# Newest Trend in High-Performance Data Analytics: Tracing the Emergence of Clusters in High Performance Computing (1996-2005)

Sunny Jain
MatrNr: 21435344
Supervisor: Dr. Aashih Kumar Sharma

May 15, 2024

# Contents

# 1 Abstract

The period between 1996 and 2005 marked a significant juncture in the evolution of High Performance Computing (HPC), particularly with the emergence and proliferation of HPC clusters [3]. This study explores the pivotal developments and key milestones during this transformative era. It delves into the technological innovations that facilitated the rise of HPC clusters, including advancements in networking. It also examines the diverse range of applications and use cases that propelled the demand for HPC clusters across scientific and industrial domains. Economic factors and accessibility considerations are also addressed, elucidating how the affordability and democratization of HPC clusters revolutionized research and industry.

# 2 Introduction to High Performance Computing (HPC)

High Performance Computing (HPC) refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation. HPC systems are designed to solve complex problems that require substantial computational resources, making them indispensable for scientific research, engineering, and various industrial applications.

## 2.1 What is High Performance Computing (HPC) in Clusters ?

High Performance Computing (HPC) using clusters involves the aggregation of multiple computers, or nodes, to work together as a single, cohesive system to solve complex computational problems at high speeds. These clusters are interconnected through high-speed networks and are typically composed of commodity hardware, making them a cost-effective alternative to traditional supercomputers. Each node in the cluster runs its own operating system and works on a portion of the problem in parallel with other nodes, coordinated by middleware such as MPI (Message Passing Interface) [9] or PVM (Parallel Virtual Machine) [29]. This parallel processing capability enables HPC clusters to handle large-scale simulations, data analysis, and scientific computations with greater efficiency and speed than single, high-end machines. Scalability is a significant advantage of HPC clusters, as additional nodes can be added to increase computational power [2]. The flexibility and modularity of clusters also allow for tailored configurations to meet specific workload demands, making them suitable for a variety of fields including scientific research, engineering, financial modeling, and big data analytics. Effective management and scheduling of resources are critical to maximize the performance and utilization of HPC clusters, often achieved through sophisticated software solutions that ensure load balancing and minimize downtime.

## 2.2 How did HPC Evolve Prior to 1996 ?

The origins of HPC began with the development of mainframes in the 1950s and 1960s, such as IBM's 700 series [11]. These early systems were primarily used for business applications but laid the groundwork for more powerful computing machines. The concept of supercomputing emerged with the introduction of the CDC 6600 by Control Data Corporation in 1964,

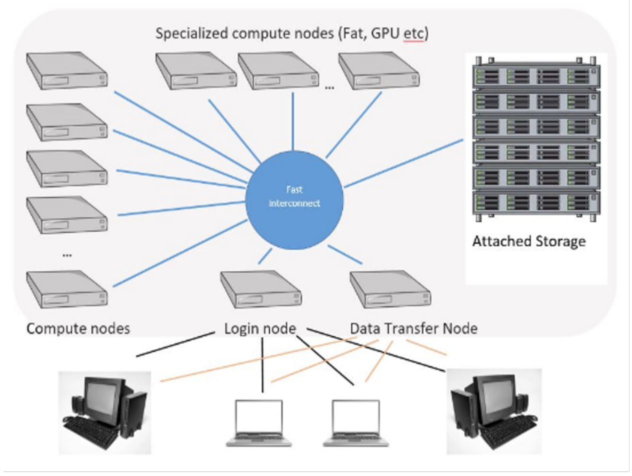

Figure 1: simple cluster representation
[3]

designed by Seymour Cray [6]. The CDC 6600 is often considered the world's first supercomputer and was capable of executing three million instructions per second. The 1970s and 1980s saw the development of vector processors, which could perform the same operation on multiple data points simultaneously. The Cray-1, introduced in 1976, was a notable example, utilizing vector processing to achieve unprecedented computational speeds [8]. This era also witnessed the rise of specialized architectures and the proliferation of companies like Cray Inc. and IBM, which became synonymous with high-performance computing. The 1980s introduced the concept of parallel processing, where multiple processors are used to perform computations simultaneously. This approach aimed to overcome the limitations of single-processor systems [6]. By the early 1990s, researchers began experimenting with clusters of commodity hardware, connected via high-speed networks, to perform parallel computations. This shift was driven by the desire to achieve supercomputing power at a fraction of the cost of traditional supercomputers. Alongside hardware advancements, significant progress was made in software development, particularly in the creation of parallel programming models such as the Message Passing Interface (MPI) [9] and Parallel Virtual Machine (PVM) [29]. These tools allowed for more efficient utilization of multiple processors and clusters, paving the way for more sophisticated HPC applications.

