System Design: A Professional Approach

Shraddha Mahesh Thanki[1]

*Corresponding author: shraddha.thanki@stud.th-deg.de

Keywords:
Network Topology
Computational Efficiency
Parallel File systems
Benchmarking in HPC
FLOPS, IOPS
Power Usage Effectiveness(PUE)
Total Cost of Ownership (TCO)

Abstract This paper explores system design in High-Performance Computing (HPC), focusing on the alignment of CPU and network topology selection with application requirements. It examines how understanding application characteristics is vital for optimizing HPC system performance, extending to storage, parallel file systems, and processing techniques. The study also considers environmental factors, such as power consumption and cooling technologies, evaluating system efficiency through metrics like Power Usage Effectiveness (PUE) and Total Cost of Ownership (TCO). The findings offer insights into the balance between technological advancement and environmental sustainability in HPC system design, contributing to the field's evolving landscape.

 Submitted: 22 December 2023 as the Master Short paper.

1. Introduction

High-Performance Computing (HPC) is now recognized as a promising field. Fundamental technology has led to major developments in a variety of scientific along industrial domains. The capacity of HPC systems' ability to process and analyze large datasets at previously unheard-of speeds has opened up new avenues for research and development. However, the effectiveness of these systems is dependent on their design, which requires an in-depth knowledge of both hardware capabilities and application requirements.

Moreover, the negative environmental effect of HPC systems has come out as a major concern. These powerful machines' energy consumption and cooling demands pose significant sustainability challenges. Power Usage Effectiveness (PUE) and Total Cost of Ownership (TCO) data have become essential for evaluating the environmental and economic viability of HPC systems, [1], [2].

This paper aims to bridge the knowledge gap in HPC system design, focusing on the alignment of hardware selection with application requirements and environmental sustainability. By analyzing current best practices and emerging trends, the study provides insights into designing HPC systems that are not only technologically advanced but also environmentally conscious. The ultimate goal is to contribute to the development of HPC systems that balance high computational power with energy efficiency and cost-effectiveness, paving the way for sustainable technological progress.

2. Foundations of High-Performance Computing (HPC)

High-performance computing (HPC) is the consolidation of computing power leading toăsignificantly higher performance than the average desktop or workspace. This is achieved by using supercomputers and computer clusters to carry out complex calculations on huge databases. HPC is vital for solving advanced problems in a variety of fields, including climate research, molecular modeling, physical simulations, and the analysis of large data sets used in high-energy physics and genome sequencing.

3. Core Principle: Parallel Processing

The concept of parallel processing is crucial to HPC. Compared to traditional computing, where tasks are executed sequentially, parallel processing entails breaking down a large problem into smaller sub-problems that are solved simultaneously using multiple processors. This method significantly reduces computation time, allowing for the efficient handling of complex, large-scale computations.

Parallel processing in HPC is possible through the use of a variety of architectures, including multi-core processors, graphics processing units (GPUs), and distributed computing environments. Potent parallel processing requires careful coordination and communication among processors, which is necessary in the design and operation of HPC systems.

3.1 Types of Parallelism in High-Performance Computing

Let's dive into these different types of parallelism, which are key concepts in High-Performance Computing (HPC), while focusing on the example of Data Parallelism as shown in Figure[1]. Each type plays a unique role in how computational tasks are divided and processed concurrently.

- Data Parallelism
- Task Parallelism
- Pipeline Parallelism
- Bit-level Parallelism
- Instruction-level Parallelism
- Thread-level Parallelism
- Request-level Parallelism

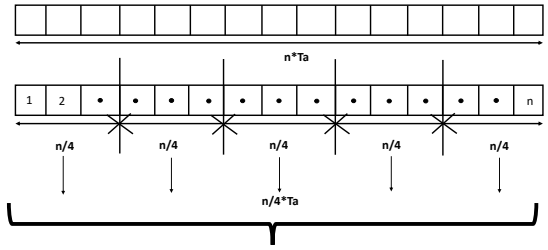

FIGURE 1. Data Parallelism

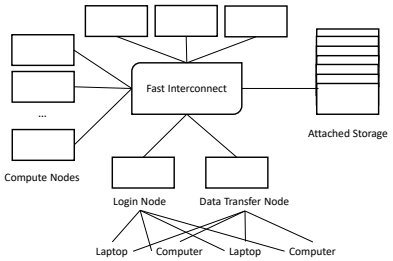

FIGURE 2. HPC Cluster Architecture

4. HPC Cluster Architecture

High-Performance Computing (HPC) clusters are complex assemblies of hardware and software components designed to perform large-scale computation tasks. The architecture of these clusters plays a major role in determining their computational capabilities and efficiency. Nodes, Interconnects, and storage devices are essential parts of HPC cluster architecture.

