International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com‖ Volume 2 Issue 3‖ March 2016 ‖ PP 24-29

Multi-Core Architectures for parallel programs with a Survey of Performance Evaluation Methods

L.AMUDHA[1], T.M. NITHYA[2], J. RAMYA[3]
[1]*Asst. Prof., Department of CSE,K. Ramakrishnan College Of Engineering, Samayapuram, Tamilnadu.*
[2]*Asst. Prof., Department of CSE,K. Ramakrishnan College Of Engineering, Samayapuram, Tamilnadu.*
[3]*Asst. Prof., Department of CSE,K. Ramakrishnan College Of Engineering, Samayapuram, Tamilnadu.*

ABSTRACT: The Number of cores on the processor chip increases at each release of a new generation of CPU. The processor performance increases rapidly every decade. At the same time, Multi core CPUs reduces power consumption when compared to multiprocessors. As the number of cores on multi-cores increase, all cores must be effectively utilized. So there should be a different approach in programming. Today movement from traditional programming to parallel programming becomes inevitable. And most of the parallel programs need to be hard coded manually by a person. A special compiler and scheduler is needed to handle the parallel programs that would run on a multi-core processor. The performance metrics, factors that affect the performance, and challenges faced in multi-core and parallel programming are discussed briefly in this paper.
Keywords: Multi-core, parallel programming languages, performance, speed-up.

I. Introduction

When compared to multiprocessors where multiple CPUs are on different chips on the same mother board, Fig. 1, multi-core CPU can be of very high performance if all the cores are utilized effectively[1]. For any processor that handles hungry applications like music players, if a single core like Fig. 1 is used, a significant slowdown of the entire system could be observed. Alternatively, a multi-core allots different tasks for different processors. Core 1 may be allotted with word document processing, Core 2 with antivirus and core 3 with music and so on. The full potential of all the cores are not utilized effectively. A 2 core architecture in Fig. 2 shows the allocation of separate hardware and cache for each core.

Very large applications that are having parallel processing ability can be made to effectively use all the cores effectively and efficiently. Sequential tasks like large programs still need to use a single core because of Data dependency and Structural hazards. To overcome this, sequential programs need to be converted to parallel programs that can run on different cores without affecting the meaning or outcome of the application, at the same time improving the performance.

This research paper considers the issues while transforming sequential program into parallel program. Issues are dealt and performance metrics are obtained.

II. MULTI-CORES

Moore‘s Law suggests that the number of cores per CPU chip will double every 18 to 24 months. With reduction in size, reduced power consumption, it is a great challenge to design parallel applications to utilize all the cores on the chip efficiently [8]. Divide and conquer parallel computing scheme on multi-cores greatly reduces the processing time and also performance of all cores increases drastically [9]. The interaction of various components of a computer system with many cores is shown in Fig.3.

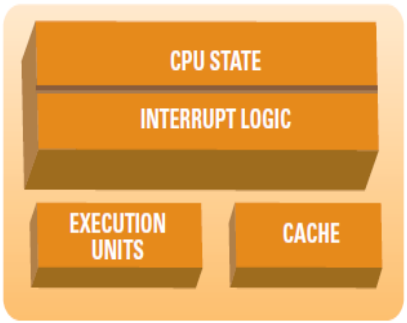

Fig. 1 Single core

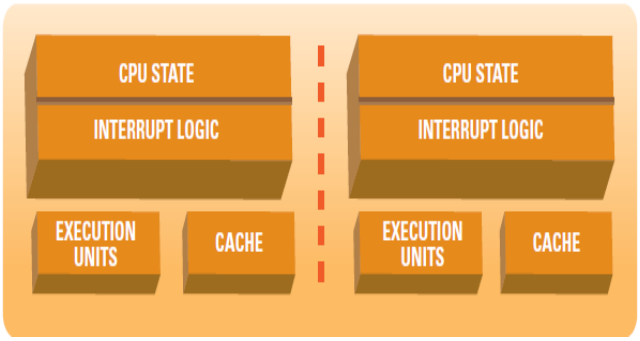

Fig. 2. Multiple Cores [2 cores]

