International Journal of Computer Applications (0975 – 8887)
Volume 148 – No.14, August 2016

Perspective Study and Analysis of Parallel Architecture

Varsha Thakur
Research scholar
Pt.Ravishankar shukla
university,Raipur(C.G),India

Sanjay Kumar
Associate Professor
Pt.Ravishankar shukla
university,Raipur(C.G),India

ABSTRACT

As technology has advanced, Parallel Computing and Architecture has emerged as a research area with the potential of providing satisfactory and faster result for real time applications. Parallel architecture is those that emphasize on parallel and concurrent computation among different processors. This paper presents a thorough survey of the parallel architecture and performane is analysed on the basis of the execution time of few parallel sorting algorithms in multicore processors .To implement these algorithms we have used C programming language with OpenMp Libraries under Linux environment.

General Terms

Parallel and distributed computing.

Keywords

Parallel Architecture;Openmp;Sorting;Bitonic Sorting

1. INTRODUCTION

Parallel processing means dividing a problem into sub problem and executing these sub problems simultaneously. Parallel architecture emphasize on parallel processing between operation in some way. Parallel computing makes use of concurrently running the processes that are belonging to larger computation, for this reason the divide-and-conquer approach is usually preferred over other techniques [1]. With the rapid development of last few decades the area of Parallel architectures have emerged as challenging task for discovering new architectures that can perform better. Basic objective of parallel architecture is improve the speed. The factors generally considered in developing a new architecture include the internal circuitry of processors or PEs(Processing Elements), number of PEs, arrangement of PEs and memory modules in an architecture, the communication mechanism among the PEs and between the PEs and memory modules, number of instruction and data streams, nature of memory connections with the PEs, nature and types of interconnection among the PEs, and program overlapping. The internal circuitry of PEs plays a vital role in designing parallel architecture. Some parallel architecture are designed with small number of PEs of complex internal circuitry to enhance the overall performance of the architecture. Other architectures, on the other hand, are designed with a substantial number of PEs of simple internal circuitry to achieve the desired performance. [2]

2. CLASSIFICATION OF PARALLEL ARCHITECTURE

Parallel computers are classified into different categories based on various factors such as: Flynn’s classification, Fengs Classification, Handlers and Shores Classification, Classification based on granularity. Classification based on memory arrangement and communication among PEs. Classification based on interconnections among PEs and memory modules. Classification based on characteristic nature of PEs and Specific types of parallel architectures.

2.1 Flynn’s classification

Flynn’s classification scheme is based on multiplicity of instruction stream and data stream. Flynn’s classification is introduced by Michel J Flynn in the year 1966. Generally digital computer can be classified into four categories according to multiplicity of the instruction stream and data stream. SISD (Single Instruction Stream Single data Stream),SIMD (Single Instruction Stream Multiple data Stream),MISD (Multiple Instruction Stream Single data Stream),MIMD (Multiple Instruction Stream Multiple data Stream).SISD: - Most serial computers are SISD. Instruction are executed sequentially but may be overlapped during their execution phases. Most uniprocessor systems are pipelined. SISD computers may have more than one functional units, all the functional units are under the supervision of one control unit .SIMD:-In SIMD there are multiple processing elements supervised by the same control unit. All the processing units receive the same instructions and broadcast from the control unit but operates on different sets from distinct data stream. The shared memory subsystem may contain multiple modules. MISD: In this type of organization there are n processing units, each receiving distinct instruction over the same data stream and it derivates output of one processor became input of the next processor in the micropipeline (cascade of processors). This structure has received much less attention and has been challenged as impractical by some architecture .MIMD;-Multiprocessor system and multiple computers can be classified in these categories. Intrinsic MIMD architecture implies interaction among n-processors. All the memory streams are divide from the same data space and shared by all the processors. MIMD is the set of independent SISD uniprocessor systems. MIMD computer is tightly coupled in the degree of interaction among the processor is high otherwise we consider them as loosely coupled.

