High Performance Computing

Course Survey Paper

Sri Keerthi Yenumula
Sunavya Varagani
Saqlain Patel Mohammed
Saaketh Garlapati

srikeerthi.yenumula01@student.csulb.edu
sunavya.varagani01@student.csulb.edu
aqlainpatel.mohammed01@student.csulb.edu
saaketh.garlapati01@student.csulb.edu

Advance Computer Architecture CECS 530
Computer Engineering and Computer Science Department
CSULB
Fall-2022

Instructor:
Dr. Maryam S. Hosseini

May 11, 2024

Abstract

The demand for high-performance computing, or HPC, is rising as a result of the increasing computational demands in several fields, including science, physics, chemistry, and biology, as well as Big Data, AI, and Data Science. A significant amount of processing power is required for these sectors' complex algorithms. Choosing the best computer system to improve performance is essential. The main issues that are anticipated for supercomputers in the future are examined in this abstract, with an emphasis on the areas where HPC resources are most needed. Additionally, it looks at new designs such as quantum processors and heterogeneous processors with AI chips, as well as the growing use of cloud-based HPC systems. It also tackles the challenges that software developers have when trying to parallelize applications in an efficient manner. Lastly, non-functional needs like resilience and energy efficiency are covered in detail in the abstract.

Contents

Chapter 1

Introduction

The landscape of High-Performance Computing (HPC) has recently undergone significant transformations, driven by advancements in technology and evolving demands. Here are some key trends shaping the future of HPC:

Exascale Computing: We are now in the exascale era, where supercomputers can perform at least one exaflop (10^{18} floating-point operations per second).These systems enable breakthroughs in scientific research, weather modeling, and complex simulations.

HPC Integration with AI and ML: Incorporating artificial intelligence (AI) and machine learning (ML) into HPC models has been revolutionary. Researchers and practitioners utilize AI/ML to optimize algorithms, enhance accuracy, and expedite scientific discoveries.

Quantum Computing: While still in its early stages, quantum computing holds vast potential for solving complex problems beyond classical computers' capabilities. Quantum algorithms could revolutionize fields like cryptography, material science, and optimization.

Portability and Productivity: HPC users increasingly demand portability across various architectures and platforms. Efforts are underway to develop tools and frameworks facilitating seamless application migration while preserving performance and productivity.

Interdisciplinary Collaboration: HPC now extends beyond specific scientific domains. Collaborations between researchers, industry experts, and policymakers drive innovation across diverse sectors, including automotive, finance, healthcare, and manufacturing.

Cloud-based HPC solutions are gaining traction, with cloud providers building global networks of systems tailored for HPC workloads, offering scalability, accessibility, and cost-effectiveness. Moreover, advancements in cybersecurity, thermal management, and edge computing are shaping HPC's future.

The convergence of HPC, AI, and other emerging technologies presents exciting opportunities for scientific breakthroughs, economic growth, and societal impact. As we continue to push boundaries, HPC remains a vital facilitator of progress across various domains.

Cloud-based Accessibility

Cloud-based HPC resources offer accessibility without the need for physical infrastructure investment. Users can scale resources dynamically, optimizing usage and costs with pay-per-use pricing models, making it more economical than traditional setups.

Accessible to All

Cloud-based HPC democratizes access to powerful computing resources, extending beyond specialized users. This accessibility fosters innovation across various fields by making HPC available to a wider audience.

Diverse Architectures

Modern supercomputers integrate various hardware accelerators such as GPUs, FPGAs, PIM, and quantum processors. Maximizing performance across these architectures requires new parallel programming approaches and libraries.

Chapter 2

The Need for HPC

2.1 Demands from Big Data area

Big Data Challenges The term "Big Data" was coined in 1997 by Michael Cox and David Ellsworth in their seminal work "Managing Big Data for Scientific Visualization" presented at the 1997 Conference on Visualization. This pioneering study shed light on the rapid proliferation of structured and unstructured data globally, marking a significant milestone in understanding the complexities and opportunities within this expansive data landscape.

Here are some significant facets of Big Data:

Exponential Growth

The generation of global data is doubling every two years, reaching extraordinary scales such as "Yotta data" (10^{24} zettabytes). Sources like Internet of Things (IoT) sensors contribute to this explosion, resulting in vast datasets that require specialized systems for efficient management, analysis, and insights.

Data Complexity

Data Complexity: Big Data encompasses massive and intricate datasets that conventional database tools struggle to handle. These datasets often amalgamate legacy data with real-time streams, posing substantial processing challenges. The sheer volume and intricacy necessitate substantial computing power to derive meaningful insights.

Transforming Raw Data to Knowledge

Raw data comprises numbers and codes devoid of inherent meaning. It is through processing and analysis that data evolves into valuable information. Ultimately, knowledge represents the pinnacle of understanding, where information is utilized for deeper insights and informed decision-making.

Societal Impact

Companies equipped with extensive data resources and sophisticated analysis capabilities wield remarkable influence. Their reach extends globally as they gather data from diverse sources. Leveraging this data for insightful analysis provides a competitive advantage and contributes to societal advancements.

