IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 5, Issue 2, January 2025

A Descriptive Study on Interconnection Networks for Parallel Computing and Algorithm Models in Parallel Computing

Er. Rajbhan Singh[1] and Dr. Govind Narayan Singh[2]

Computer Lecturer, S. R. Government Polytechnic College, Sagar, M.P, India[1]

Professor and Head (Physics), Government College Lal Ganow, Rewa, M.P, India[2]

Abstract*: In parallel computing, Interconnection networks are very crucial for efficient communication among all processors within a similar system.*

Parallel computing has become a crucial topic in the concern of computer science and also it is revealed to be critical when researching in high performance. The evolution of computer architectures towards an improved number of nodes, where parallelism could be the approach to option for speeding up an algorithm within the last few decades.

Efficient data transfer between processors is an essential component in any large scale parallel computation. Motivated by the growing interest in parallel computers, a significant amount of theoretical research has been devoted to the area of interconnection networks for parallel computers, most of it to the packet routing (or store-and-forward) model of communication. We survey some of the major developments in this field, and discuss several new alternative models of communication, In many large scale applications, communication time dominates the execution time of the whole parallel computation. Thus, the performance of a large scale parallel computer is highly correlated with the efficiency of its network and communication algorithm.

The combination of processing units build a model of computation (circuits) has gained an essential place in the area of high performance computing (HPC) due to its configuration and considerable processing supremacy that is parallel, series, etc. The aim of the Presenting this paper is study on the idea of parallel computing and its programming models and also explore some theoretical and technical concepts which can be often needed to understand the Interconnection network. In particular, we show how this technology is new in assisting the field of computational physics, especially when the issue is data parallel.

In the real-life example of parallel computing, there are two queues to get a ticket of anything; if two cashiers are giving tickets to 2 persons simultaneously, it helps to save time as well as reduce complexity.

Keywords: parallel computing

I. INTRODUCTION OF INTERCONNECTION NETWORK

An interconnection network is used for exchanging data between two processors in a multistage network. In case of multiprocessor systems, the performance will be severely affected in case the data exchange between processors is delayed.

The multiprocessor system has one global shared memory and each processor has a small local memory. The processors can access data from memory associated with another processor or from shared memory using an interconnection network. Thus, interconnection networks play a central role in determining the overall performance of the multiprocessor systems.

The interconnection networks are like customary network systems consisting of nodes and edges. The nodes are switches having few input and few output (say n input and m output) lines. Depending upon the switch connection, the data is forwarded from input lines to output lines. The interconnection network is placed between various devices in the multiprocessor network.

In the multiprocessor systems, these are multiple processor modules (each processor module consists of a processing element, small sized local memory and cache memory),shared global memory and shared peripheral devices.

II. INTERCONNECTION NETWORKS FOR PARALLEL COMPUTING

Interconnection Networks play a vital role in parallel computing by providing the communication infrastructure. Choosing the appropriate network topology is crucial to meet the specific requirements of parallel computing applications. Parallel Computing is defined as the process of distributing a larger task into a small number of independent tasks and then solving them using multiple processing elements simultaneously. Parallel computing is more efficient than the serial approach as it requires less computation time.

In the parallel computing, Interconnection networks are very crucial for efficient connection among all processors within a similar system. There are 2 types of Interconnection Network

(i) Static Interconnection Network

(ii) Dynamic Interconnection Network

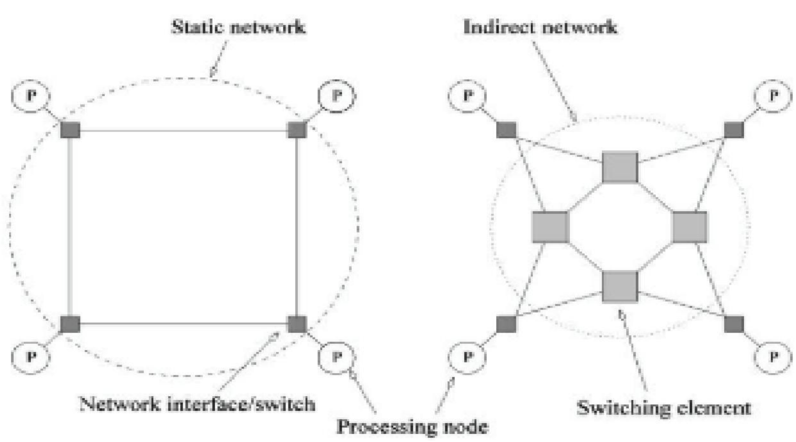
1. Static Interconnection Network

Static interconnection networks are fixed. In a unidirectional static interconnection network connections between nodes allow communication to occur in only one direction. So the data can be transmitted from one node to another node but not in the reverse direction. However, in a bidirectional static interconnection network, the connection between nodes allows communication to occur in both directions. The choice between both connections depends on the specific requirements of the parallel computing system.

