



A CPE 407 TERM PAPER

TOPIC:

PARALLEL COMPUTING

presented by

GROUP 10

LECTURER: DR OLANIYAN

2021/2022 Session

NAME	MATRIC NUMBER
Adekola Boluwaji Emmanuel(GROUP LEADER)	CPE/2018/1012
Dada Olalekan	CPE/2018/1050
Adeyemi Sunday Oluwaseyi	CPE/2018/1019
Adebayo Emmanuel Iyanuoluwa	CPE/2018/1004
Hamzat Utman Ayomide	CPE/2018/1116
Olatunji Olayide Nelson	CPE/2018/1089
Daodu Jude Ayomide	CPE/2018/1051
Onyemenam Miriam Uchechukwu	CPE/2018/1102
IBIYEMI ADEDAYO PRAISE	CPE/2018/1062
IGE JEREMIAH AYOMIPOSI	CPE/2018/1063

Introduction

Parallel computing is a method of computation in which many calculations or the execution of processes are carried out simultaneously. The goal of parallel computing is to increase the computational power and efficiency of a system by utilizing multiple processors or cores to perform tasks simultaneously. The concept of parallel computing has been around for decades, but it has only recently become more prevalent due to the advancement of technology and the increasing need for powerful computing systems in various fields such as scientific research, data analysis, machine learning, and gaming. The goal of parallel computing is to increase the speed and efficiency of computation, by taking advantage of the increased processing power available in modern computers. The working principle behind parallel computing is to divide a large computational task into smaller sub-tasks and to perform these sub-tasks simultaneously, on multiple processors or computers, to obtain results faster than if the task was performed on a single processor. The multiple processors work together, in parallel, to perform the overall computational task. Parallel computing enables faster processing times by allowing multiple processors to work on different parts of the task at the same time. This results in improved performance, as each processor can contribute its processing power to the overall task.

The history of parallel computing dates back to the early days of computing when scientists and engineers first realized the potential of using multiple processors to solve complex computational problems. In the 1950s, early parallel computers, such as the Illiac I, were developed for scientific and military applications. These machines were limited in their capabilities and were expensive to build and

maintain, making them accessible only to government and research institutions.

In the 1970s, the development of minicomputers, such as the PDP-11, and the introduction of the concept of time-sharing systems paved the way for the use of parallel computing in more general-purpose applications. In the following decades, the development of computer networks and the growth of the Internet increased the availability and accessibility of parallel computing. The rise of parallel computing in the scientific and engineering communities was fueled by the increasing demand for more computing power for simulations and modeling.

In the 1990s and early 2000s, parallel computing became more mainstream with the advent of high-performance computing (HPC) clusters and the introduction of parallel programming models and libraries, such as MPI and OpenMP. The development of multi-core processors and the growth of GPU computing further increased the importance of parallel computing in solving complex computational problems.

Today, parallel computing is a critical component of modern computing, providing the computing power and performance required for many scientific, engineering, and commercial applications. The growth of parallel computing has been driven by the increasing demand for more computing power, the growth of data-intensive applications, and the need for faster and more accurate solutions to complex computational problems. As the demand for more computing power continues to grow, parallel computing will play an increasingly important role in the future of computing.

Types of Parallel Computing

There are three main types of parallel computing:

- **Bit-level parallelism:** The form of parallel computing in which every task is dependent on processor word size. In terms of performing a task on large-sized data, it reduces the number of instructions the processor must execute. There is a need to split the operation into series of instructions
- **Instruction-level parallelism:** In a single CPU clock cycle, the processor decides in instruction-level parallelism how many instructions are implemented at the same time. For each clock cycle phase, a processor in instruction-level parallelism can have the ability to address that is less than one instruction.
- **Task Parallelism:** Task parallelism is the form of parallelism in which the tasks are decomposed into subtasks. Then, each subtask is allocated for execution. And, the execution of subtasks is performed concurrently by processors.