4.1 Nodes: The Building Blocks

- An HPC cluster is a collection of many separate servers (compute), called nodes.
- Connected via a fast interconnect.
- Different types of nodes for different types of tasks.

4.1.1 Compute Nodes

The workhorses of the HPC cluster Consisting of

- Processors (CPUs or accelerators like GPUs)
- Memory (RAM)
- Local storage

Each compute node operates independently to execute specific tasks assigned by the parallelism applications

4.1.2 Login Nodes

Entry points for users to access the cluster (while not involved in heavy computational tasks) Serve as the interface for

- Job submission
- File transfers
- Overall cluster management

4.1.3 Data Transfer Node (DTN)

- Dedicated systems deployed and configured specifically for transferring data over networks
- Place where data transfer applications run -> place where the user interface to the Science DMZ (= demilitarized zone) resides (most users interact with a Science DMZ through data transfers)

4.2 Interconnects: The Communication Backbone

Interconnects in an HPC cluster refer to the network systems that enable communication between nodes. The efficiency of data transfer within the cluster is critical for performance, especially for applications that require frequent data exchange between nodes.

4.2.1 High-Speed Interconnects

- Provide low-latency and high-bandwidth connections
- Facilitate efficient data exchange between nodes
- Especially important for applications that require frequent communication between different parts of the computation

4.2.2 Network Topology

Network Topology refers to the arrangement and interconnection of nodes within an HPC system. It dictates data flow patterns, affecting latency, bandwidth, and fault tolerance.

1. **Mesh and Torus:** In Figure [3] Offers multiple communication paths, enhancing fault tolerance and load balancing.

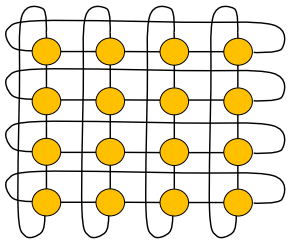

FIGURE 3. Torus Topology

2. **Hypercube:** In Figure [4] facilitates efficient routing and scalability, ideal for complex parallel computations.

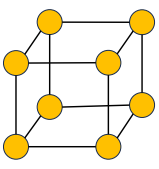

FIGURE 4. Hypercube Topology

3. **Fat Tree:** In Figure [5] features a hierarchical layout that optimizes bandwidth and minimizes network congestion.

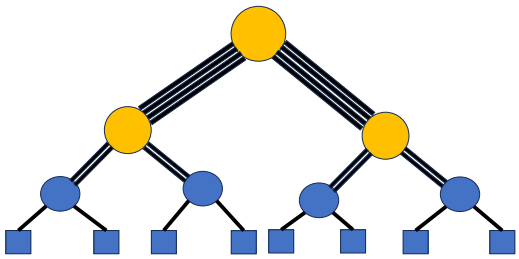

FIGURE 5. Fat Tree Topology

4. **Dragonfly:**In Figure [6] newer topology that reduces latency and cable length, is beneficial in large-scale systems.

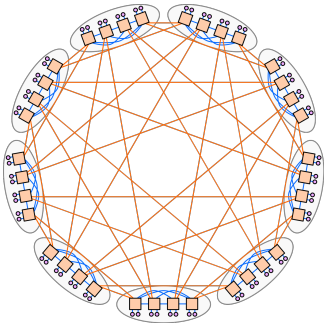

FIGURE 6. Dragonfly Topology

4.3 Storage Solutions: The Data Repositories

Storage solutions in HPC clusters are designed to provide high-speed, scalable, and reliable data storage, [3], [4].

There are typically two types of storage in HPC:

- local storage (internal to the nodes)
- shared storage (accessible to multiple nodes).

Shared storage is often implemented using technologies like Parallel File Systems, which allow multiple nodes to access and process data concurrently, thereby enhancing performance. The choice of storage technology impacts not only the data storage capacity but also the speed at which data can be written and read, which is vital for data-intensive tasks.

4.3.1 Parallel File Systems

- These file systems distribute data across multiple storage servers, allowing for simultaneous access by multiple nodes.
- This parallelization enhances data transfer rates, a critical factor for applications that generate or consume large datasets, [5].
- Example: Lustre in the Figure[7].