High-Performance Machines

In this data-centric era, high-performance machines play a pivotal role. They empower us to harness the potential of data and unlock novel possibilities. Embracing innovation and leveraging state-of-the-art technologies will equip us to address the formidable challenges posed by Big Data.

Big Data's influence is profound, shaping industries, decision-making processes, and our understanding of the world. As we navigate this data-rich landscape, HPC remains an indispensable tool for extracting knowledge from the vast expanse of information.

2.2 Demands from the Artificial Intelligence Area

The US Department of Energy's "AI for Science" publication asserts that the integration of AI and Exascale computing signifies a significant shift in scientific exploration, marking the dawn of a new era characterized by innovation and discovery. By embracing this fusion, scientists can expedite progress towards a promising future by fully leveraging the capabilities of Exascale systems.

Simulation presents captivating opportunities through the amalgamation of Artificial Intelligence (AI) and High-Performance Computing (HPC). This dynamic combination empowers scientists to address complex challenges and achieve notable breakthroughs across various scientific disciplines. The convergence of AI and HPC is reshaping the landscape of scientific simulation, propelling advancements and shaping a future where simulations play a pivotal role in both scientific discovery and technological innovation.

While machine learning (ML) and deep learning (DL) techniques are revolutionizing numerous fields, a common bottleneck arises during the training phase. Due to resource constraints, this critical learning stage—essential for model performance and accuracy—may extend over weeks or months. In such scenarios, High-Performance Computing (HPC) serves as a catalyst, significantly expediting the learning process and facilitating accelerated development and implementation of AI models.

The escalating demand for High-Performance Computing (HPC) resources in AI development has prompted investments in parallel frameworks by companies like NVIDIA and Intel. These frameworks harness the immense processing power of GPUs and other specialized hardware to optimize the execution of AI software. The symbiotic relationship between HPC resources and AI development underscores the crucial role of parallel frameworks in accelerating progress.

As High-Performance Computing (HPC) and artificial intelligence (AI) increasingly converge, hardware manufacturers and software developers encounter unique challenges. However, this convergence also holds immense potential for driving technological innovation and scientific discoveries. Through collaborative efforts to address issues related to efficient hardware utilization and specialized processor design, hardware vendors and developers can ensure widespread accessibility and effectiveness of this transformative technology.

2.3 Demands from Data Science and Related Areas

The landscape of scientific research and innovation has undergone a profound transformation due to the intersecting rapid advancements in supercomputing technology and the exponential growth of data production technologies. This convergence has heralded the era of scalable data science, where vast volumes of data are swiftly processed and analyzed to expedite scientific inquiry and facilitate groundbreaking discoveries. The fusion of HPC and data science, propelled by advancements in supercomputing and data production technology, has reshaped scientific research, highlighting the indispensable role of scalable data science in advancing science and fostering innovation across various sectors. The potential of scalable data science is expected to drive even more remarkable discoveries and innovations in the future as technology continues to evolve.

In specific subject domains, data science represents a multidisciplinary approach that integrates various disciplines such as mathematics, statistics, machine learning, artificial intelligence, and specialized programming. This synergistic amalgamation enables the extraction of crucial information and hidden insights from an organization's data.

High-performance computing (HPC) systems now serve as the driving force accelerating all phases of the data science lifecycle, uncovering discoveries that propel society forward across diverse fields. HPC is reshaping data-driven decision-making in various sectors, spanning from fraud and hazard detection to the enhancement of search engine algorithms, advanced image and speech recognition, and even the planning of airline routes.

Despite the significant strides made in performance brought forth by HPC, optimizing the utilization of hardware resources remains a formidable challenge in the realm of data science. Efficiently minimizing the training costs of machine learning algorithms while maintaining exceptional performance is imperative to ensure the widespread adoption and accessibility of data science tools and methodologies.

Chapter 3

HPC Architectures and Processors Challanges

3.1 New Generation of Processors and Accelerators

The domain of parallel processing architectures is currently undergoing a wave of significant innovations, with a number of promising new designs poised to revolutionize the processor industry. At present, there are four supercomputers that stand out for their exceptional computational capabilities:

-Fugaku* in Japan: Equipped with ARM's A64FX processors and NVIDIA GPUs, Fugaku has maintained the number one position on the TOP500 list for two years in a row, 2020 and 2021, with a peak performance of 442 Petaflops. - *Frontier*: The first supercomputer to break through the 1 Exaflop threshold, Frontier was developed by Cray-HPE and is located at Oak Ridge National Laboratory in the USA. It boasts an impressive 1.1 Exaflops performance on the TOP500 list, marking a historic moment in high-performance computing. -*Aurora*: On the horizon is Aurora, currently in development at Argonne National Laboratory and utilizing Intel's Ponte Vecchio architecture. This advanced technology combines Xeon processors and Xe accelerators on a single chip, aiming for even higher performance and efficiency. - *El Capitan* at Lawrence Livermore National Laboratory: Preparing to join the elite group of exascale computers, El Capitan is being designed with AMD's upcoming EPYC processors, known as "Genoa" and featuring the Zen 4 processor core, as well as AMD Radeon Instinct GPUs. It is expected to surpass the 2 Exaflop mark, pushing the limits of high-performance computing even further.