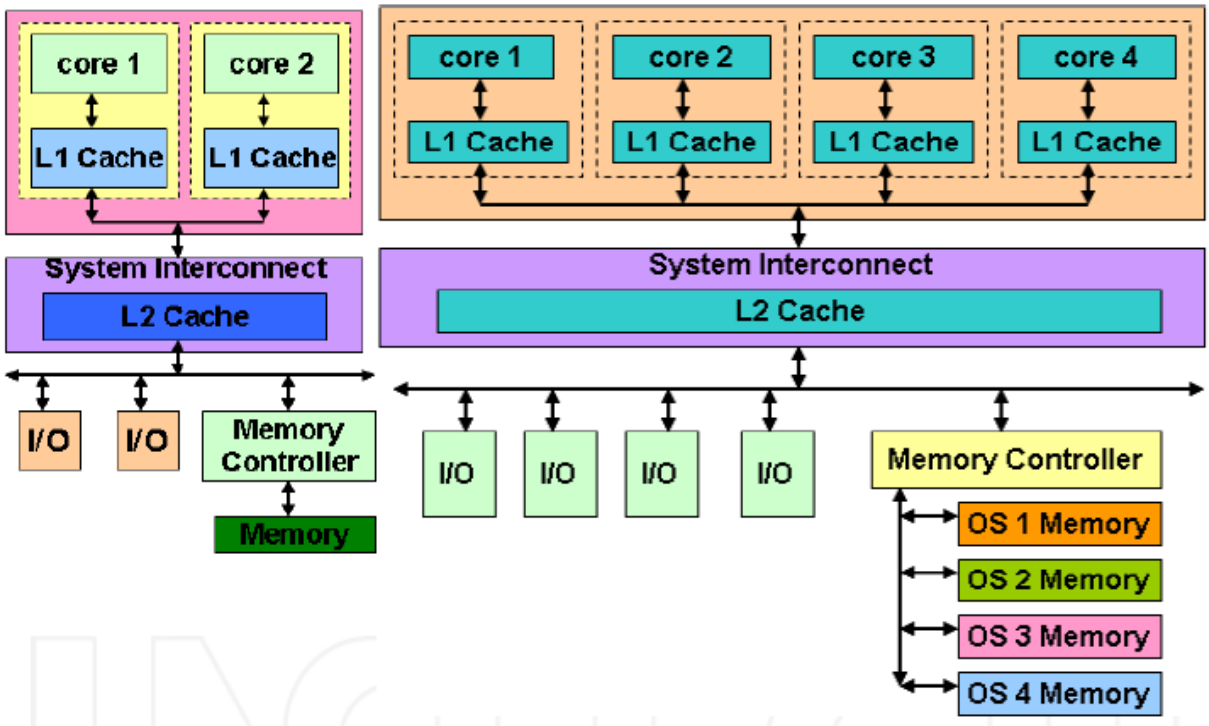

Fig. 3. A Typical Multi-core processor

III. PERFORMANCE EVALUATION METHODS

A comparison of execution sequence in single core[4] and multi-core shows the issues in improvement of execution process. The Fig.[5] depicts the utilization of Multi-core and fast execution compared with single core computation.

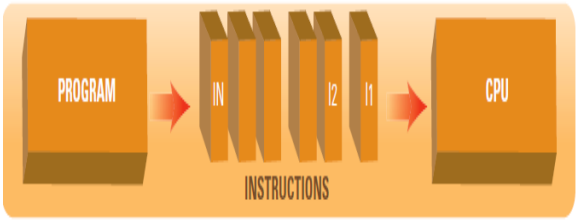

Fig.4. Instruction execution in single core processor

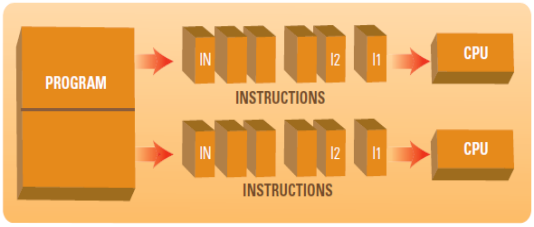

Fig 5. Instruction execution with multi-core processors.