## 2.3 What Were the Main Drivers for the Development of HPC Technologies ?

The increasing complexity of scientific research and engineering problems necessitated more powerful computational tools. Fields such as climate modeling, astrophysics, molecular biology, and materials science required simulations and calculations that were beyond the capabilities of standard computers [30]. Industries such as aerospace, automotive, and pharmaceuticals also demanded high computational power for design, testing, and optimization processes. For instance, computational fluid dynamics (CFD) simulations in aerospace engineering or molecular modeling in drug discovery required substantial HPC resources [18]. Advances in semiconductor technology, leading to more powerful and energy-efficient processors, played a crucial role. Moore's Law [26], which predicts the doubling of transistors on a microchip approximately every two years, facilitated continuous improvements in computational power. Innovations in networking technology, particularly the development of high-speed interconnects like Ethernet and InfiniBand, enabled the effective linking of multiple processors and systems into powerful clusters [17]. The cost of traditional supercomputers was prohibitive for many organizations. The development of HPC clusters, which used commodity hardware, provided a cost-effective alternative, democratizing access to high-performance computing capabilities. Government agencies, research institutions, and universities played a significant role in funding and promoting HPC research. Initiatives like the High-Performance Computing and Communications (HPCC) program in the United States aimed to advance HPC technologies and infrastructure [19]. International competition in science, technology, and industry spurred investments in HPC. Countries and organizations sought to achieve leadership in various fields by leveraging the computational advantages provided by HPC systems [2].

## 2.4 How do clusters differ from other types of HPC architectures?

Clusters, supercomputers, and grids represent distinct high-performance computing (HPC) architectures, each with unique characteristics. Clusters are composed of a set of interconnected stand-alone computers, often referred to as nodes, that work together to perform parallel processing tasks [2]. This configuration leverages the combined processing power of multiple machines, making clusters scalable and relatively cost-effective, as they often use commodity hardware. Supercomputers, in contrast, are highly specialized, purpose-built machines designed for maximum performance, featuring tightly integrated processors, memory, and storage with specialized interconnects, all housed within a single system. They are typically more expensive and less flexible than clusters but deliver superior computational power and speed for complex simulations and large-scale computations. Grids, meanwhile, link geographically dispersed computing resources over a network to form a virtual supercomputer, enabling resource sharing across multiple locations [30]. Unlike clusters, grids are designed to handle diverse and distributed workloads, often in a more decentralized and heterogeneous environment, making them ideal for tasks that require substantial computing resources but do not necessitate tight coupling of processors. Each of these architectures addresses different needs and use cases in the realm of HPC, balancing trade-offs between performance, cost, flexibility, and scalability [3].