2. Dynamic Interconnection Network

Unlike Static Interconnection Network, where connections are fixed between node, it enables the dynamic reconfiguration of connection to adapt to changing communication requirements Reconfiguration, Switching Mechanisms, Flexibility, Scalability, Communication Efficiency and Fault Tolerance are the some key points about Dynamic Interconnection Network,

Following diagram represent the classification of Static Interconnection Network and the Dynamic Interconnection Network



III. CONCEPT OF PARALLEL COMPUTING

Parallel computing is a type of computation where many calculations or the execution of processes are carried out simultaneously. Large problems can often be divided into smaller ones, which can then be solved at the same time.

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 5, Issue 2, January 2025

There are several different forms of parallel computing: bit-level, instruction-level, data, and task parallelism. Parallel computing is more efficient than the serial approach as it requires less computation time.

Parallel Computing is defined as the process of distributing a larger task into a small number of independent tasks and then solving them using multiple processing elements simultaneously. Parallel computing is more efficient than the serial approach as it requires less computation time.

Parallel computing refers to the process of executing several processors an application or computation simultaneously. Generally, it is a kind of computing architecture where the large problems break into independent, smaller, usually similar parts that can be processed in one go. It is done by multiple CPUs communicating via shared memory, which combines results upon completion. It helps in performing large computations as it divides the large problem between more than one processor.

Parallel computing also helps in faster application processing and task resolution by increasing the available computation power of systems. The parallel computing principles are used by most supercomputers employ to operate. The operational scenarios that need massive processing power or computation, generally, parallel processing is commonly used there.

Typically, this infrastructure is housed where various processors are installed in a server rack; the application server distributes the computational requests into small chunks then the requests are processed simultaneously on each server. The earliest computer software is written for serial computation as they are able to execute a single instruction at one time, but parallel computing is different where it executes several processors an application or computation in one time.

There are many reasons to use parallel computing, such as save time and money, provide concurrency, solve larger problems, etc. Furthermore, parallel computing reduces complexity. In the real-life example of parallel computing, there are two queues to get a ticket of anything; if two cashiers are giving tickets to 2 persons simultaneously, it helps to save time as well as reduce complexity.

IV. TYPES OF PARALLEL COMPUTING

From the open-source and proprietary parallel computing vendors, there are generally three types of parallel computing available

(i) Bit-level parallelism

(ii) Instruction-level parallelism

(iii) Task Parallelism

1. 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. For example, there is an 8-bit processor, and you want to do an operation on 16-bit numbers. First, it must operate the 8 lower-order bits and then the 8 higher-order bits. Therefore, two instructions are needed to execute the operation.

The operation can be performed with one instruction by a 16-bit processor.

2. 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. The software approach in instruction-level parallelism functions on static parallelism, where the computer decides which instructions to execute simultaneously.

3. 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.

V. APPLICATIONS OF PARALLEL COMPUTING

There are various applications of Parallel Computing, which are as follows:

1. One of the primary applications of parallel computing is Databases and Data mining.
2. The real-time simulation of systems is another use of parallel computing.
3. The technologies, such as Networked videos and Multimedia.
4. Collaborative work environments.
5. The concept of parallel computing is used by augmented reality, advanced graphics, and virtual reality.

Advantages of Parallel computing

Some of the Parallel computing advantages are given as below:

1. In parallel computing, more resources are used to complete the task that led to decrease the time and cut possible costs. Also, cheap components are used to construct parallel clusters.
2. Comparing with Serial Computing, parallel computing can solve larger problems in a short time.
3. For simulating, modelling, and understanding complex, real-world phenomena, parallel computing is much appropriate while comparing with serial computing.
4. When the local resources are finite, it can offer benefit you over non-local resources.
5. There are multiple problems that are very large and may impractical or impossible to solve them on a single computer; the concept of parallel computing helps to remove these kinds of issues.

Disadvantages of Parallel Computing

There are many limitations of parallel computing, which are as follows:

1. It addresses Parallel architecture that can be difficult to achieve.
2. In the case of clusters, better cooling technologies are needed in parallel computing.
3. It requires the managed algorithms, which could be handled in the parallel mechanism.
4. The multi-core architectures consume high power consumption.
5. The parallel computing system needs low coupling and high cohesion, which is difficult to create.
6. The code for a parallelism-based program can be done by the most technically skilled and expert programmers.