There are different ways to evaluate multi-core CPU's performance. Different metrics and factors are to be considered which radically has an influence on Multi-core CPU performance. The main techniques in performance analysis are: Analytical Modeling, Simulation and Measurement.
Some of the evaluating criteria are Stage, Time, Tools, Accuracy, Trade-off Evaluation, Cost and Scalability.

IV. EVALUATION METRICS FOR APPLICATIONS

While most of the metrics are related to Time, Rate and Resources, for any parallel processing application, its performance is evaluated using some or all of the below mentioned factors:

1. Throughput: Average rate of successful processes
2. Response Time: The time that the user finishes placing the request and starts getting the response.
3. Execution Time: The time needed to complete execution of the program.
4. Energy: The power needed to run a program
5. Memory Latency: The time delay between memory controllers signaled the memory module to access data from RAM and the time data became available to memory module.
6. Memory Bandwidth: The rate of data sustained from CPU core to RAM.
7. Memory Contention Percent: The percentage among the cores trying to access the RAM at the same time.

V. FACTORS THAT AFFECT MULTICORE PROCESSOR PERFORMANCE

Performance of processors is measured using few or all of these factors such as Memory, Scalability, I/O bandwidth, Inter-core communication, Operating system, CPU clock speed, No of cores, Cache coherence [2][7].

Memory
Memory architecture used and speed of the memory can affect the multi-core performance. With increase in memory size and speed the processor performance increases.
Scalability
Scalability is the number of tasks available at a time. If a problem can be split into n number of tasks, then speed up is total time/n, which greatly improves the multi-core performance.
I/O bandwidth
Can affect the performance by utilizing the CPU cores which leads to more resources consumption without performance increasing.
Inter-core communication
The interaction between cores in multi-core CPU's can be implemented by various mechanisms, affecting overall CPU performance due to shared workloads between cores
Operating system
OS is the manager for the CPU and it assigned tasks to cores based on a scheduling mechanism, affecting the multicore CPU performance
CPU clock speed
Clock speed has an impact on processor performance with slow clock speed reducing throughput
No of cores
Affect the CPU performance as multicore architecture workload is divided between the cores
Cache coherence
Multicore architectures uses different caching mechanisms as the cache is shared among the cores, causing cache coherent to affect CPU performance.

VI. PARALLEL PROGRAMMING

Today parallel computing resources are available for inexpensive desktop systems. We can expect many software products that are parallel applications than that of sequential scientific computing[3]. Three main things to be considered while writing parallel programs are discussed in this section.