# 3 Typical Cluster Architecture

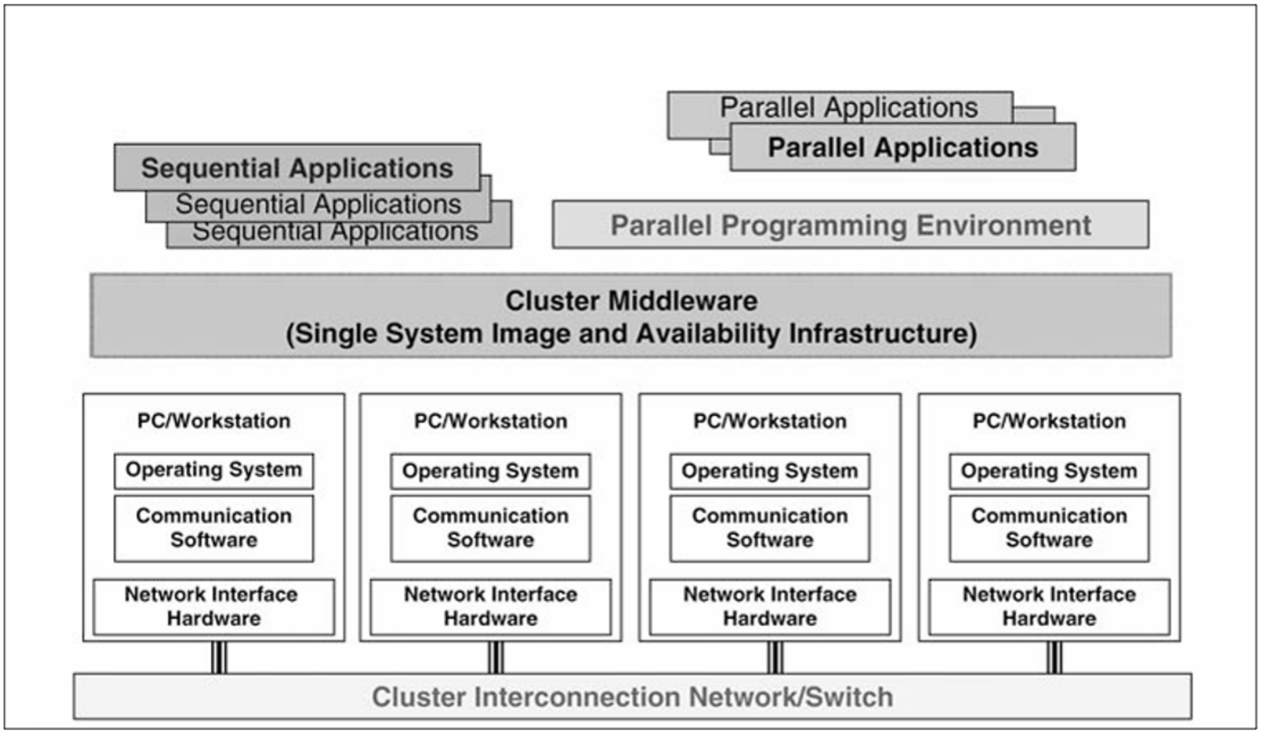

Figure 2: Typical Cluster Architecture
[21]

This diagram provides a detailed view of a Cluster Computing System and explains how various components interact to provide both parallel and sequential computing capabilities [5]. At the base of the architecture are the individual PCs or Workstations that make up the cluster. Each of these nodes consists of the following components:

- Operating System (OS): Each node runs its own operating system, such as Linux, Windows, or another OS. This OS manages the node's hardware resources and provides the necessary platform for applications and processes to run [5].

- Communication Software: This component is critical for enabling the communication between different nodes in the cluster. It manages data transmission between the nodes via the cluster's network. Examples of communication software include MPI (Message Passing Interface), PVM (Parallel Virtual Machine), or proprietary communication systems [5].

- Network Interface Hardware: This refers to the physical network interface cards (NICs) and related hardware that allow nodes to communicate with each other over the network. High-speed network interfaces (like InfiniBand or Gigabit Ethernet) are often used to facilitate rapid data exchange between nodes [5].

**Cluster Middleware (Single System Image and Availability Infrastructure)**

- Single System Image (SSI): The middleware provides an abstraction that makes the entire cluster look like a single unified system to users and applications. This simplifies both management and usage by offering a consistent interface, regardless of the hardware in use [1].
- Availability Infrastructure: Ensures high availability and reliability through fault tolerance mechanisms like failover, load balancing, and redundancy. This infrastructure supports continuous operation and quick recovery from any hardware or software failures.

**Parallel Programming Environment:** This environment includes libraries, compilers, and tools that aid in developing parallel applications [9]. Examples include:

- MPI (Message Passing Interface): A standardized, portable message-passing system compatible with various parallel computing architectures.
- OpenMPI: An API supporting multi-platform shared memory multiprocessing programming in languages like C, C++, and Fortran.

**4 PC/Workstations:** The cluster comprises four interconnected PCs or workstations that function as compute nodes. Each workstation contributes processing power, memory, and storage to the overall cluster. A high-speed network switch links the workstations, facilitating fast communication and data transfer between nodes [3].

- Network Switch: A multi-port high-speed device, such as a 10 Gigabit Ethernet switch or an InfiniBand switch, enables data transfer between PCs/workstations.
- Interconnection Network: This network infrastructure connects the nodes, providing low-latency, high-bandwidth communication, essential for the efficient execution of parallel applications.

# 4 Scope

We are dealing with the scope 1995-2005. We are analyzing the papers and articles published during this periods because the 1995 Top500 list, which ranks the world's most powerful supercomputers, saw significant changes in its user demographics [10]. There was a notable decrease in industrial customers and a rise in government-funded research sites. This shift was largely influenced by government high-performance computing (HPC) initiatives that encouraged the use of parallel systems and clusters. Clusters, a type of parallel computing system, consist of interconnected computers working as a unified resource, offering a cost-effective solution for high-performance needs, particularly appealing to government-funded research institutions.

The Berkeley NOW project is working on enabling a network of workstations (NOW) to function as a distributed supercomputer at a building-wide level [7]. Given the mass production of commercial workstations, they now offer better price-to-performance ratios compared to individual nodes of MPP systems. Furthermore, switch-based networks like ATM provide affordable, high-bandwidth communication. This advantage in price and performance is amplified if the NOW can handle both typical workstation tasks and large-scale programs.

Beowulf clusters, a type of parallel computing system utilizing multiple standard computers connected via a LAN, gained popularity in academic and research institutions due to affordability and scalability. Hardware improvements, including faster processors, more RAM, and better network interfaces, significantly boosted Beowulf clusters' performance. Advances were also seen in software development, as well as community and collaborative efforts [21].

Oracle 9i RAC enabled multiple computers (nodes) to access a single database simultaneously, enhancing scalability and availability—an essential feature for business-critical applications requiring high uptime and seamless load handling [14]. Clusters built with commodity hardware provided a cost-effective alternative to traditional large SMP (Symmetric Multiprocessing) systems [14].

By 2001, clusters comprised around 40% of systems on the TOP500 list, highlighting their rising popularity in high-performance computing due to their cost-effectiveness and scalability [22].