VI. PARALLEL ALGORITHM MODELS IN PARALLEL COMPUTING

Parallel Computing is defined as the process of distributing a larger task into a small number of independent tasks and then solving them using multiple processing elements coincidently. Parallel computing is more efficient than the serial approach as it requires less computation time.

The need for a parallel algorithm model arises in order to understand the strategy that is used for the partitioning of data and the ways in which these data are being processed. Therefore every model being used provides proper structuring based on two techniques. They are as follows:

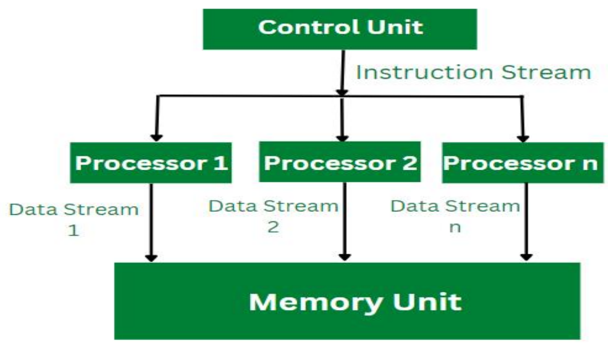
1. Selection of proper partitioning and mapping techniques.
2. Proper use of strategy in order to reduce the interaction**.**

Types of Parallel Models

1. The Data-Parallel Model

The data-parallel model algorithm is one of the simplest models of all other parallel algorithm models. In this model, the tasks that need to be carried out are identified first and then mapped to the processes. This mapping of tasks onto the processes is being done statically or semi-statically. In this model, the task that is being performed by every process is the same or identical but the data on which these operations or tasks are performed is different.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 5, Issue 2, January 2025

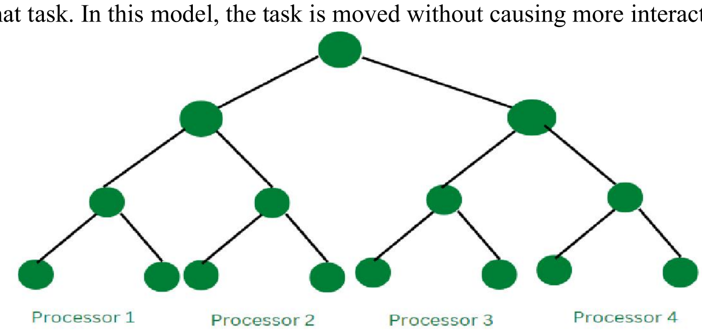


In the above example of dense matrix multiplication, the instruction stream is being divided into the available number of processors. Each processor computes the data stream it is allocated with and accesses the memory unit for read and write operation. As shown in the above figure, the data stream 1 is allocated to processor 1, once it computes the calculation the result is being stored in the memory unit.

2. Work Pool Model

The work pool model is also known as the task pool model. This model makes use of a dynamic mapping approach for task assignment in order to handle load balancing. The size of some processes or tasks is small and requires less time. Whereas some tasks are of large size and therefore require more time for processing. In order to avoid the inefficiency load balancing is required.

The pool of tasks is created. These tasks are allocated to the processes that are idle in the runtime. This work pool model can be used in the message-passing approach where the data that is associated with the tasks is smaller than the computation required for that task. In this model, the task is moved without causing more interaction overhead.

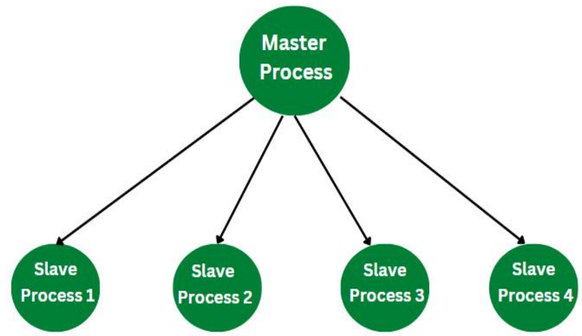


In the above example of the parallel search tree, that uses the work pool model for its computation uses four processors simultaneously. The four sub-tress are allocated to four processors and they carry out the search operation.