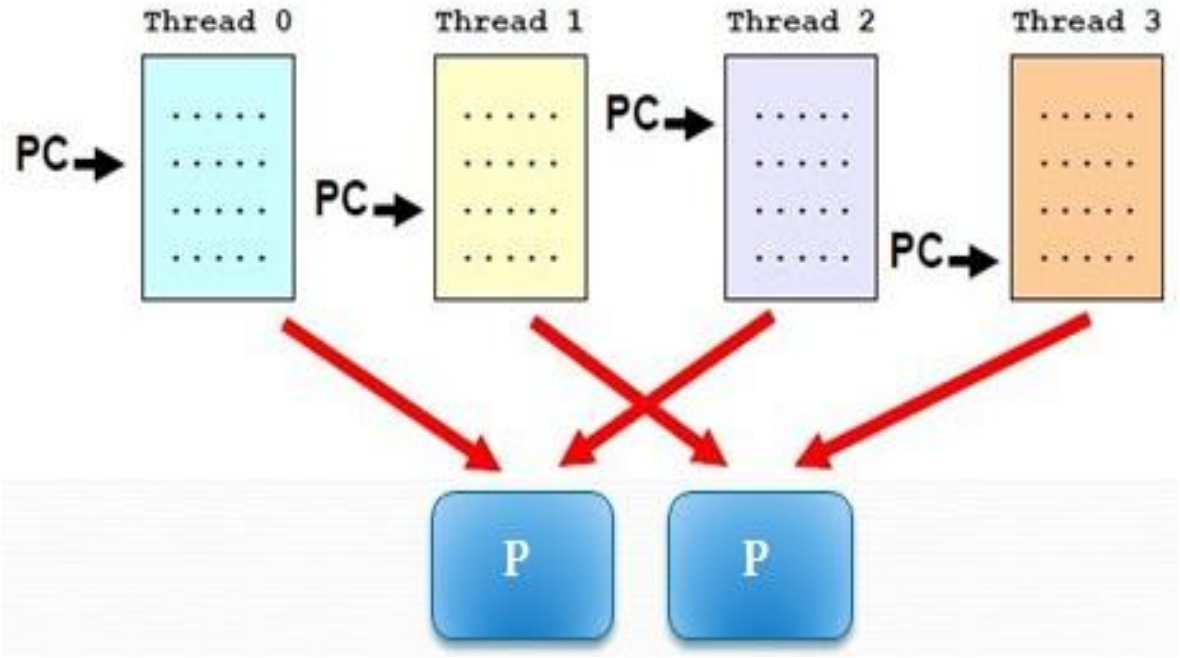

Fig. 6 Each core executing different threads

6.1 ARCHITECTURE AND PARALLEL SYSTEMS.

The parallel system identifies the area in programs that are independent and schedules each core of the system. Fig. 6 shows the allocation of 4 threads to a dual-core processor, where the program counter for each thread may differ. The output of execution is similar as in sequential execution in a single core processor.

6.2 PARALLEL PROGRAMMING MODELS AND ENVIRONMENTS.

Parallel programming models are divided into two main disciplines: process interaction and problem decomposition. Process interaction defines how parallel processes are capable to communicate with each other[12]. Problem decomposition defines the way in which the constituent processes are identified. They define the simultaneous execution of tasks[4].

6.3 USAGE OF EFFICIENT ALGORITHMS

Traditional serial algorithm, is an algorithm which can be executed a piece at a time on any processing device. In case of parallel programming as in Fig. 6, many parallel algorithms are executed concurrently. A subtype of parallel algorithms, distributed algorithms are algorithms designed to work in cluster computing and distributed computing environments[6].

Multi-Core Architectures for parallel programs with a Survey of Performance Evaluation Methods

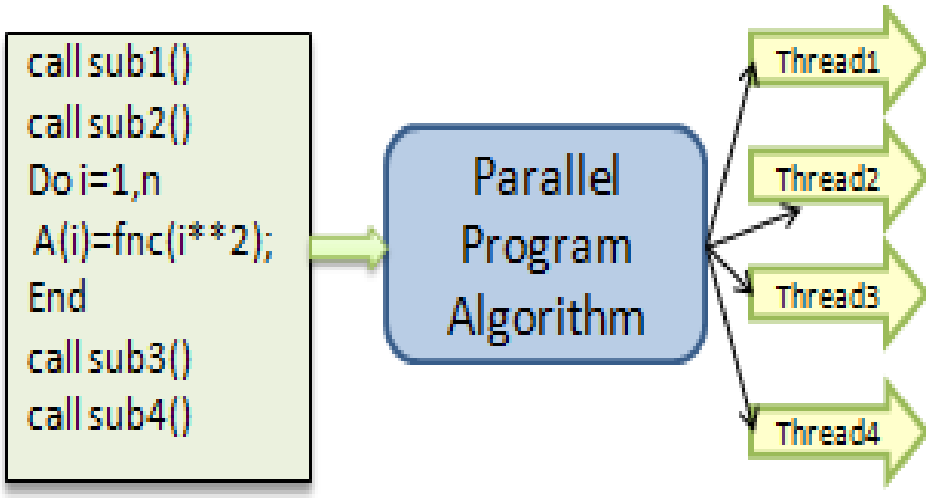

Fig. 6 Single program split into 4 threads

A major issue to be worried about is, if the communication overhead of additional processors outweighs the benefit of adding another processor, there will be a slowdown in performance.