Each type of parallel computing has its own advantages and disadvantages, and the choice of architecture depends on the specific requirements of the application being developed.

Hardware for Parallel Computing

The hardware for parallel computing typically consists of the following components:

- **Multi-Core Processors:** Multi-core processors are the most common hardware components used in parallel computing. They contain multiple processing cores on a single chip, allowing for parallel processing of multiple tasks.
- **Graphics Processing Units (GPUs):** GPUs are specialized processors designed for parallel processing of large amounts of data, typically used for graphics rendering and scientific simulations.
- **High-Performance Computing Clusters:** High-performance computing clusters are groups of interconnected computers working together to solve complex problems. Clusters can range from a few interconnected computers to large-scale supercomputers consisting of thousands of nodes.
- **Interconnects:** Interconnects are the communication networks that connect the processing nodes in a parallel computing system. They are responsible for transmitting data between the processing nodes and allowing communication between processors.
- **Storage Systems:** Storage systems are used to store the large amounts of data generated by parallel computing applications. They can range from local storage on individual nodes to large-scale distributed storage systems.

- **Input/Output Devices:** Input/output devices are used to transfer data to and from the parallel computing system. They can range from simple disk drives to high-speed network interfaces.

The choice of hardware for a parallel computing system depends on the specific requirements of the application, including the amount of data to be processed, the number of processing nodes required, and the level of performance needed.

Programming Model for Parallel computing

Programming models are abstractions that simplify the process of developing parallel programs. Some of the commonly used programming models for parallel computing include:

- **Message Passing Interface (MPI):** MPI is a standardized and portable programming model for distributed memory parallel computing. It provides a set of functions for inter-process communication, data movement, and synchronization.
- **OpenMP:** OpenMP is a shared memory parallel programming model that supports multi-threaded programming on shared memory systems. It provides a set of compiler directives and library functions for specifying parallelism and synchronizing parallel tasks.
- **Threading Building Blocks (TBB):** TBB is a C++ template library for writing efficient, concurrent, and scalable parallel applications. It provides a high-level abstractions for task parallelism, data parallelism, and concurrent containers.

- **CUDA:** CUDA is a parallel computing platform and programming model developed by NVIDIA for programming GPUs. It provides a C-based programming language and a set of libraries for developing high-performance parallel applications on GPUs.
- **OpenCL:** OpenCL is a cross-platform parallel computing framework for developing applications that run on CPUs, GPUs, and other heterogeneous systems. It provides a C-based programming language and a set of libraries for specifying parallelism and communication.

The choice of programming model depends on the specific requirements of the application, including the type of parallel computing system being used, the level of parallelism desired, and the complexity of the application.

Applications of Parallel Computing

Parallel computing has a wide range of applications in various fields, including:

Scientific and Engineering Applications: Parallel computing is widely used in scientific and engineering applications, such as weather forecasting, molecular dynamics, fluid dynamics, and structural mechanics.

Machine Learning and Artificial Intelligence: Parallel computing is used in many machine learning and AI applications, such as deep learning, image and video processing, and natural language processing.

Financial Modeling: Parallel computing is used in financial modeling, such as risk management, portfolio optimization, and pricing of financial instruments.

Gaming: Parallel computing is used in gaming to improve graphics and physics simulation, as well as to provide a more immersive gaming experience.

Healthcare: Parallel computing is used in healthcare, such as medical imaging, genomic sequencing, and drug discovery.

Big Data Analytics: Parallel computing is used in big data analytics, such as large-scale data processing and storage, data mining, and predictive analytics.

Weather and Climate Modeling: Parallel computing is used in weather and climate modeling, such as global climate simulations and regional weather forecasting.

Supply Chain Optimization: Parallel computing is used in supply chain optimization, such as logistics, inventory management, and transportation scheduling.

The use of parallel computing can significantly improve the performance of these applications, enabling faster and more accurate solutions to complex problems.