2.2 Fengs, Handlers and Shores Classification

Tse-yunFeng in 1972 has prposed a scheme based on serial versus parallel processing. Fengs classification is based on degree of parallelism. The maximum number of bits that can be processed within a unit time by a computer system is called maximum parallelism degree. Fengs classified the system into four types on the basis of sequential and parallel operations at bit and word levels as Word serial and bit serial, Word serial and bit parallel, Word parallel and bit serial finally word parallel and bit parallel.In 1977, Wolfgang Handler proposed an elaborate notation for expressing the pipelining and parallelism of computers. Handler's classification addresses the computer at three distinct levels: Processor control unit (PCU), Arithmetic logic unit (ALU) and • Bit-level circuit (BLC). Shores, in 1973 classified the computer on the basis of organization of the constituent elements in computer system .Six different kinds of machines were recognized and distinguished by numerals designator as Machine 1,Machine 2 upto Machine 6.

2.3 Classification based on grain size , Memory arrangement and Interconnection network

Classification based on grain size deals with recognizing the extent of parallelism in a program executed on a multiprocessor system. Types of Grain sizes :-Fine Grain: This type contains approximately less than 20 instructions. 2) Medium Grain: This type contains approximately less than 500 instructions and 3) Coarse Grain: This type contains approximately greater than or equal to one thousand instructions. Classification Based on Memory Arrangement and Communication among PEs can be classified into two major categories in terms of memory arrangement. These are: shared memory and message passing or distributed memory.Shared Memory architecture is also known as Multiprocessorsor Parallel Computing and Distributed Memory architecture isalso known as Multicomputer or Distributed computing. Types of Multiprocessor are Quadcore, Dualcore or any Multicore Processor. Types of Distributed Computing is Cluster Computing ,Grid Computing and Cloud Computing. Parallel architectures are also classified in terms of interconnecting network arrangements for communication among the various PEs, Basically there are two types of interconnection network static and dynamic. Types of Static Interconection Network are Linear array,Mesh,Ring,Hypercube and Cube connected.Types of Dynamics Interconnection network is Bus based ,Switch based and Crossbar. Parallel Architectures can be classified as Homogeneous and Heterogeneous. In a homogeneous parallel system all the PEs are identical. In a Hetrogeneous parallel system all the PEs are different. Parallel architectures are also classified in terms of the nature of the PEs comprising them. An architecture may consist of either only one type of PE or various types of PEs. The different types of processors that are commonly used to form parallel architectures are. CISC Processors.(Complex Instruction Set Computer) and RISC (Reduced Instruction Set Computer). [2]

3. MULTICORE ARCHITECTURE

A multicore places multiple processors on a single chip and each processor is called a core [3]. As we increase the capacity of chip placing multiple processors on a single chip became practical. These architectural designs are known as Chip Multiprocessors (CMPs), chip Multiprocessors are known as Multicore. A multi-core processor is a single with two or more independent processors. The instructions on multicore are ordinary CPU instructions, but the multiple processors can run multiple processes parallel at the same time by increasing the overall speed of the programs. Multicore span threads which divide the tasks between cores. It can execute multiple tasks at single time. Multicore is shared memory processors, all processors shares the same memory. Multicores are becoming popular for both server and desktop processors. By the next decade, it is expected to have processors with hundreds of cores on a chip.