VII. PARALLEL PROGRAM LANGUAGES

The main goal is to develop correct and efficient parallel program. Parallel programming models such as MPI, Fast Flow, Cilk Plus, Erlang, OpenHMPP, Scandium, and OpenMP are available for programming on multi-core platforms.

VIII. APPLICATIONS SUITED FOR PARALLEL PROCESSING

A wide variety of applications like Encryption, Big Data Analytics, Multi-Player Online Games, Graph coloring, Telephone or Cellular, Routers with Algorithm, Bio-Informatics can use parallel processing ability,

IX. RESEARCH SCOPE

Large scale multi-core processors bring a lot of challenges to software domain. And if we want faster software, we have to rewrite it. Common algorithms that we use today may become obsolete in future. To improve the performance predictable code and predictable linear data structure is needed. Every software needs to become highly multi-threaded. Algorithms also must be redefined as parallel algorithms. Therefore, researchers can turn their focus now from sequential programs to parallel programs and think of parallel programming language with parallel data structures and algorithms suited for multi-core systems. Another most common issue to be considered is about the thermal issues as the number of cores increase.

X. CONCLUSION

In order to take advantage of massively parallel hardware, we need to redefine every software. The performance evaluation metrics, issues and application area were discussed. For a N-Core compiler with multi-cores, when a parallel application is designed, a Debugger is needed to promptly report for any bugs and errors during the scheduling and processing. The speed of processing is always found to be dependent on the number of processing units available, the size of the program and the no of independent parts a problem can be broken into.

REFERENCES

Websites:

[1]. https://en.wikipedia.org/wiki/Parallel_computing
[2]. http://en.wikipedia.org/wiki/Intel 4004

Books:

[1]. S. Akhter, and J. Roberts, *Multi–Core Programming.* USA: Inter Press, 2006.

Journal Papers:

[1]. Borkar, S., Chien, A.A.*,"The Future of Microprocessors," Communications of the ACM, May 2011,Vol. 54*, No.5, pp.67-77.

[2]. *Performance Issues on Multi-Core Processors*, Ransford Hyman Jr.

[3]. Chaparro, P.,Gonzalez, J., Magklis, G., Cai, Q., Gonzalez, *"Understanding the Thermal Implications of Multicore Archtectures". IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS Vol.18, Iss.* 8, Aug. 2007, pp. 1055 - 1065.

[4]. Amdahl, G.M., *"Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities",* Proc. AFIPS Conf., AFIPS Press, 1967, pp. 483-485

[5]. Held, J., Bautista, J., and Koehl,S., *" From a Few Cores to Many: A Tera-scale Computing Research Overview*", *White Paper: Research at Intel.Intel Leap ahead*, 2006.

[6]. *Multicore Processors: Challenges, Opportunities, Emerging Trends*, Christian Märtin Faculty of Computer Science Augsburg University of Applied Sciences Augsburg, Germany embedded world Conference 2014

[7]. *Factors Impacting Performance of Multithreaded Sparse Triangular Solve*, Michael M. Wolf and Michael A. Heroux and Erik G. Boman Scalable Algorithms Dept., Sandia National Laboratories, Albuquerque, NM, USA.

[8]. *International Journal of Distributed and Parallel Systems (IJDPS) Vol.4, No.4*, July 2013, STUDY OF VARIOUS FACTORS AFFECTING PERFORMANCE OF MULTI-CORE PROCESSORS Nitin Chaturvedi,S Gurunarayanan, DOI : 10.5121/ijdps.2013.4404