The TeraGrid initiative proposed creating a national grid of interconnected high-performance computing (HPC) clusters, aimed at providing vast computational power to support scientific research across various disciplines [25].

## 5 Major Components of Cluster technology

Clusters rely on several major components to support high-performance computing. **Programming models** like MPI (Message Passing Interface) [9] and PVM (Parallel Virtual Machine) are fundamental [29]. MPI, which became the dominant model for distributed memory systems, allowed parallel programs to communicate and synchronize across distributed nodes efficiently. Its versions, MPI-1 (introduced in 1994) and MPI-2 (introduced in 1997), were widely adopted in cluster environments [9]. Similarly, PVM provided mechanisms for process creation, communication, and synchronization across a network of heterogeneous computers, making it popular for parallel computing.

**Cluster resource management** is another critical component, encompassing job scheduling systems and resource management frameworks. Job scheduling systems like PBS (Portable Batch System), originally developed in the early 1990s, became widely used for job scheduling and resource management in clusters [27]. SGE (Sun Grid Engine), developed by Sun Microsystems, offered scalable resource management for clusters and grids, while LSF (Load Sharing Facility) by Platform Computing was also widely adopted for job scheduling and cluster management. Among resource management frameworks, Condor, developed at the University of Wisconsin-Madison, provided a specialized high-throughput computing environment [27]. SLURM (Simple Linux Utility for Resource Management), initially released in the early 2000s, gained popularity for managing resources in Linux clusters [27].

**Cluster monitoring tools** such as Ganglia and Nagios are essential for real-time monitoring of system health and performance. Ganglia, developed in the early 2000s, provided a scalable distributed monitoring system, offering insights into CPU, memory, and network metrics across high-performance computing systems [31]. Nagios, released in 1999, became a popular tool for monitoring IT infrastructure, including clusters, with its comprehensive monitoring capabilities and alerting features [31].

In terms of **parallel file systems**, the development of systems like Lustre (introduced in 2003) and GPFS (General Parallel File System) significantly improved I/O performance for clustered systems. Lastly, there was growing interest in **grid computing** technologies, which aimed to federate resources across multiple administrative domains to facilitate collaboration and resource sharing in distributed environments [29].

**Interconnect** technologies evolved rapidly to meet the increasing demands of high-performance computing (HPC) clusters. Fast Ethernet (100 Mbps) was initially common but quickly became a bottleneck, leading to the adoption of Gigabit Ethernet (1 Gbps) in the early 2000s for improved speed and compatibility, though it still struggled with latency. High-performance alternatives like Myrinet and InfiniBand emerged, offering significantly lower latency and higher bandwidth. Myrinet, introduced in the late 1990s, delivered up to 2 Gbps with latencies under 10 microseconds, making it popular in scientific clusters. InfiniBand, introduced in the early 2000s, provided even greater performance with speeds of up to 10 Gbps and excellent scalability, quickly becoming the standard in many HPC environments. Quadrics and SCI (Scalable Coherent Interface) also offered low-latency solutions with specific advantages, though they saw limited adoption compared to InfiniBand. Although ATM (Asynchronous Transfer Mode) was used briefly, it was ultimately surpassed by faster and more cost-effective options. These advancements in interconnect technology helped HPC clusters achieve better communication efficiency and computational performance [17].

## 5.1 Interconnects :

This research primarily focuses on interconnects. The interconnection technologies that will be discussed are Gigabit Ethernet, Fast Ethernet, Scalable Coherent Interface (SCI), Infiniband, QsNet-II and Myrinet [20].

### 5.1.1 Fast Ethernet :

Fast Ethernet, introduced in 1995 as the IEEE 802.3u standard, served as the fastest Ethernet version for three years before the advent of Gigabit Ethernet. This cost-effective local area network (LAN) technology offers a bandwidth of 100 Mbps while preserving the original Ethernet transmission protocol, CSMA/CD [20]. Although TCP/IP is the most widely used communication protocol for Fast Ethernet, alternatives like VIA can also be employed. TCP/IP is a versatile protocol suite designed to connect diverse networks from different vendors into a cohesive network of networks [20]. When paired with a high-performance communication protocol like VIA, Fast Ethernet provides an appealing cost-benefit ratio, making it a viable option for connecting machines in small cluster environments where bandwidth demands are not critical. However, its use in modern high-performance computing (HPC) clusters is rare due to its limited bandwidth [20].

### 5.1.2 Gigabit Ethernet

The rise in microprocessor speeds outpaced Fast Ethernet capabilities, leading to the approval of the Gigabit Ethernet standard in 1998 [12]. This technology builds on the IEEE 802.3 MAC and integrates the tested physical layer of the ANSI Fiber-Channel using an 8B/10B block coding system. The Advanced Computational Concepts Laboratory (ACCL) at NASA Lewis Research Center adopted Gigabit Ethernet for its high throughput and ease of deployment in cluster environments. The cluster comprises intermediate workstations connected by Gigabit Ethernet repeaters and leaf workstations linked via Fast Ethernet, effectively forming a 100 Mbps cluster testbed while supporting half-duplex and full-duplex operations [12].