With the Exaflop barrier now crossed, the international high-performance computing scene is buzzing with activity. In addition to the aforementioned giants, several other Exascale systems are scheduled for launch in the near future. The US Department of Energy is leading this initiative, providing funding for national computing centers to acquire state-of-the-art systems from various companies, each displaying their own unique architectural breakthroughs.

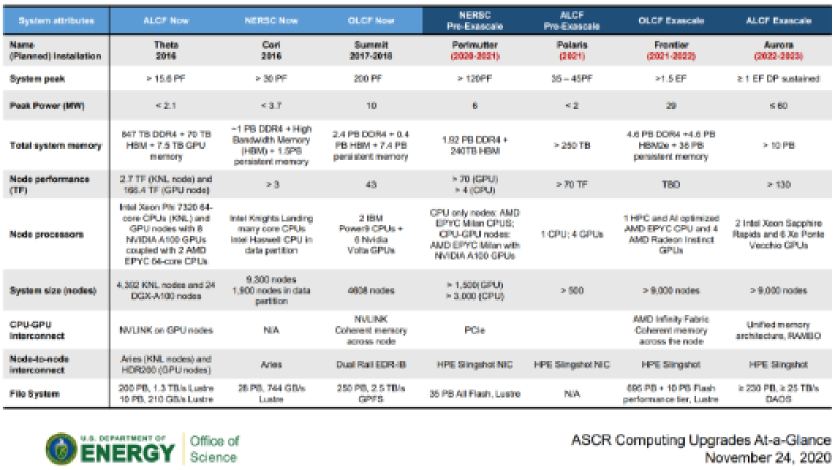
A timeline of these impending installations is depicted in Figure 3.2, beginning with Perlmutter at NERSC-Berkeley, followed by Polaris at ALCF-Argonne, and culminating with Aurora at ALCF-Argonne. Moreover, China and Japan are vigorously advancing their Exascale projects with ongoing development of the Sunway, Tianhe, and Fugaku supercomputers.

This international competition for dominance in supercomputing heralds a new era in the progression of High-Performance Computing (HPC). As these formidable systems become operational, they will provide researchers and scientists from various fields with unparalleled computational resources, paving the way for revolutionary discoveries and innovations in a multitude of disciplines.

Figure 3.1: : Fugaku, Frontier, Frontier, and El Capitan supercomputers

3.2 Heterogeneous Architectures

Emerging processors entering the market are progressively integrating heterogeneous architectures. In the realm of high-performance computing (HPC), the amalgamation of processors with varying computational capabilities yields enhancements in both performance and energy efficiency compared to homogeneous systems, such as those solely reliant on CPUs.



System attributes	ALCF Now	NERSC Now	OLCF Now	NERSC Pre-Exascale	ALCF Pre-Exascale	OLCF Exascale	ALCF Exascale
Name (Planned) Installation	Theta 2016	Cori 2016	Summit 2017-2018	Perlmutter (2020-2021)	Polaris (2021)	Frontier (2021-2022)	Aurora (2022-2023)
System peak	> 15.6 PF	> 30 PF	200 PF	> 120PF	35 – 45PF	>1.5 EF	≥ 1 EF DP sustained
Peak Power (MW)	< 2.1	< 3.7	10	6	< 2	29	≤ 60
Total system memory	847 TB DDR4 + 70 TB HBM + 7.5 TB GPU memory	~1 PB DDR4 + High Bandwidth Memory (HBM) + 1.5PB persistent memory	2.4 PB DDR4 + 0.4 PB HBM + 7.4 PB persistent memory	1.92 PB DDR4 + 240TB HBM	> 250 TB	4.6 PB DDR4 +4.6 PB HBM2e + 36 PB persistent memory	> 10 PB
Node performance (TF)	2.7 TF (KNL node) and 166.4 TF (GPU node)	> 3	43	> 70 (GPU) > 4 (CPU)	> 70 TF	TBD	> 130
Node processors	Intel Xeon Phi 7320 64-core CPUs (KNL) and GPU nodes with 8 NVIDIA A100 GPUs coupled with 2 AMD EPYC 64-core CPUs	Intel Knights Landing many core CPUs Intel Haswell CPU in data partition	2 IBM Power9 CPUs + 6 Nvidia Volta GPUs	CPU only nodes: AMD EPYC Milan CPUS; CPU-GPU nodes: AMD EPYC Milan with NVIDIA A100 GPUs	1 CPU; 4 GPUs	1 HPC and AI optimized AMD EPYC CPU and 4 AMD Radeon Instinct GPUs	2 Intel Xeon Sapphire Rapids and 6 Xe Ponte Vecchio GPUs
System size (nodes)	4,392 KNL nodes and 24 DGX-A100 nodes	9,300 nodes 1,900 nodes in data partition	4608 nodes	> 1,500(GPU) > 3,000 (CPU)	> 500	> 9,000 nodes	> 9,000 nodes
CPU-GPU Interconnect	NVLINK on GPU nodes	N/A	NVLINK Coherent memory across node	PCIe		AMD Infinity Fabric Coherent memory across the node	Unified memory architecture, RAMBO
Node-to-node interconnect	Aries (KNL nodes) and HDR200 (GPU nodes)	Aries	Dual Rail EDR-IB	HPE Slingshot NIC	HPE Slingshot NIC	HPE Slingshot	HPE Slingshot
File System	200 PB, 1.3 TB/s Lustre 10 PB, 210 GB/s Lustre	28 PB, 744 GB/s Lustre	250 PB, 2.5 TB/s GPFS	35 PB All Flash, Lustre	N/A	695 PB + 10 PB Flash performance tier, Lustre	≥ 230 PB, ≥ 25 TB/s DAOS

