia.org/wiki/Remote_direct_memory_access
  - http://www.mellanox.com/page/products_dyn?product_family=79
  - http://www.mellanox.com/related-docs/solutions/deploying-hpc-cluster-with-mellanox-infiniband-interconnect-solutions-archive.pdf
  - https://community.mellanox.com/docs/DOC-2402
  - http://www.roceinitiative.org/
  - https://www.cisco.com/c/dam/en_us/solutions/industries/docs/education/ethernet-solutions-high-performance-computing-education.pdf

- Out-of-band management:
  - https://en.wikipedia.org/wiki/Out-of-band_management
  - http://hpcmicrosystems.net/?page_id=79
  - https://insidehpc.com/2014/05/manage-hpc-cluster-software-complexity/

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

REDP-5478-00

ISBN 0738456837

Printed in U.S.A.