Design of Analytical Model for Memory Architecture of Parallel Processing

Mallikarjun Reddy Ireddy[1] ,Indrani Vasireddy [2]

[1,2]Computer Science and Engineering, Sri Satya Sai
University of Technology and Medical Sciences, Sehore, India.

ABSTRACT

Different architectures, different programming models are suitable for different applications and so the characteristics of the applications should make the decision for the selection of parallel hardware architecture and also the parallelization of the applications. A stream is just a sequence of items (instruction or data) SISD: A type of computer architecture in which there is a single instruction cycle, and operands are fetched in serial fashion into a single processing unit before execution. In computing, MIMD (multiple instruction, multiple data) is a technique employed to achieve parallelism. Machines using MIMD have a number of processors that function asynchronously and independently. At any time, different processors may be executing different instructions on different pieces of data. The two categories of MIMD architecture are Shared Memory Systems (tightly coupled) and Distributed Memory Systems (loosely coupled). To exploit these computing resources efficiently, hybrid systems are also in practice, which uses both shared as well as distributed memory for running a parallel application. In tightly coupled systems, a single system-wide primary memory (address space) is shared by all the processors whereas in loosely coupled systems, the processors do not share memory and each processor has its own local memory. Usually, tightly coupled systems are referred to as Parallel Processing Systems and loosely coupled systems are referred to as Distributed Computing Systems

I.INTRODUCTION

Moving forward, if every computer will be a parallel computer, most programs must execute in parallel and most programming teams must be able to develop parallel programs, a daunting goal given the above problems. Illinois has a rich history in parallel computing starting from the genesis of the field and continues a broad research program in parallel computing today. This program includes the Universal Parallel Computing Research Center (UPCRC), established at Illinois by Intel and Microsoft, together with a sibling center established at Berkeley. These two centers are focused on the problems of multicore computing, especially in the client and mobile domains.

Parallel computing is a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently ("in parallel"). There are several different forms of parallel computing: bit-level, instruction level, data, and task parallelism. Parallelism has been employed for many years, mainly in high-performance computing, but interest in it has grown lately due to the physical constraints preventing frequency scaling. As power consumption (and consequently heat generation) by computers has become a concern in recent years, parallel computing has become the dominant paradigm in computer architecture, mainly in the form of multi-core processors and memory allocation.

Parallel computing is the simultaneous execution of the same task on multiple processors in order to obtain faster results. It is widely accepted that parallel computing is a branch of distributed computing, and puts the emphasis on generating large computing power by employing multiple processing entities simultaneously for a single computation task. These multiple processing entities can be a multiprocessor system, which consists of multiple processors in a single machine connected by bus or switch networks, or a multicomputer system, which consists of several independent computers interconnected by telecommunication networks or computer networks. Besides in parallel computing, distributed computing has also gained significant development in enterprise computing. The main difference between enterprise distributed computing and parallel distributed computing is that the former mainly targets on integration of distributed resources to collaboratively finish some task, while the later targets on utilizing multiple processors simultaneously to finish a task as fast as possible. In this study, because we focus on high performance computing using parallel distributed computing with memory allocation, we will not cover enterprise distributed computing, and we will use the term “Parallel computing”.

Addressing these challenges will require research innovations that depart from the evolutionary path of conventional architectures, memory allocation and programming systems. This has made multiprocessor designs, being hard to program and providing at most linear speedups, virtually impractical since single processor systems have been easy to program and they have catched the performance of their multiprocessor counterparts quite soon. Recently, this development has slowed down radically due to heating problems that have almost killed the growth of clock frequencies and increasing resistance of shrinking wires that effectively prevents raising the number of functional units and thus the number of executed instruction per clock cycle for general purpose applications.

II. STATEMENT OF THE PROBLEM

The main purpose is to study the memory architecture of parallel processing computer and examine the various problems associated into it.

III. DELIMITATIONS

1. The study is delimited to the parallel computing.
2. The study is also delimited to use in memory allocation in parallel computing.