U.S. DEPARTMENT OF ENERGY | Office of Science

ASCR Computing Upgrades At-a-Glance
November 24, 2020

Figure 3.2: : Evolution of supercomputers

As we go toward the future of High-Performance Computing (HPC), we see a discernible trend toward heterogeneous architectures in addition to the classic CPU-GPU and FPGA-based systems (Figure 3.5). Heterogeneous memory systems are designed to balance performance and energy economy for applications that require a lot of data. They do this by using a combination of memories, such as DRAM and SRAM. Artificial Intelligence (AI) operations can be expedited by specialized processors called Neural Processing Units (NPUs), while quantum processors—still in the early phases of development—have the potential to outperform traditional computers in calculations.

Systems that combine CPUs and GPUs onto a single chip are becoming more and more common in the market. Examples of these are AMD's Accelerated Processing Units (APUs) and Intel's Ponte Vecchio. Another design is the System on Chip (SoC), which is especially common in sensor, Internet of Things (IoT), and edge computing contexts. In this configuration, processors, accelerators, memory, and I/O devices coexist on the same chip. As seen in the Intel A10, heterogeneity can also be attained at the board level when a processor coexists with a GPU or FPGA.

Programming models have to change from being optimized for a single accelerator type to fitting within a programming environment that can support several accelerators as the computing landscape grows more diversified with the addition of numerous accelerators. For example, new parallel programming interfaces are required with the release of GPUs from AMD and Intel, rather than depending on NVIDIA-only proprietary languages like CUDA.

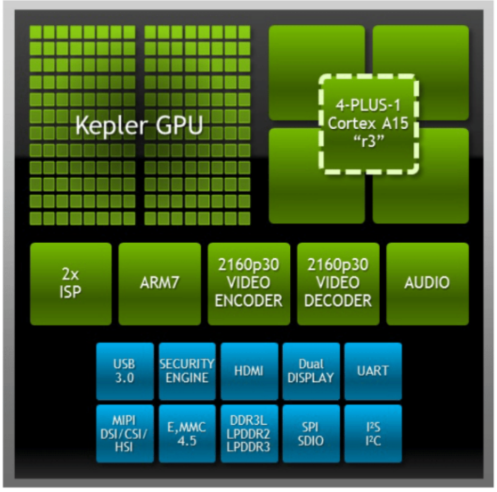


Figure 3.3: NVIDIA Tegra K1 chip integrating processors with GPUs in an SoC

3.3 AI chips

Artificial intelligence (AI) is predicted to become more and more important in the next years in a number of fields, most notably national and international security. However, modern AI software, datasets, and algorithms cannot be efficiently executed on conventional High-Performance Computing (HPC) equipment. As a result, attention has shifted to specialized computer hardware made specifically to run cutting-edge AI applications.

This specialized hardware, sometimes called "AI chips," includes accelerators designed specifically for AI activities, such as GPUs, FPGAs, and application-specific integrated circuits (ASICs). Even if general-purpose processors, such as CPUs, are capable of handling basic AI tasks, their usefulness diminishes as AI technologies progress. Artificial intelligence (AI) processors are designed with features that are optimized to speed up the computations needed by AI algorithms. These features include high levels of parallel processing, mixed precision for quicker memory access, and programming languages that are efficient for running AI code.

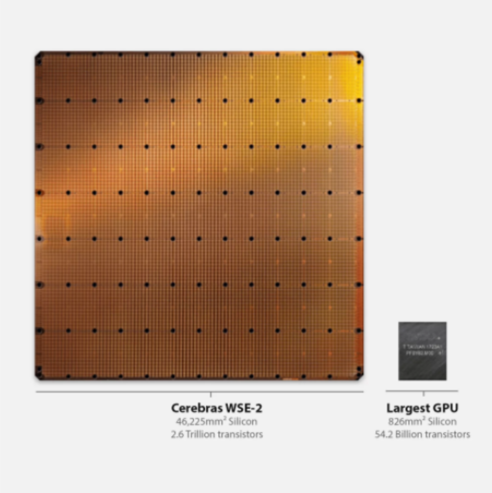


Figure 3.4: Comparison of Cerebras WSE-2 with the largest GPU at the moment

It can be difficult to choose the right AI chip for a particular AI algorithm. FPGAs are superior in inference, although GPUs are usually favored during the training stage. However, ASICs can function in both stages. The 2020 SuperComputing conference recently unveiled Cerebras and SambaNova, two new AI processors. Driven by the second generation of Wafer Scale Engine (WSE-2), Cerebras chips feature impressive specifications like 40 gigabytes of high-performance on-wafer memory, 850,000 AI-optimized cores, and 2.6 trillion transistors. These chips serve as central processors for deep learning and sparse tensor operations.