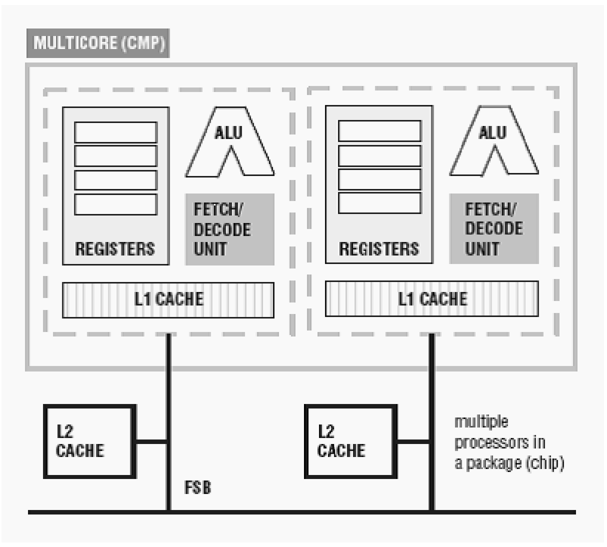


Fig 1. Multicore Architecture

4. SORTING

Given a sequence on n numbers{ a0, a1, a2,……. , an-1},the sorting problem is to find a permutation { a0', a1', a2',……. , an-1'}such that a0'<= a1<= a2<=',……. <= an-1'. [4] Sorting means arranging the number in ascending or descending order. Sorting is a common and important problem in computing. Given a sequence of *N* data elements, we are required to generate an ordered sequence that contains the same elements There are two types of sorting internal Sorting (algorithms that sort sequences small enough to fit entirely in primary memory) and External Sorting(orders a list o values too large to fit at one time in primary memory). Four sorting algorithm namely radix ,quick ,merge and bitonic sort is implemented . In radix-sorting algorithms, the pieces of the keys are of fixed size, so there is a fixed number of different values each piece could have. Indeed, it is usually the case that the R different possible values for each piece are the integers 0, 1, ..., R−1.[9] Radix-sorting algorithms treat the keys as numbers represented in a base-R number system, for various values of R (the radix), and work with individual digits of the numbers.[7]Radix sorts rely on a binary representation of the sort key. Each iteration of a radix sort processes b bits of the key, partitioning its output into 2 b parts. The complexity of the sort is proportional to(b) the number of bits, and (n) the size of the input (O(bn)), and fast scan-based split routines that efficiently perform these partitions have made the radix sort the sort of choice for key types that are suitable for the radix approach, such as integers and floating-point numbers.[10] However, as keys become longer, radix sort becomes proportionally more expensive from a computational perspective, and radix sort is not suitable for all key types/comparisons (consider sorting integers in Morton order [7].Quicksort is a well-known sorting algorithm developed by C. A. R. Hoare that, on average, makes O(nlogn) comparisons to sort n items. The key thing to note is that this implementation is nothing but a divide and conquers strategy where a problem is divided into subproblems that are of the same form as the larger problem. Each sub problem can be recursively solved using the same technique. Once partititon is done, different sections of the list can be sorted in parallel. If we have p processors, we can divide a list of n elements into p sublists in Θ(n) average time, then sort each of these in Θ ((n/p)log(n/p)) average time. Merge sort is a recursive algorithm that continually splits a list in half. If the list is empty or has one item, it is sorted by definition (the base case). If the list has more than one item, we split the list and recursively invoke a merge sort on both halves. Once the two halves are sorted, the fundamental operation, called a **merge**, is performed. Batcher's Bionic sort [6] is a parallel sorting algorithm whose main operation is a technique for merging two bitonic sequences. A bitonic sequence is the concatenation of an ascending and a descending sequence of numbers.To sort a sequence of n numbers, the Batcher's algorithm reqired following steps:. The first step is to convert the n numbers into a bitonic sequence with n=2 numbers in an increasing subsequence and n=2 numbers in a decreasing subsequence. After the bitonic sequence with n numbers is obtained, it is merged into an ordered sequence (either increasing or decreasing, depending upon which is needed). The merging technique consists of log n stages, and each stage consists of three operations: shuffle, compare, and unshuffle.

5. OPENMP

OpenMp is an API (Application Program Interface) that use multithreaded and shared memory parallelism. Openmp is basically divided into three parts Compiler directives, runtime library routines and environment variable. It is an open specification for multiprocessing. OpenMp worked as a fork-join model where fork is master thread that use to create a team of parallel thread and join is used when the team of parallel threads complete their task they synchronize and terminate and left the master thread to execute sequential program. OpenMp visualize as parallel programming model on multicore architecture [8].

6. PERFORMANCE MEASUREMENTS

Performance of a parallel algorithm is measured using two factors speed-up and efficiency. [11]

6.1 Speedup

In parallel computing, speedup refers to how much faster a parallel algorithm is run in parallel.

$$\text{Speed up} = \frac{\text{Sequential execution time}}{\text{Parallel execution time}}$$

Speed up depends on the ratio of the amount of time your code spends communicating to the amount of time it spends computing.