3. Master-Slave Model

Master Slave Model is also known as Manager- worker model. The work is being divided among the process. In this model, there are two different types of processes namely master process and slave process. One or more process acts as a master and the remaining all other process acts as a slave. Master allocates the tasks to the slave processes according to the requirements. The allocation of tasks depends on the size of that task. If the size of the task can be calculated on a prior basis the master allocates it to the required processes.

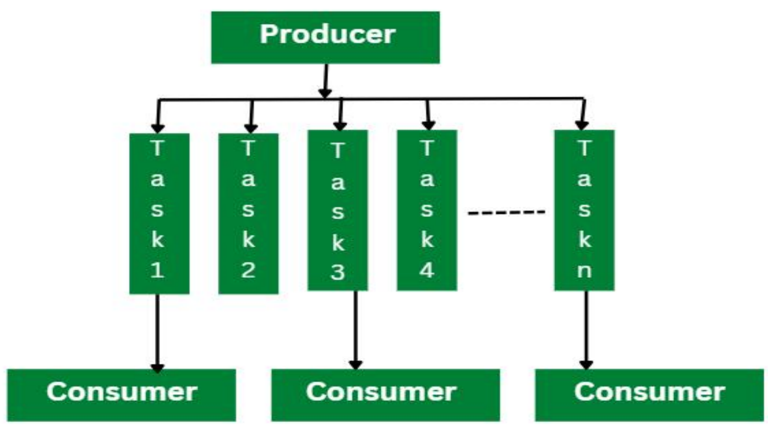
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 5, Issue 2, January 2025



As shown in the above example of the Master-Slave model, the distribution of workload is being done across multiple processes. As shown in the above diagram, one node is the master process that allocates the workload to the other four slave processes. In this way, each sub-computation is carried out by multiple slave processes.

4. The Pipeline Model

The Pipeline Model is also known as the Producer-Consumer model. This model is based on the passing of a data stream through the processes that are arranged in succession. Here a single task goes through all the other processes. They are then accessed by the required processes in a sequential manner. Once the processing of one process is finished it goes to the next present process. In this model, the pipeline acts as a chain of producers and consumers.

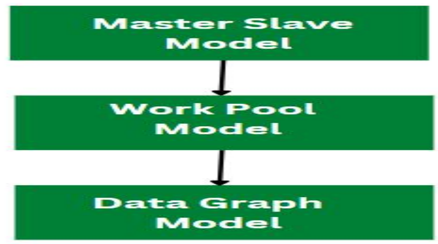


As shown in the above diagram, the Parallel LU factorization algorithm uses the pipeline model. In this model, the producer reads the input matrix and generated the tasks that are required for computing the LU factorization as an output. The producer divides this input matrix into a smaller size of multiple tasks and shares them into a shared task queue. The consumers then retrieve these blocks and perform the LU factorization on each independent block.

5. Hybrid Model

A hybrid model is the combination of more than one parallel model. This combination can be applied sequentially or hierarchically to the different phases of the parallel algorithm. The model that can be efficient for performing the task is selected as a model for that particular phase.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 5, Issue 2, January 2025



As shown in the above hybrid model where three different models are used at each phase master-slave model, the work pool model, and the data graph model. Consider the above example where the master-slave model is used for the data transformation task. The master process distributes the task to multiple slave processes for parallel computation. In the second phase work pool model is used for data analysis and similarly data graph model is used for making the data visualization. In this way, the operation is carried out in multiple phases and by using different parallel algorithm models at each phase.

VII. CONCLUSION

The computational graph has undergone a great transition from serial computing to parallel computing. Tech giant such as Intel has already taken a step towards parallel computing by employing multicore processors. Parallel computation will revolutionize the way computers work in the future, for the better good.

The parallel algorithm model solves the large problem by dividing it into smaller parts and then solving each independent sub-task simultaneously by using its own approach. Each parallel algorithm model uses its own data partitioning and data processing strategy. However, the use of these parallel algorithm models improves the speed and efficiency of solving a particular task.

REFERENCES

[1]. Rakesh Kumar Katare, Sunil Tiwari, Anupriya Singh, A Study of Dependencies among processors in Interconnection Network

[2]. Chamberlain, R.D.; Franklin, M.A.; Ch'ng, S.B. Gemini: An optical interconnect network for parallel processing. IEEE TPDS2002,13, 1038–1055.

[3]. K.E. Batcher, "The Flip Network in STARAN," Proc.1976 on Parallel Processing, IEEE, New York,

[4]. H.J. Siegel, "Interconnection Networks for SIMD Machines," Computer, Vol. 12, No. 6, June 1979, pp. 57-65

[5]. An on line article on Interconnection Networks for Parallel Processors