The Reconfigurable Dataflow Architecture (RDA) of the SambaNova microprocessor allows machine learning and high-performance computing to come together (HPC). Although it is programmed for each unique model rather than having a fixed Instruction Set Architecture (ISA), the RDA provides a flexible dataflow execution paradigm that may be adjusted to different compute issues.

In summary, the growing adoption of AI chips with specialized hardware is poised to revolutionize the execution of AI algorithms in the coming years. However, this shift may pose challenges in integrating such chips into traditional HPC systems and could lead to the development of new AI frameworks.

3.4 Aware Computing

Aware Computing has recently gained significant traction in HPC systems, employing a variety of optimization techniques. These methods dynamically adjust hardware and software parameters during program execution based on optimization heuristics, such as thread count, CPU frequency, and memory utilization. By striving to achieve the optimal state, this optimization aims to enhance non-functional metrics like power consumption, energy efficiency, and performance. Consequently, the landscape of hardware and software behavior is evolving in response to these changes.

The demand for increasingly high performance in supercomputers has fueled the adoption of techniques like power and energy-aware computing. The objective is to boost performance while keeping power and energy consumption in check, thus reducing expenses related to electrical infrastructure and cooling. Power and energy-aware computing employs strategies to adapt software and hardware configurations to maintain energy and power consumption below predefined thresholds. Dynamic voltage and frequency scaling, a commonly used hardware approach, automatically adjusts the voltage and frequency levels of hardware components based on workload usage. Alternatively, thread-throttling adjusts the number of active threads based on the thread-level parallelism of an application.

Similarly, the rise of memory-aware computing is driven by the growing memory requirements of emerging HPC applications in scientific fields. This model optimizes computer system architecture for memory performance, considering the widening gap between processor speed and memory latency. Memory-aware computing encompasses techniques such as data compression, prefetching, memory hierarchy optimization, and efficient data allocation. Applications reliant on memory, such as scientific simulations, machine learning, and big data analytics, face significant challenges under this computing model. The AMD High Bandwidth Memory (HBM) is an example tailored to deliver high-performance memory for memory-intensive workloads like graphics.

Aware computing manifests in various forms within HPC servers. Network-aware computing aims to maximize network performance through congestion control, load balancing, and network topology optimization. Data-aware computing focuses on efficient data handling, while security-aware computing prioritizes system security. Additionally, user-aware computing aims to provide personalized user experiences.

3.5 Processing In Memory

The groundbreaking method called Processing In Memory (PIM) eliminates the necessity of transferring data to the processor for executing instructions. This eliminates the time wasted in data transport, presenting a significant advantage. While research on PIM has expanded and processors based on this technology are increasingly prevalent, it also introduces new challenges for software developers and system architects working in high-performance computing (HPC).

Despite recent advancements in Processing-In-Memory (PIM) architectures, several questions remain unanswered. One such challenge is how HPC programmers can leverage the benefits of PIM without resorting to complex programming paradigms. Understanding the constraints of different substrates when designing PIM logic is another challenge. Therefore, before most HPC developers can efficiently utilize PIM, several daunting tasks need resolution, including designing a PIM programming model, data mapping, and runtime scheduling.

Leading high-tech companies in memory development with computational capabilities include Samsung, Micron, and Synopsys. They envision a future where AI computing and memory merge into a unified architecture, giving rise to AI-based memory chips.

3.6 Quantum Computing

As quantum processors become tangible, we anticipate the integration of heterogeneous systems featuring both conventional processors and quantum processing units. Breakthroughs in technology, such as the ability to create qubits using techniques like germanium transistors, make it feasible to introduce atom-sized operators. This advancement enables the potential inclusion of qubit processors alongside conventional x86 processor units in future architectures, as depicted in Figure 3.5.

Initially, quantum processors are poised to serve as accelerators in computational tasks. They hold promise across diverse fields, including security, cryptography, meteorology, pharmaceuticals, biotechnology, and economic modeling. These fields pose intricate challenges for classical computers, which quantum computing could address more efficiently.

However, it's essential to recognize that quantum computing is not set to replace classical computing entirely. Rather, it will complement classical computing by tackling complex problems that are particularly time-consuming for classical systems. Examples include modeling protein or molecular simulations, developing robust encryption methods, processing data from particle accelerators like CERN, and addressing various other intricate computational tasks.

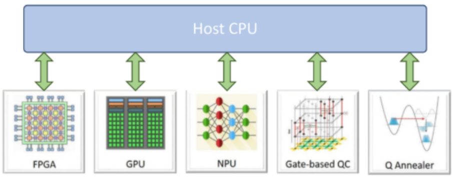


Figure 3.5: Future of Heterogeneous Architectures with Quantum Processors

3.7 Cloud Computing

Significant cloud computing providers have started providing High-Performance Computing (HPC) services in response to the growing demand for these services across a variety of industries. There have been reports of the appearance of cloud instances with more potent CPUs, GPUs, FPGAs, and better networking systems. Thanks to this development, users can set up a cluster of computers that can handle more demanding processing tasks.

In this environment, High-Performance Computing as a Service (HPCaaS) is becoming more and more popular. It functions as the cloud's infrastructure for running concurrent workloads, as seen in Figure 3.6. The cloud environment (HPCaaS) manages machine allocation and job deployment on High-Performance Computing (HPC) systems once users submit their workloads online.