6.2 Efficiency

In parallel computing, efficiency refers to speed up divided by number of processors. Efficiency is a measure of how much of your available processing power is being used.

$$\text{Efficiency} = \frac{\text{Sequential execution time}}{\text{Parallel execution time X processor used}}$$

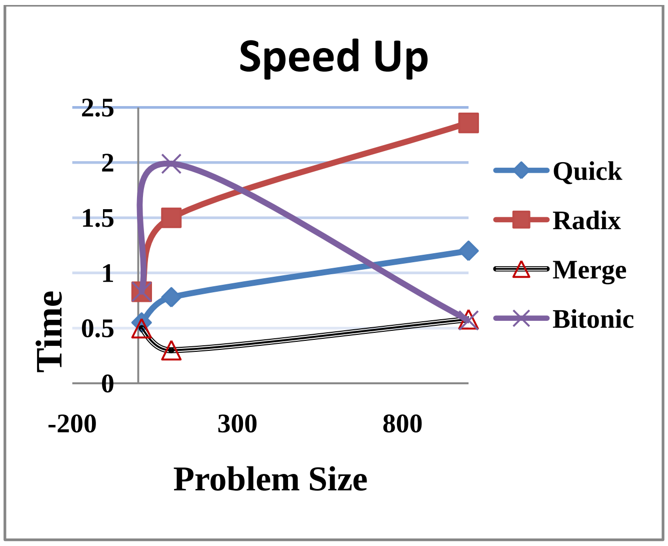
7. EXPERIMENTAL SETUP

For Experiment a computer system is taken with dual core processor having 1.83 GHz speed and Linux operating system. We have run the sequential sorting and parallel sorting on dual core processor. Execution time in seconds was recorded as shown in Table I for dual core and analyzed graphically.

Table 1: Speed up of sorting algorithms for dual core processors

Sorting	Problem Size	Sequential time(Sec)	2 Threads	4 Threads	Speedup
Quicksort	10	0.000015	0.000027	0.000071	0.55
	100	0.000095	0.000121	0.000129	0.78
	1000	0.000197	0.000163	0.000179	1.2
Radixsort	10	0.0000232	0.0000279	0.0000304	0.83
	100	0.0000317	0.0000291	0.0000297	1.5
	1000	0.0000357	0.000151	0.000175	2.36
Mergesort	10	0.0000232	0.0000279	0.0000304	0.5
	100	0.0000317	0.000159	0.000057	0.3
	1000	0.000101	0.000175	0.000151	0.58
Bitonic Sort	8	0.00039	0.00021	0.00019	2.05
	128	0.0003	0.00059	0.00061	0.6
	1024	0.00041	0.00023	0.00028	1.7

Graph was plotted for Speedup in Linux platform in dual core processor.



The horizontal axis represents problem size and vertical axis represents execution time in milliseconds.

8. CONCLUSION

In this experiment we have calculated execution time and speedup for four different sequential and parallel Sorting.Performance wise merge sortand bitonic sort are better as compared to others for larger problem size. It is clear from the graph as OpenMp parallaize sequential program performance get increases for higer problem size while for lower problem size execution time with parallalization is more than sequential Sorting.Beyond a certain optimum problem size only parallization is effective below that point because of communication overhead sequential algorithm on sequential machines will give better results. Below optimum problem size following overheads like interprocess communication overhead, synchronization and concurrency prominently play their roles.

9. REFERENCES

[1] Cheng, John, Max Grossman, and Ty McKercher. *Professional Cuda C Programming*. John Wiley & Sons, 2014.

[2] http://www.springer.com/cda/content/document/cda_downloaddocument/9781852335991-c1.pdf?SGWID=0-0-45-80044-p2275868

[3] Cameron, H., Tracy, H., Professional Multicore programming, Wiley publication,2008.

[4] Quin M "parallel programming in Cwith MPI and OpenMP" Tata McGraw Hills edition 2000.pp338

[5] http://www.cs.princeton.edu/courses/archive/spring09/cos226/handouts/Algs3Ch10.pdf

[6] K. E. Batcher. Sorting networks and their applications. In *AFIPS Springer Joing Computer Conference*, pages 307–314, Arlington,VA, April 1968

[7] G. Morton. A Computer Oriented Geodetic Data Base and ANew Technique In File Sequencing. International BusinessMachines Co., 1966.

[8] Venkatesan P., Harish B., S. Sarholz, Proceedings of the 3rd international workshop on OpenMP "A Practical Programming Model for the Multi-core Era", 2008.

[9] Davidson A, David T, Michael G, John D. Owens," Efficient Parallel Merge Sort for Fixed and Variable Length Keys".

[10] http://www.cs.princeton.edu/courses/archive/spring09/cos226/handouts/Algs3Ch10.pdf

[11] Kai hwang, Naresh Motwani, Advanced computer Architecture, pp 108 chapter 3