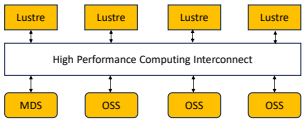

FIGURE 7. Parallel File System

5. Performance Metrics and Benchmarking in HPC

Performance metrics are quantifiable measures used to evaluate the capabilities and efficiency of HPC systems, [6], [7].

Scope: These metrics assess various aspects like processing speed, data handling, and resource utilization, providing insights into the system's overall performance.

Purpose:
Evaluation: Performance metrics are crucial for gauging the effectiveness of HPC systems in executing complex computational tasks.

Optimization: They help in identifying bottlenecks and areas for improvement, guiding optimization efforts to enhance system performance.

Comparison: Metrics also enables the comparison of different HPC systems and configurations, facilitating informed decisions in system design and upgrades.

5.1 Common Metrics

- FLOPS (Floating Point Operations Per Second)

  Description: In Figure[8] Measures the number of floating-point calculations an HPC system can perform per second, indicating its computational power.

  Usage: Widely used for benchmarking the raw processing capabilities of HPC systems.

Unit	Abbreviation	Floating Exp	FLOPS Decimal
Flops	FLOPS	10^0	1
Megaflops	MFLOPS	10^6	1.000.000
Gigaflops	GFLOPS	10^9	1.000.000.000
Teraflops	TFLOPS	10^{12}	1.000.000.000.000
Petaflops	PFLOPS	10^{15}	1.000.000.000.000.000
Exaflops	EFLOPS	10^{18}	1.000.000.000.000.000.000

FIGURE 8. Floating Point Operations Per Second

- IOPS (Input/Output Operations Per Second)

Explanation: In Figure[9]Quantifies the number of read/write operations a system can handle per second, reflecting its data handling capacity.

Importance: Critical for understanding the performance of storage systems and data-intensive applications.

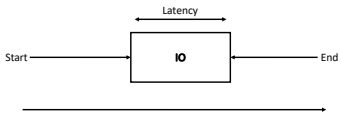

FIGURE 9. Input/Output Operations Per Second

5.2 Key Benchmarks in HPC

- Benchmarking in HPC involves executing a series of standard tests on computer systems to quantitatively assess their performance.
- Purpose: These tests provide measurable data on various aspects of system performance, such as speed, efficiency, and capacity.

5.2.1 Types of Benchmarks in HPC

- **LINPACK Benchmark**
  Developed in the 1970s by Jack Dongarra, it was originally used to benchmark a system's linear algebra computational abilities, specifically for solving dense systems of linear equations.

  Over time, LINPACK has become a standard for evaluating the performance of supercomputers and HPC systems.

  Often used to rank supercomputers in the TOP500 list.

  How LINPACK Works

  Procedure: LINPACK measures the speed of a computer in solving a dense N by N system of linear equations, which is a common task in engineering and scientific applications.

  FLOPS Measurement: The benchmark calculates the number of Floating Point Operations Per Second (FLOPS) the system can achieve, giving a clear picture of its computational power.

- **HPC Challenge Benchmark (HPCC)**
  The HPCC benchmark suite was developed as a more comprehensive set of tests compared to the LINPACK benchmark. It was designed to provide a broader understanding of HPC system performance.
  HPCC aims to examine a broader range of hardware and software capabilities beyond floating-point operations.

  Components of HPCC

  HPL (High-Performance LINPACK): A variation of LINPACK, measuring the systems peak floating-point performance.
  DGEMM: Measures the performance of double precision real matrix multiplication.
  STREAM: Assesses sustainable memory bandwidth and the corresponding computation rate.
  PTRANS (Parallel Matrix Transpose): Evaluates the rate of data transfer between different levels of memory hierarchy.
  Random Access: Measures the rate of integer updates to random locations in memory.
  FFT (Fast Fourier Transform): Tests the system's ability to handle complex, one-dimensional FFT computations.
  Communication bandwidth and latency: Benchmarks to measure inter-node communication capabilities.

- **Green 500**
  The Green500 List complements the traditional performance-focused TOP500 list by ranking supercomputers based on their energy efficiency, [8].

  It aims to promote a balance between computational power and energy consumption, highlighting the importance of environmentally sustainable HPC practices.

  Criteria for Ranking

  Energy Efficiency Measurement: Supercomputers are ranked based on their performance per watt, specifically measuring the number of floating-point operations they can perform per second per watt (FLOPS/Watt).

  Holistic Assessment: The list encourages the development of energy-efficient supercomputers without compromising on performance capabilities.