Leading cloud providers emphasize the value of providing HPC capabilities in the cloud by heavily promoting their HPC products on their websites. They highlight characteristics like "High-Performance Computing on AWS Redefines What is Possible," "Cray in Azure - a dedicated supercomputer on your virtual network," "Build your high-performance computing solution on IBM Cloud," or "Google Cloud - HPC in the cloud becomes a reality."

Gradually, the early performance issues with cloud computing are being resolved, leading to good results when executing HPC applications in this setting. The processing power needed to handle complicated cloud applications is provided by improved storage management, next-generation processors, and improved connectivity. To enable HPC applications, cloud servers need to have an affordable and scalable infrastructure that allows for on-demand resource provisioning over the internet.

Additionally, consideration ought to be given to Serverless Computing, a cutting-edge method of creating cloud applications. In order to simplify complexity for consumers, this paradigm combines components of virtualization, containers, and Function as a Service (FaaS).

Chapter 4

Programming Challenges in HPC

This part delves into how design choices affect the execution of parallel applications on High-Performance Computing (HPC) servers. We commence by discussing design patterns for parallel algorithms and the programming interfaces that can be employed to maximize the advantages offered by HPC systems. Additionally, given the importance of memory in application execution, we explore strategies for optimizing data and thread locality.

4.1 Parallel Algorithms Design Patterns

When parallelizing applications, programmers can use a variety of communication models to make sure that cores operating concurrently cooperate with one another. These models include message-passing and shared memory. Thread-level parallelism, in which threads share the same memory address space, can be effectively used using shared memory, which depends on a memory address space that is available to all processors. On the other hand, distributed memory spaces or processes without shared memory addresses are more suited for message-passing. The difficult part is figuring out which programming model to use in relation to the target architecture. For multicore and many-core processors, shared memory works better, whereas message-passing works better for huge computers that communicate over interconnection links.

Apart from the communication model, another difficulty is deciding on the parallel programming style. Software developers have traditionally preferred the fork-join architecture because it makes utilizing parallelism easier. As seen in Figure 4.2, this architecture has a master thread that starts the sequential phase and forms a group of threads to run the parallel region simultaneously (fork operation). At the end of the region, all threads synchronize (join operation), and only the master thread keeps running the application until it meets another parallel region. Creating algorithms that grow the number of threads without sacrificing performance and energy efficiency is necessary to fully utilize architectures with more cores, such as those from AMD, ARM, Intel, and NVIDIA.

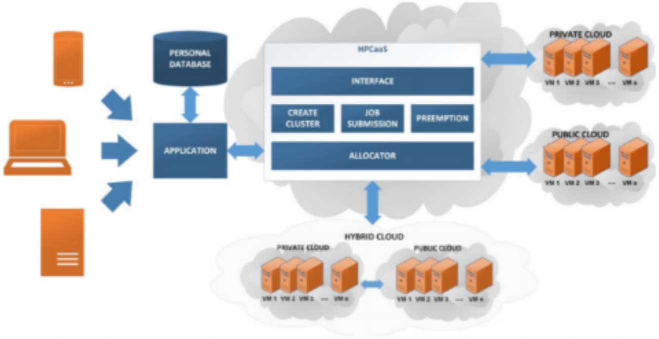


Figure 4.1: HPC as a Service on Cloud Computing

Rigid implementations, on the other hand, that rely on the fork-join model might not be able to cope with variations in the execution environment or application behavior, which could result in issues like cache congestion and data synchronization. This may lead to increased power usage and jeopardize parallel application performance. As a result, task-based programming approaches are becoming more and more common since they provide improved load balancing and greater flexibility on multicore systems. However, identifying data relationships across several parallel zones might be difficult for software engineers, which could limit the advantages of parallelization.

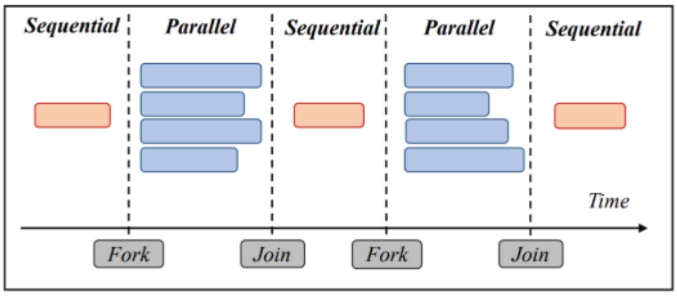


Figure 4.2: The fork-join shared-memory programming model

Software developers can use design patterns to make parallelization easier, regardless of the programming style they choose. There is widespread use of patterns including Reduction, Stencil, and Map. The Map pattern (Figure 4.3) splits the workload (e.g., array or list) into separate segments that can run in parallel without requiring any data—a feature known as embarrassed parallelism. All of the collection's elements are given a function, which usually results in a new collection whose shape is identical to the input. Emerging computer architectures, on the other hand, might call for applying intricate functions to every member of a collection via numerous iterations in order to fully utilize the processor's cores.