### 5.1.3 InfiniBand

InfiniBand (IB) is a high-performance computing networking standard known for its high throughput and low latency, facilitating data interconnect among and within computers. Originating in 1999 from the merger of Future I/O and Next Generation I/O (led by Intel), InfiniBand has become essential in supercomputing [28]. In 2009, it was used in 181 of the top 500 supercomputers, compared to 259 using Gigabit Ethernet. InfiniBand enhances system performance through increased bandwidth, reduced latency for real-time interactions, improved scalability for larger datasets and workloads, and optimized data transfer protocols, maximizing resource utilization and overall application performance [28].

### 5.1.4 Myrinet

Myrinet is a high-cost technology offering low latency and high bandwidth for end-to-end communication between cluster nodes. It includes both a switch and host interface, allowing packets of any length, which can encapsulate other packet types like IP without needing an adaptation layer [4]. Myrinet scales effectively, supporting large configurations with hundreds or thousands of nodes. Its Network Interface Cards (NICs) feature specialized processing capabilities that offload communication tasks from host CPUs, enhancing application performance. With cross-platform support and stable drivers for Linux, Windows, and Unix, Myrinet ensures reliable cluster operation through features like CRC checking and retransmission, alongside management tools for deployment [4].

### 5.1.5 QsNet II

QsNet II, launched in 2003 as the fourth generation of Quadrics Interconnect products, features intelligent network interface cards (NICs) with dedicated processing power and memory, offloading communication tasks from host CPUs [23]. Designed for low-latency communication, it achieves typical round-trip latencies of 3-5 microseconds. While it can operate with TCP/IP, it is commonly used with communication APIs like Message Passing Interface (MPI) or SHMEM. QsNet II connects to host computers via the PCI-X bus, with later versions supporting PCIe, albeit with a performance penalty. Its architecture utilizes two ASICs : Elan4 for communication processing and Elite4 for switching—supporting up to 4,096 nodes and over 10,000 CPUs [23],

### 5.1.6 Scalable Coherent Interface

While SCI gained some traction in high-performance computing (HPC) and server systems, it never became a mainstream interconnect standard. The IEEE P1596 working group maintained and updated the SCI standard, focusing on performance, reliability, and interoperability improvements [13]. Major vendors like Compaq, DEC, HP, IBM, and Sun developed SCI-based systems, but competition from technologies like InfiniBand increased. By the mid-2000s, innovation and new products related to SCI began to decline, partly due to a limited ecosystem and software support, which complicated integration into common computing environments. Additionally, SCI's cache coherency and distributed directory protocols added unnecessary complexity for many applications [13].

## 5.2 Comparison

| **Network** | **BW**[1] | **Lat.**[2] | **HW Avail.** | **Linux** | **Max Nodes** | **Protocol Impl.** | **VIA** | **MPI** |
|---|---|---|---|---|---|---|---|---|
| Gigabit Eth. | <100 | <100 | Yes | Yes | 1000s | Hardware | NT/Linux | MVICH,<br>M-VIA,<br>TCP |
| Fast Eth. | <100 | <125 | Yes | Yes | <800 | Hardware | NT/Linux | MVICH |
| Infiband | 850 | <7 | Yes | Yes | >1000s | Hardware | SW | MPI/Pro |
| Myrinet | 230 | 10 | Yes | Yes | 1000s | FW Adaptor | Linux | 3rd Party |
| QsNet II | 1064 | <3 | Yes | Yes | 4096 | FW Adaptor | None | Quadrics |
| SCI | <320 | 1-2 | Yes | Yes | 1000s | FW Adaptor | SW | 3rd Party |

Table 1: Comparison of Network Technologies

BW: Bandwidth (Gbps)

Lat.: Latency (s)

Note: FW = Firmware, SW = Software, Eth. = Ethernet

## 5.3 Choosing Interconnects for Clusters

First-time builders typically face a choice between low-cost commodity interconnects, like Gigabit Ethernet, and more expensive high-performance options, such as Myrinet or SCI. This decision is often driven by budget considerations, which can result in performance limitations or misallocation of funds. Many builders opt for Fast Ethernet initially due to its affordability and later upgrade to more capable interconnects as their needs evolve [15].

The comparison between Commodity Interconnect and High-Performance Interconnect technologies reveals distinct characteristics suited for different use cases. Commodity Interconnect, which includes Fast Ethernet (100 Mbps) and Gigabit Ethernet (1000 Mbps), offers a cost-effective solution with flexible redeployment options but has a shorter lifespan. It's ideal for applications with infrequent, large messages. In contrast, High-Performance Interconnect, featuring Myrinet and SCI, comes at a higher cost with less flexible redeployment options but provides a longer lifespan. It's optimized for applications requiring frequent transmission of small messages, making it suitable for specialized performance needs [17].