Advantages of Parallel Computing

Parallel computing provides several advantages over traditional serial computing, including:

- **Increased Processing Power:** Parallel computing allows for the simultaneous processing of multiple tasks, leading to increased processing power and faster problem-solving times.
- **Improved Scalability:** Parallel computing systems can be easily scaled by adding additional processing nodes, allowing for improved performance as the size of the problem increases.
- **Increased Throughput:** Parallel computing can increase the overall throughput of an application by processing multiple tasks simultaneously, reducing the total time to complete a given set of tasks.
- **Better Resource Utilization:** Parallel computing systems can make better use of available computing resources, such as CPU cycles and memory, leading to improved performance and reduced costs.
- **Improved Accuracy:** Parallel computing can provide improved accuracy in some applications, such as weather forecasting, scientific simulations, and financial modeling, by processing a larger amount of data in a shorter amount of time.
- **Greater Flexibility:** Parallel computing systems can be easily adapted to changing requirements, making them well suited for a wide range of applications.
- **Increased Reliability:** Parallel computing systems can provide increased reliability by allowing for the parallel processing of multiple tasks, reducing the impact of a single processing node failure.

Overall, parallel computing provides significant advantages over traditional serial computing, making it an essential tool for solving complex problems in a wide range of fields.

Challenges and Limitations of Parallel computing with Solutions

While parallel computing provides many benefits, it also comes with several challenges and limitations, including:

- **Programming Complexity:** Parallel programming can be complex and time-consuming, requiring a different mindset and skill set than traditional serial programming.
Solution: High-level parallel programming models, such as OpenMP and MPI, can simplify the process of developing parallel programs by providing abstractions for inter-process communication and synchronization.
- **Debugging and Testing:** Debugging and testing parallel programs can be difficult due to the complexity of parallel systems and the interactions between parallel tasks.
Solution: Debugging and testing tools, such as parallel debugger and performance analysis tools, can help simplify the process of identifying and fixing bugs in parallel programs.
- **Data Dependencies:** Parallel programs can be limited by data dependencies, which arise when the outcome of one task depends on the outcome of another task
Solution: Algorithmic techniques, such as data decomposition and data parallelism, can be used to reduce data dependencies and improve parallel performance.

- **Load Imbalance:** Load imbalance can arise when some processing nodes have more work to do than others, leading to poor performance and reduced scalability.
Solution: Load balancing algorithms and techniques can be used to distribute work evenly across processing nodes, improving performance and scalability.
- **Network Communication:** Network communication can be a bottleneck in parallel systems, affecting performance and scalability.
Solution: High-speed interconnects, such as Infiniband and 10 Gb Ethernet, can be used to improve network communication performance, while message-passing libraries, such as MPI, can be used to optimize communication between processing nodes.
- **Scalability:** As the number of processing nodes increases, the difficulty of parallel programming, debugging, and testing also increases.
Solution: Scalability techniques, such as domain decomposition and hierarchical parallelism, can be used to simplify parallel programming and improve scalability.

Overall, parallel computing presents several challenges and limitations, but with the use of appropriate tools and techniques, these can be effectively addressed, leading to improved performance and scalability

Conclusion

In conclusion, parallel computing is a powerful tool for solving complex problems in a wide range of fields, from scientific and engineering applications to financial modeling and big data analytics. By

leveraging the processing power of multiple processing nodes, parallel computing can significantly improve performance, increase accuracy, and provide greater scalability. As the demand for computational power continues to increase, parallel computing will play an increasingly important role in solving complex problems and driving innovation in a wide range of fields. With continued advances in hardware and software, the future of parallel computing looks bright, offering the potential for even greater performance, scalability, and efficiency.

REFERENCE

- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar. Introduction to Parallel Computing, 2nd edition, Addison-Wesley, 2003.
- William Gropp, Ewing Lusk, and Anthony Skjellum. Using MPI: Portable Parallel Programming with the Message-Passing Interface, 2nd edition, MIT Press, 1999.
- <https://parallel.org/>
- <https://www.openmp.org/>
- <https://www.computer.org/web/computingnow/parallel>