IV. LIMITATION

The facts discussed in this study will be based entirely on the responses to the questionnaire and modeling of memory allocation in parallel computing therefore, ascertaining the genuineness of the responses will identify as the limitation of the study.

V. HYPOTHESIS

- There is no significant difference between present and past parallel computing.
- There is no significant difference between memory allocation in parallel computing.

VI.DEFINATIONS AND EXPLANATION TERMS

6.1 Parallel Computing

Parallel computing is a type of computation in which 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.

6.2 Memory

Computer **memory** is any physical device capable of storing information temporarily or permanently.

6.3 Memory Allocation

Memory allocation is the process of setting aside sections of memory in a program to be used to store variables, and instances of structures and classes.

VII.SIGNIFICANCE OF THE STUDY

The study will be the family of scalable Emulated shared memory (ESM) CMP / MP-SOC architectures to address both programmability and performance issues of CMP / MP-SOC design for a wide range of general purpose applications. The performance of ESM CMPs / MP-SOCs will be turned out to be insensitive to the degree of intercommunication. An ESM CMP / MP-SOC consists of chained multithreaded processors with dedicated instruction memory modules, highly interleaved data memory modules and a high-capacity sparse mesh (SM) interconnection network . It consists the followings:-

1. The study would deal in parallel computing.
2. The study would coordinate the memory allocation in parallel computing.

VIII. OBJECTIVES

- To study the parallel computing.
- To examine the concept of memory.
- To define the model of memory allocation in parallel computing.
- To discuss the various algorithms related to it.

- To measure the various issues and challenges involved into it.

IX.REVIEW OF RELATED LITERATURE

Saxena (2015) The recent switch to parallel microprocessors is a milestone in the history of computing. Industry has laid out a roadmap for multicore designs that preserves the programming paradigm of the past via binary compatibility and cache coherence. Conventional wisdom is now to double the number of cores on a chip with each silicon generation.

Rajput (2014) Face is significant method to identifying person. Face Recognition is process which is base on graphic & image processing. GPU is Graphic processing unit. GPU has become an integral part of today's mainstream computing systems, becoming more widely used in demanding consumer applications and high-performance computing. This article describes the rapid evolution of GPU architectures with Flynn's Taxonomy how it can exploit for Face Recognize. This article also describe the graphics processors to massively parallel many-core multi threading multiprocessor multiprocessors, recent developments in GPU computing architectures, and how the enthusiastic adoption of CPU + GPU co-processing is accelerating parallel applications.

Navvaro (2014) Parallel computing has become an important subject in the field of computer science and has proven to be critical when researching high performance solutions. The evolution of computer architectures (multi-core and many-core) towards a higher number of cores can only confirm that parallelism is the method of choice for speeding up an algorithm.

Trelles (2013) This document surveys the computational strategies followed to parallelize the most used software in the bioinformatics arena. The studied algorithms are computationally expensive and their computational patterns range from regular, such as database searching applications, to very irregularly structured patterns (phylogenetic trees). Fine- and coarse-grained parallel strategies are discussed for these very diverse sets of applications. This overview outlines computational issues related to parallelism, physical machine models, parallel programming approaches, and scheduling strategies for a broad range of computer architectures. In particular, it deals with shared, distributed, and shared/distributed memory architectures.

Tarunet al (2013) Despite the tremendous progress in this area, video compression remains a challenging research problem due to its computational requirements and also because of the need for higher picture quality at lower data rates. Designing efficient coding algorithms continues to be a prolific area of research. For circumvent the computational requirement, researchers has resorted to parallel processing with a variety of approaches using dedicated parallel VLSI architectures as well as software on general-purpose available multiprocessor systems. Despite the availability of fast single processors, parallel processing helps to explore advanced algorithms and to build more sophisticated systems.

X. PROCEDURE

This Chapter deals with the procedure adopted for algorithm and selection of subjects, non measures, collection of data and the statistical techniques led for analysis of data.