## 5.4 Graphs of Comparison

The **Pallas Ping-Pong Bandwidth** benchmark evaluates communication performance between two nodes in high-performance computing clusters [16].

This benchmark operates by exchanging messages between two nodes (A and B) in a back-and-forth pattern, with increasing message sizes per iteration. The test utilizes MPI (Message Passing Interface) to measure maximum achievable bandwidth, though real-world performance typically falls below these ideal conditions [16].

Performance analysis shows distinct characteristics across different interconnect technologies:

The x-axis represents the message size in bytes, while the y-axis represents the achieved bandwidth in Mbps (Megabits per second). Figure-3 depicts how the bandwidth changes as the message size increases for each interconnect technology.

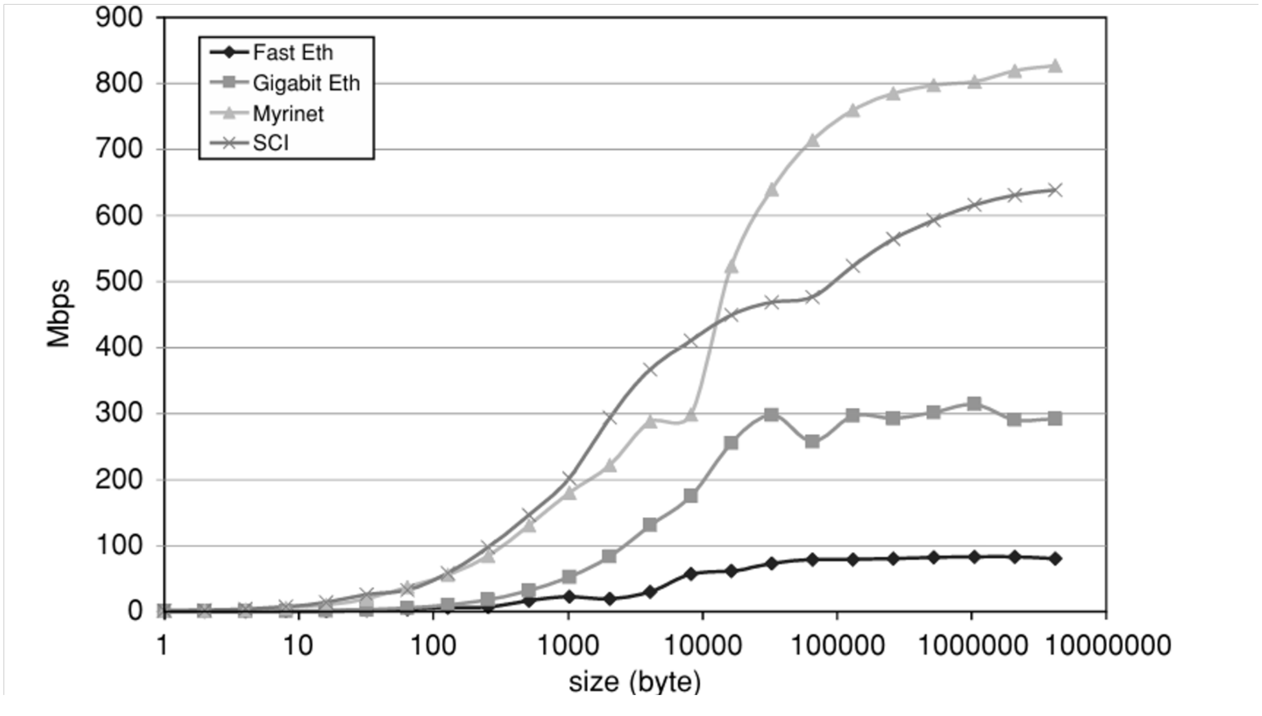

Figure 3: Pallas Ping-Pong Bandwidth

The Figure-4 compares latency performance across different interconnect technologies (Fast Ethernet, Gigabit Ethernet, Myrinet, and SCI) in relation to message size. For small messages up to 32 bytes, all technologies demonstrate similar, low latency [24]. As message size increases, performance diverges significantly. Fast Ethernet shows the highest latency (up to 262 milliseconds), while Gigabit Ethernet performs better but still lags behind specialized