5.2.2 The Role of Benchmarking in HPC

Evaluating System Performance

- **Process:** By running standardized tests, benchmarking provides a quantitative measure of an HPC system's capabilities.
- **Outcome:** Helps identify strengths and weaknesses of the system in various computational tasks.

Comparing Different Systems

- **Benchmark Role:** Provides a common ground for comparing diverse HPC systems, irrespective of their architecture.
- **Informed Decisions:** Assists in choosing the right HPC system for specific research and computational needs.

6. HPC Challenges and Solutions

6.1 Performance commitments

Requirements:

- Customer asks to get a system with X Pflop/s Linpack performance.
- Customer asks to get a system with a certain Application performance.
- Customer asks for a system design to run a specific set of applications in a given time.

Issue:
How to calculate the performance of an application on a future system?

6.2 Performance Prediction & Extrapolation

Figure[10] provides a visual representation of the performance prediction and extrapolation for the HPC systems, illustrating the projected computational efficiencies under varying configurations.

- f: Processor clock speed
- BW: Per core memory bandwidth
- IB: Interconnect technology
- ref: Reference platform (e.g., benchmarking system)
- α_{CPU}: Part of the code that depends on processor speed
- α_{MEM}: Part of the code that depends on memory bandwidth
- $\alpha_{\text{MPI+IO}}$: Part of the code that depends on communication speed and bandwidth

$$T_{target}(f, BW, IB) = T_{CPU} + T_{BW} + T_{MPI+IO}$$

$$T_{CPU} = \alpha_{CPU} \frac{f_{ref}}{f_{target}} T_{ref} \left((1 - \varepsilon_{AVX2}) + \varepsilon_{AVX2} \frac{\eta_{op\_ref}}{\eta_{op\_target}} \right)$$

(non AVX2 part; AVX2 part)

$$T_{BW} = \alpha_{BW} \frac{BW_{ref}}{BW_{target}} T_{ref}$$

$$T_{MPI+IO} = \alpha_{MPI+IO} T_{ref} \left((1 - \eta_{IB\_BW}) + \eta_{IB\_BW} \frac{IB_{ref}}{IB_{target}} \right)$$

(latency part; IB BW part)

FIGURE 10. Performance Prediction Calculation [9]

6.3 Impact of DLC Cooling on HPC Applications

Observations from Figure[11] are as following:

- Application runs faster (6.5%) due to better use of Turbo Mode.
- Application uses less average power no fans in DLC system; offsets even higher power draw due to higher clock frequency in turbo mode.
- Significant (82%) lower total energy consumption of a DLC solution.

6.4 Impact of Liquid Cooling: Nvidia Example

The efficacy of liquid cooling over traditional air-cooling methods is depicted in Figure[12], where the Nvidia example demonstrates a 66% reduction in space usage and a 28% decrease in power consumption. Notably, the liquid-cooled solution achieves a significantly improved PUE (Power Usage Effectiveness) of 1.15 compared to 1.6 for the air-cooled solution.

NAMD v2.9 (ApoA1) E5-2690 CPU	Air Cooling	Direct Liquid Cooling	Difference
Application Wall time	63min 21sec	59min 29sec	**6.5%**
Average Power	491 Watt	425 Watt	**15.5%**
Consumed Energy	0.518 kWh	0.421 kWh	**23%**
Est. Cooling PUE	1.55	1.1	**50%**
Est. Total consumed Energy	0.8 kWh	0.44 kWh	**~82%**

FIGURE 11. DLC Cooling on HPC Applications(source: Andrey Semin[10])

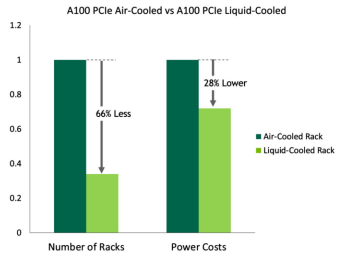

FIGURE 12. Nvidia Example: Liquid Cooling(source: NVIDIA[11]) [9]

6.5 TCO of cooling solutions

The economic and infrastructural aspects of water cooling, including its higher initial cost and infrastructure requirements, alongside the substantial TCO savings, are illustrated in Figure[13].