XI. SELECTION OF SUBJECTS

The study will discuss the operating system & evaluate the stepping stones to a model & various algorithm and examine the various issues and challenges involved it. The key features are

• easy-to-program lock-step-synchronous arbitrary ordered multiprefix CRCW PRAM-style programming model providing uniform shared memory with the single step access latency and fine-grained machine instruction level synchronous parallel execution,

• software-based design methodology ultimately supporting flexibility and general purpose operation in "Parallel application development",

• homogeneous structure for simplifying design and making it easier to integrate it as a part of larger system,

• zero overhead multithreading for hiding the memory latency, balancing computation and memory access, and providing support for fine-grained TLP,

• super pipelining for clock cycle minimization,

• hazard-free inter thread pipelining with multiple functional units organized as a chain for hiding the thread switch costs and maximizing ILP within a super step of parallel execution,

• totally cacheless design for avoiding cache coherency problems and simplifying the memory system,

• a physically feasible acyclic two-dimensional sparse mesh network exploiting locality for high-bandwidth and deadlock free inter-resource communication,

• randomized hashing of memory words across the modules to avoid hotspots and congestion in the network,

• wave based synchronization separating references belonging to successive supersteps and allowing for different parts of the machine to execute different phases of the program depending on the progress of communication,

• an advanced explicit synchronization mechanism supporting arbitrary multiple simultaneous barrier synchronization, and

• highly interleaved memory modules for eliminating the speed difference between processors and memory banks.

XII.CRITERION MEASURE

Studies shall be descriptive in nature. Different data shall be prepared to ascertain the trend and impact. Data would be collected through different primary and secondary sources like:

PRIMARY SOURCES:-

1. Questionnaire
2. Interviews

SECONDARY SOURCES:-

1. Publications
2. Internet
3. Journals

REFERENCES

[1.] Rajput Anil, Bhushan and Sarita, “Parallel Processing Unit with MIMD Architecture”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol 4, Issue 4, pp 1055-1059.

[2.] M. Gebhart et al., “Energy-Efficient Mechanisms for Managing Thread Context in Throughput Processors,” Proc. ACM/IEEE Int’l Symp. Computer Architecture, ACM Press, 2011, pp. 235-246.

[3.] Guo Liang Chen, Yun Quan Zhang, “Survey on Parallel Computing,” Journal of Computer Science & Technology, Sept. 2012, Vol. 21, No. 5, pp. 665-673.

[4.] Peter N. Glaskowsky, “NVIDIA’s Fermi: The First Complete GPU Computing Architecture”, Prepared under contract with NVIDIA, 2009.

[5.] J. Nickolls and W.J. Dally, ‘‘The GPU Computing Era,’’ IEEE Micro, vol. 30, no. 2, 2010, pp. 56-69.

[6.] Rajkumar Sharma and Priyesh Kanungo, “Performance Evaluation of MPI and Hybrid MPI+OpenMP Programming Paradigms on Multi-Core Processors Cluster,” IEEE International Conference on Recent Trends in Information Systems, Jadavpur University, Kolkata, December 2011, pp. 137-140.

[7.] S. Galal and M. Horowitz, ‘‘Energy-Efficient Floating Point Unit Design,’’ IEEE Trans. Computers, vol. 60, no. 7, 2011, pp. 913-922.

[8.] T. Vogelsang, ‘‘Understanding the Energy Consumption of Dynamic Random Access Memories,’’ Proc. Int’l Symp, Micro architecture, IEEE CS Press, 2010, pp. 363-374.

[9.] Y. Komura and Y. Okabe, GPU-based single-cluster algorithm for the simulation of the missing model, J. Comput. Phys., 231(4):1209–1215, February 2012.

[10.] Sadrozinski, Hartmut F.-W.; Wu, Jinyuan, Applications of Field-Programmable Gate Arrays in Scientific Research, Taylor & Francis, 2015.