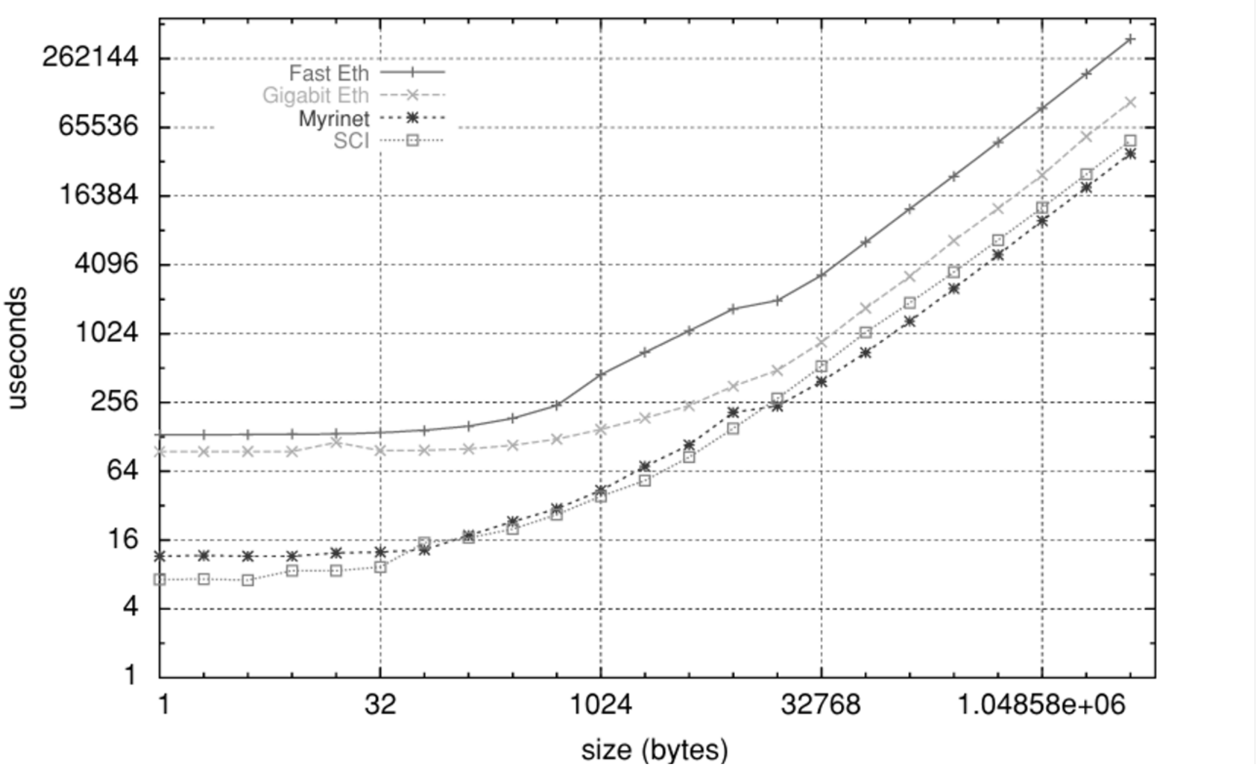

Figure 4: Latency of Ping-Pong Packets

solutions. High-performance interconnects - Myrinet and SCI - exhibit superior latency characteristics, with SCI showing optimal performance for larger messages. This comparison aids in selecting appropriate interconnect technology based on specific latency requirements [24]. The y-axis represents **the latency in microseconds**, while the x-axis shows the message size on a logarithmic scale.

# 6 Key Results

The selection between commodity and high-performance interconnects involves careful consideration of budget, performance needs, and future scalability. While commodity interconnects like Fast Ethernet offer cost-effective initial solutions with flexible redeployment options, they may face performance limitations as cluster demands grow. High-performance interconnects provide superior performance and better scalability but require larger upfront investment and have limited redeployment flexibility due to their specialized nature. A strategic approach often involves starting with commodity solutions and gradually upgrading to high-performance interconnects as operational needs evolve and resources become available, ensuring a balance between immediate budget constraints and long-term performance requirements.

# List of Tables

# List of Figures

# References

[1] Jameela Al-Jaroodi, Nader Mohamed, Hong Jiang, and David Swanson. Middleware infrastructure for parallel and distributed programming models in heterogeneous systems. *IEEE Transactions on Parallel and Distributed Systems*, 14(11):1100–1111, 2003.

[2] Mark Baker and Rajkumar Buyya. Cluster computing at a glance. *High Performance Cluster Computing: Architectures and Systems*, 1(3-47):12, 1999.

[3] Geoffrey G. Bell. Clusters, networks, and firm innovativeness. *Strategic Management Journal*, 26(3):287–295, 2005.

[4] Nanette J Boden, Danny Cohen, Robert E Felderman, Alan E. Kulawik, Charles L Seitz, Jakov N Seizovic, and Wen-King Su. Myrinet: A gigabit-per-second local area network. *IEEE micro*, 15(1):29–36, 1995.

[5] Rajkumar Buyya et al. High performance cluster computing: Architectures and systems (volume 1). *Prentice Hall, Upper Saddle River, NJ, USA*, 1:29, 1999.

[6] Seymour Cray and William Norris. Control data corporation.