4.2 Parallel Programming Libraries

A number of programming languages and libraries have been established to help software developers take use of parallelism with the introduction of contemporary architectures that claim various processing capabilities. The progression of citations for the most popular parallel programming languages over time, as seen in Figure 11 of the Scopus database, will be discussed in more detail below.

The Message-Passing Interface (MPI) has become the go-to option for utilizing parallelism over time. The reason for its popularity is that all High-Performance Computing (HPC) systems require a library that can both create processes and control their communication in distributed memory environments. Challenges include balancing communication and computation between computing nodes to optimize hardware resource utilization, effectively using asynchronous communication to overlap communication and computation, and guaranteeing fault tolerance mechanisms. These are made more difficult by the evolution of MPI, from MPI-1 to MPI-3, and by technological advancements in HPC systems.

Simultaneously with the increase in core counts in multicore architectures, there has been a boom in the use of libraries like POSIX Threads and OpenMP that enable parallelism in shared memory settings. The first step toward developing parallel programs was taken by the POSIX Threads library, which is well-known for its efficient thread implementation. However, because of its complexity and the requirement for programmers to manage different operating system components, its use has decreased and is currently primarily restricted to creating operating system-level applications.

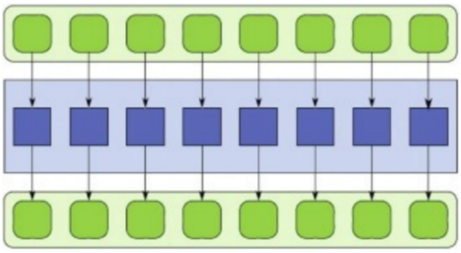


Figure 4.3: Map pattern example, where a function is applied to all elements of a collection, producing a new collection with the same shape as the input

Chapter 5

Other Challenges

The increasing diversity of hardware resources and the availability of frameworks and libraries for utilizing parallelism in High-Performance Computing (HPC) servers present new difficulties that need to be addressed, as we've covered in previous sections. We will examine new developments in energy consumption in this area, with an emphasis on elements that are vital to raising the energy efficiency of HPC systems. Furthermore, increasing resilience is essential as new HPC settings appear in order to lessen the effects of hardware and software failures. We will also examine two methods—mixed-precision computing and data localization strategies—that are intended to maximize the energy efficiency of HPC systems.

5.1 Energy Demand

Manufacturers have been forced to build clusters with thousands of processors due to the growing need for processing power in High-Performance Computing (HPC) systems, which has resulted in high power consumption. Some of the devices on the TOP 500 list have energy consumption of about 30 MW, which is comparable to the energy requirements of a city with about 300,000 people. As a result, chip and machine makers are concentrating on improving architectures to lower power usage. Among other things, these optimizations involve controlling inactive units, lowering CPU operating frequencies, and altering processor architecture. Nonetheless, present efforts to create machines with faster instruction execution times and lower energy usage may result in unexpected effects.

One of the difficulties in the evolution of processor architectures is shown in the picture below, which was created from data gathered on a contemporary multicore CPU while it was doing typical parallel tasks using the WattWatcher program. It demonstrates that the actual processing and execution of instructions only accounts for roughly 17 percentage of the overall energy (including out-of-order execution and arithmetic logic unit utilization). On the other hand, static energy, also known as leakage, accounts for about 34 percentage of the energy

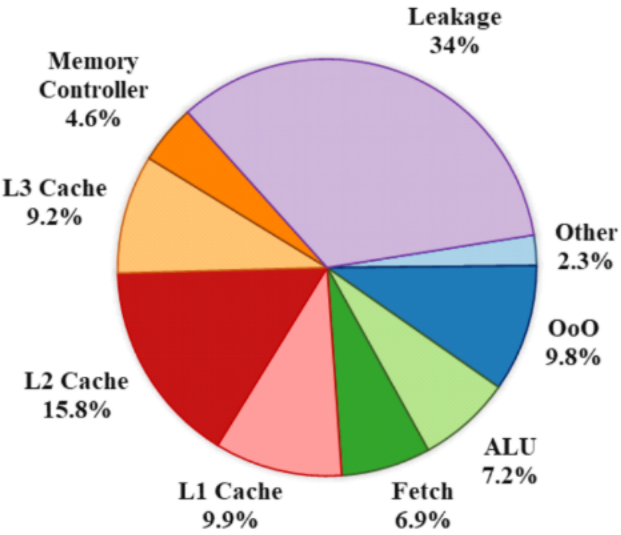


Figure 5.1: Distribution of energy consumption in a core

used by the circuit even when it is not in use.

Furthermore, registers consume 11 percentage of the energy, and about 35 percentage goes toward accessing different layers of cache memory. It is clear that the energy devoted to the main objective of instruction execution is negligible in comparison to other aspects of processor operation, even though these percentages may increase in the future.

5.2 Resilience

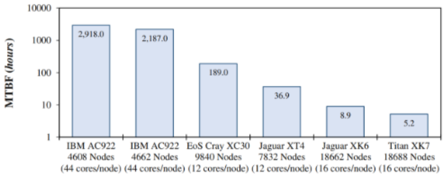


Figure 5.2: Mean time between failures on different HPC servers

The ability of a system to continue operating even in the face of malfunctions or performance variations is referred to as resilience. Component failure is more likely in the world of supercomputers, which are designed to handle high-performance tasks and have several cores, memory, and coupled circuits. Thus, maintaining resilience becomes an important obstacle to be addressed in order to keep scientific applications running at high performance levels while keeping infrastructure expenses to a minimum.

