ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 5, Issue 8, August 2016

Performance Analysis of Sorting Algorithms in Multicore Architeture

Varsha Thakur[1], Sanjay Kumar [2]

[1] Research Scholar, Pt.Ravishankar shukla University ,Raipur Chhatisgarh ,India.
[2] Associate Professor, Pt.Ravishankar shukla University ,Raipur Chhatisgarh ,India

Abstract As technology has advanced, Multicore Computing has emerged as a research area with the potential of providing satisfactory and faster result for real time applications. Multicore architecture is those that emphasize on more than one processors work together for Parallel Processing. Multicore is a logic circuit in which two or more processors are attached for performance enhancement and processes basic instruction that are needed by the system. This paper presents comparisons of few sorting algorithm on multicore architecture. Performances are analyzed on the basis of execution time of parallel sorting algorithms in multicore processors. To implement these algorithms we have used C programming language with OpenMp Libraries under Linux environment.

Index Terms **Bitonic Sort, Multicore, OpenMp, Sorting.**

I. INTRODUCTION

A Multicore processor is an Integrated Circuit in which "A Multi-core processor is typically a single processor which contains several cores on a chip". The cores are functional units made up of computation units and caches [7].A Multicore processor is an Integrated Circuit in which more than one processor or core are included for performance improvement and Simultaneous processing of parallel jobs. Moore's Law has been proven to be true overthe passage of time - the performance of microchips has been increasing at an exponential rate, doubling every two years. In most proposed multicore platforms, different cores share the common memory. The multiple cores inside the chip are not clocked at a higher frequency, but instead their capability to execute programs in parallel is what ultimately contributes to the overall performance making them more energy efficient and low power cores [6]. Multi-core processors could be implemented in many ways based on the application requirement. It could be implemented either as a group of heterogeneous cores or as a group of homogenous cores or a combination of both. In homogeneous core architecture, all the cores in the CPU are identical On the other hand heterogeneous cores consist of different core with different capabilities. A Sorting Algorithm is an algorithm that arranges the elements of a list in a certain order. The most-used orders are numerical order and lexicographical order. OpenMp is used for parallelizing the sequential matrix multiplication. In rest of paper we have define some basic concept of OpenMP, Multicore Architecture and Soting Algorithms.

A. OPENMP

OpenMp is an API (Application Program Interface) that use multithreaded and shared memory parallelism. Openmp is basically divided into three parts as 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 [3] [4].Fork-join model of parallelism is provided in OpenMP, where the programmer specifies parallelism in code using OpenMP directives. The OpenMP directives are embedded in the code either as special comments in FORTRAN or as programs in Cand C++. Parallel regions and work sharing constructs enable the programmer to express parallelism at the level of structured blocks within the program, such as loops and program sections[5].

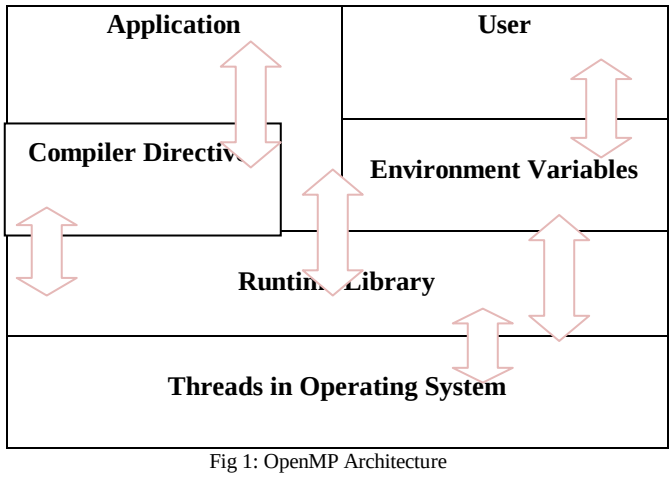


Fig 1: OpenMP Architecture

B. Muticore Architecture

A multicore places multiple processors on a single chip and each processor is called a core [2]. 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.