[7] Anderson D.E. Culler, T.E., D.A. Patterson, et al. A case for now (networks of workstations). *IEEE Micro*, 15(1):54–64, 1995.

[8] Lester T. Davis. The balance of power: A brief history of cray research hardware architectures. *Advances in Parallel Computing*, 10:121–132, 1995.

[9] Narayan Desai, Rick Bradshaw, Andrew Lusk, and Ewing Lusk. Mpi cluster system software. In *European Parallel Virtual Machine/Message Passing Interface Users' Group Meeting*, pages 277–286. Springer, 2004.

[10] Jack J. Dongarra and Horst D. Simon. High performance computing in the top500 list. *Supercomputer*, 13:19–28, 1997.

[11] Jim Elliott and Advocate-Infrastructure Solutions. History and evolution of ibm mainframes. 2005.

[12] Paul A Farrell and Hong Ong. Communication performance over a gigabit ethernet network. In *Conference Proceedings of the 2000 IEEE International Performance, Computing, and Communications Conference (Cat. No. 00CH37086)*, pages 181–189. IEEE, 2000.

[13] David B Gustavson and Qiang Li. The scalable coherent interface (sci). *IEEE Communications Magazine*, 34(8):52–63, 1996.

[14] James Hamilton. Oracle real application clusters and industry trends in cluster parallelism and availability, 2004.

[15] Wai Hong Ho and Timothy Mark Pinkston. A clustering approach in characterizing interconnection networks. In *Proceedings. Fifth International Conference on High Performance Computing (Cat. No. 98EX238)*, pages 277–284. IEEE, 1998.

[16] Swamy Kandadai and Suga Sugavanam. Redbooks paper. 2004.

[17] Jay Kandampully. Innovation as the core competency of a service organisation: The role of technology, knowledge, and networks. *European Journal of Innovation Management*, 5(1):18–26, 2002.

[18] S.J. Lawson, M. Woodgate, R. Steijl, and G.N. Barakos. High performance computing for challenging problems in computational fluid dynamics. *Progress in Aerospace Sciences*, 52:19–29, 2012.

[19] Donald A.B. Lindberg and Betsy L. Humphreys. The high-performance computing and communications program, the national information infrastructure, and health care. *Journal of the American Medical Informatics Association*, 2(3):156–159, 1995.

[20] Marcelo Lobosco, Vítor Santos Costa, and Claudio L De Amorim. Performance evaluation of fast ethernet, giganet and myrinet on a cluster. In *Computational Science—ICCS 2002: International Conference Amsterdam, The Netherlands, April 21–24, 2002 Proceedings, Part I 2*, pages 296–305. Springer, 2002.

[21] Marsha Meredith, Teresa Carrigan, James Brockman, Timothy Cloninger, Jaroslav Privoznik, and Jeffery Williams. Exploring beowulf clusters. *Journal of Computing Sciences in Colleges*, 18(4):268–284, 2003.

[22] Philip M Papadopoulos, Caroline A Papadopoulos, Mason J Katz, William J Link, and Greg Bruno. Configuring large high-performance clusters at lightspeed: A case study. *The International Journal of High Performance Computing Applications*, 18(3):317–326, 2004.

[23] Fabrizio Petrini, Wu-chun Feng, Adolfy Hoisie, Salvador Coll, and Eitan Frachtenberg. The quadrics network: High-performance clustering technology. *Ieee Micro*, 22(1):46–57, 2002.

[24] Hossein Pourreza, M Rasit Eskicioglu, and Peter CJ Graham. Preliminary performance assessment of four cluster interconnects on identical hardware. In *HPCS*, pages 209–212, 2004.

[25] Daniel A Reed. Grids, the teragrid and beyond. *Computer*, 36(1):62–68, 2003.

[26] Robert R. Schaller. Moore's law: Past, present, and future. *IEEE Spectrum*, 34(6):52–59, 1997.

[27] Kai Shen, Hong Tang, Tao Yang, and Lingkun Chu. Integrated resource management for cluster-based internet services. *ACM SIGOPS Operating Systems Review*, 36(SI):225–238, 2002.

[28] Shinji Sumimoto, Akira Naruse, Kouichi Kumon, Kouji Hosoe, and Toshiyuki Shimizu. Pm/infiniband-fj: a high performance communication facility using infiniband for large scale pc clusters. In *Proceedings. Seventh International Conference on High Performance Computing and Grid in Asia Pacific Region, 2004.*, pages 104–113. IEEE, 2004.

[29] Vaidy S. Sunderam. Pvm: A framework for parallel distributed computing. *Concurrency: Practice and Experience*, 2(4):315–339, 1990.

[30] Eric Winsberg. Computer simulations in science. 2013.

[31] Serafeim Zanikolas and Rizos Sakellariou. A taxonomy of grid monitoring systems. *Future Generation Computer Systems*, 21(1):163–188, 2005.