5.3 HPC and Mixed Precision

Because high-precision procedures cost a large amount of energy and time, researchers are increasingly concentrating on optimizing processes depending on the specific requirements of each computation. Reducing precision can save money, effort, and energy. Currently, the task at hand is to minimize accuracy while running the application.

Multiple floating-point precision arithmetic operations are often supported by mixed-precision architectures, which allows for lower computational, energy, and storage requirements. It is feasible to strike a compromise between the performance and energy efficiency of the execution and the quality of the outcome by decreasing the precision of some data and arithmetic operations.

Not only does approximate arithmetic reduce precision, but it also uses less energy and less space due to its simpler arithmetic units, which produce less accurate results.

5.4 Data Locality

Future microprocessors are anticipated to have a more complex memory hierarchy, which will have a major effect on application performance and energy consumption because data transportation alone requires a large amount of energy. Each nanojoule used to move data up and down the memory hierarchy reduces the amount of energy that may be used for processing. Because of this, task mapping and scheduling within the interconnection network must be optimized, with a focus on minimizing data movement and giving data proximity precedence over processor speed—even though local data typically enables faster processing. Under this scenario, energy conservation comes first.

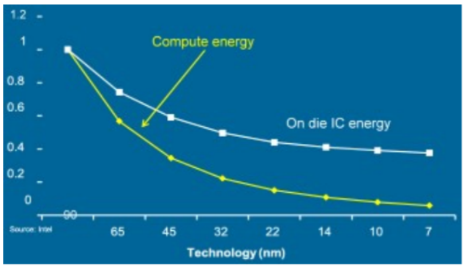


Figure 5.3: Energy consumption at Pico Joules versus technology evolution

The Department of Energy (DoE) report's data shows that even with improvements in chip technology—such as the switch to finer technologies like 7nm—the overall chip energy consumption has decreased less than the computational processes' energy usage. This pattern indicates that data transportation uses more power than doing computations on the device, indicating a major change in both CPU architecture and instruction execution. Compilers must so concentrate on maximizing data proximity to the processors.

Chapter 6

Conclusion

The evolution of High-Performance Processing (HPP) has shifted from fulfilling specialized processing requirements to becoming a pivotal driver in the advancement of computer technology. This shift is fueled by the increasing demand for processing power in fields like Big Data, AI, and data science. Simultaneously, there is a noticeable trend towards relying on cloud technologies to meet these demands, resulting in significant changes in machine and processor architectures. While enhancing performance remains a priority, there is a growing emphasis on minimizing energy consumption. Presently, modern processors and machines are distinguished by their heterogeneity, and the emergence of quantum processors is expected to further diversify computer architectures. In today's High-Performance Computing (HPC) systems, resilience is paramount, as system faults or interruptions can lead to adverse consequences such as data loss or system downtime, potentially resulting in financial losses and reduced productivity. To continue advancing HPC systems, innovative programming and storage techniques are essential. Overall, the trajectory of computing involves a continual increase in processing power, coupled with the use of innovative approaches to achieve this objective.

Bibliography

[1] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, “A survey on distributed machine learning,” ACM Computing Surveys, vol. 53, no. 2, p. 1–33, Mar. 2020.
http://dx.doi.org/10.1145/3377454

[2] Hager, Georg, and Gerhard Wellein. Introduction to High Performance Computing for Scientists and Engineers. CRC Press, 2010.

[3] S. Lee, S.-h. Kang, J. Lee, H. Kim, E. Lee, S. Seo, H. Yoon, S. Lee, K. Lim, H. Shin, J. Kim, O. Seongil, A. Iyer, D. Wang, K. Sohn, and N. S. Kim, “Hardware architecture and software stack for pim based on commercial dram technology : Industrial product,” in 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), 2021, pp. 43–56.

[4] R. Stevens, V. Taylor, J. Nichols, A. B. Maccabe, K. Yelick, and D. Brown, “Ai for science: Report on the department of energy (doe) town halls on artificial intelligence (ai) for science.” [Online].
https://www.osti.gov/biblio/1604756

[5] S. Matsuoka, J. Domke, M. Wahib, A. Drozd, and T. Hoefler, “Myths and legends in high-performance computing,” 2023.

[6] G. Freytag, J. V. F. Lima, P. Rech, and P. O. A. Navaux, “Impact of reduced and mixed-precision on the efficiency of a multi-gpu platform on cfd applications,” in Computational Science and Its Applications – ICCSA 2022 Workshops, O. Gervasi, B. Murgante, S. Misra, A. M. A. C. Rocha, and C. Garau, Eds. Cham: Springer International Publishing, 2022, pp. 570–587.

[7] E. H. M. Cruz, M. Diener, L. L. Pilla, and P. O. A. Navaux, “Online thread and data mapping using a sharing-aware memory management unit,” ACM Trans. Model. Perform. Eval. Comput. Syst., vol. 5, no. 4, jan 2021. [Online].
https://doi.org/10.1145/3433687