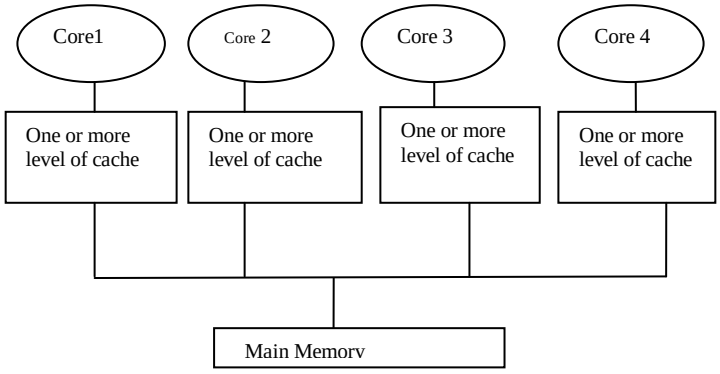


Fig 2- Multicore

II. SORTING

Given a sequence on n numbers{ a_0, a_1, a_2,..... , a_{n-1}},the sorting problem is to find a permutation { $a_{0'}$, $a_{1'}$, $a_{2'}$,..... , $a_{n-1'}$}such that $a_{0'} <= a_{1<=} a_{2<=',.....} <= a_{n-1'}$. [6] 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.[7] 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.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.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 [9] 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 (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. Complexity of various algorithms is shown in the table below-

Table I: - Complexity of sorting algorithms

Sorting	Average Case	Worst case	Best	Space complexity(worst)
Quick Sort	O(n log n)	$O(n^{2)}$	O(n log n)	O(log n)
Merge Sort	O(n log n)	O(n log n)	O(n log n)	O(n)
Radix Sort	O(nk)	O(nk)	O(nk)	O(n+k)
Bitonic Sort	$O(\log(n)^2)$	$O(\log(n)^2)$	$O(\log(n)^2)$	$O(n \log(n)^2)$

III PERFORMANCE MEASUREMENTS

Performance of a parallel algorithm is measured using two factors speed-up and efficiency.[10]

A. 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.

B. 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.

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

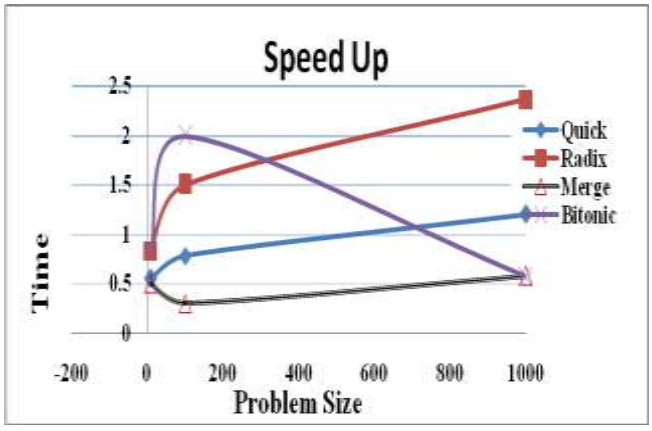
IV. EXPERIMENTAL SETUP

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 was recorded as shown in Table I for dual core and analyzed graphically.

Table II: - Speed up for sorting algorithm on dual core processors.

Sorting	Problem Size	Sequential time	2 Threads	4 Threads	Speedup
Quick Sort	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
Radix Sort	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
Merge Sort	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.

V. Conclusion

In this experiment we have compared execution time and speedup for different Sorting algorithms.Merge sort and 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

REFERENCES

[1].Lizhe Wang, Jie Tao, Gregor von Laszewski, Holger Marten. 2010, "Multicores in Cloud Computing: Research Challenges for Applications", Journal of Computers, Vol 5, No 6 (2010)

[2]. R. Ramanathan. Intel multi-core processors: Making the move to quad-core and beyond. Technology@Intel Magazine, Dec 2006.

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

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

[5]. Basumallik, Ayon Ph.D., Purdue University. Compiling SharedMemory Applications for Distributed-Memory Systems. Major Professor: Rudolf Eigenmann , December, 2007.

[6]. Quin Michael J."Parallel programming in C with MPI and OpenMP". McGraw Hill Inc., 2004.

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

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

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

[10] .Kai hwang,Naresh Motwani,:Advanced computer Architecture, pp 108 chapter 3

Varsha Thakur received his MCA. from NIT (FORMLY GEC)Raipur Chhattisgarh, India and is currently enrolled as a Ph.D. scholar in the School of Studies of Computer Science and Information Technology , Pt. Ravishankar Shukla University Raipur, Chhattisgarh , India. She is having 6 Year Teaching Experience. Her research area is Load balancing methodologies in parallel and distributed computing.

Dr. Sanjay Kumar received his B. E from Govt.Engg.college, M.E. (Computer Science & Engg.) from Motilal Nehru Regional Engineering College, Allahabad in 2000 and Ph.D. (Computer Science & Engg.) from Ravishankar Shukla University, Raipur in 2005. He is presently serving as Associate professor and Head, Computer Science and Information Technology School of Studies, Pt.Ravishankar Shukla University Raipur Chhattisgarh, India. He has about 15 years of experience in teaching and 6 yr in others. His is current research interest includes Networking and parallel and distributed.