Sample System (96 Nodes – 24 DoubleTwin)	Air Cooling	Water Cooled Door (cold water)	Direct Liquid Cooling (warm water)
Est. Purchase Price (Investment)	1.024.000 € (1000k + 12x 2k)	1.080.000 € +5,5% (1000k + 4x 20k)	1.100.000 € +7,4% (1000k + 100k)
#Racks (kW per Rack)	**12x** (12kW)	**4x** (35kW)	**1x** (140kW)
PUE	1,7	1,4	1,03
Average System Power	132 kW	140 kW	117 kW
Energy Cost per year (incl. cooling) 0.25€ per kWh	493.223 €	430.712 €	264.368 €
Total Cost over lifetime (5y)	3.490.115 €	3.233.558 € - 7,4%	2.421.840 € - 30,6%

FIGURE 13. TCO of cooling solutions [9]

7. Conclusion

In this study, we delved into the complex world of High-Performance Computing (HPC), shedding light on the critical aspects of system design and its implications for efficiency and performance. Our journey through the intricacies of HPC revealed the paramount importance of harmonizing hardware choices with the specific needs of applications. This alignment is not just a technical requirement; it is the heartbeat of a high-performing computing system.

One of the most striking revelations from our research is the rising importance of environmental considerations in the design and operation of HPC systems. In an era where

sustainability is no longer optional, we have shown that marrying computational prowess with energy efficiency is both a challenge and an opportunity. Our discussions around metrics like Power Usage Effectiveness (PUE) and Total Cost of Ownership (TCO) are not just academic exercises; they are a call to action for building more sustainable computing infrastructures.

Our exploration of parallelism in HPC took us through a variety of methods, each with its unique strengths and applications. This journey underscored the diverse and innovative approaches necessary to harness the full potential of computing resources, especially when tackling the ever-growing complexity of data and computation in various fields.

Looking ahead, the path is clear for more groundbreaking work in enhancing the efficiency and sustainability of HPC systems. Areas like advanced cooling technologies and the development of more efficient parallel processing algorithms are ripe for exploration. Furthermore, as HPC becomes more accessible, it opens up a new world of possibilities in various sectors, from large-scale research to smaller, more localized applications.

In closing, this paper is more than just a scholarly contribution; it is a reflection of our continuous quest to push the boundaries of what's possible in HPC. The journey of HPC is far from over, and its evolving narrative will undoubtedly be a cornerstone in shaping the future of technology and research.

Acknowledgement

I want to express my deepest appreciation to Dr.Thomas Warschko. Throughout this journey, Dr.Warschko's perspective on balancing computational power with energy efficiency and cost-effectiveness has not only guided this study but also illuminated the path for future sustainable technological advancements in HPC. His commitment to addressing the critical aspects of Power Usage Effectiveness (PUE) and Total Cost of Ownership (TCO) in HPC systems has significantly contributed to the depth and breadth of this work.

This research, aiming to bridge the knowledge gap in HPC system design, has benefited immensely from his experience and foresight. His contributions have been essential in shaping the study's direction and ensuring its relevance in the ever-evolving landscape of high-performance computing. For his mentorship, unwavering support, and invaluable contributions, I am profoundly thankful.

I would also like to express my profound appreciation to Prof. Dr. Helena Liebelt for providing me with the opportunity to delve into and present on this significant topic. Her decision to assign a mentor for this project was instrumental in allowing me to explore this field in greater depth. Her support and belief in my capabilities have opened new avenues for me in the realm of High-Performance Computing, for which I am deeply grateful.

References

[1] *Top500 supercomputer sites*. https://www.top500.org/.

[2] Hager, Georg and Gerhard Wellein: *Introduction to High Performance Computing for Scientists and Engineers*. CRC Press.

[3] Carns, Philip *et al.*: *Understanding and improving computational science storage access through continuous characterization*. Transactions on Storage.

[4] Schwan, Philip: *Lustre: Building a file system for 1,000-node clusters*. In *Proceedings of the Linux Symposium*.

[5] Weil, Sage A. *et al.*: *Ceph: A scalable, high-performance distributed file system*. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*.

[6] Luszczek, Piotr *et al.*: *Introduction to the hpc challenge benchmark suite*. Technical report, ICL Technical Report.

[7] Dongarra, Jack *et al.*: *A proposal for a new set of parallel benchmarks for hpc clusters*. International Journal of High Performance Computing Applications.

[8] Patterson, Mark K. *et al.*: *The green500 list: Encouraging sustainable supercomputing*. Computer Journal.

[9] Warschko, Thomas: *Personal communication*. thomas.warschko@atos.net.

[10] Semin, Andrey: *Energy efficient supercomputers*, 2013.

[11] NVIDIA: *Liquid-cooled gpus*, 2022.