Comparative Study of Parallel Odd Even Transposition and Rank Sort Algorithm

Manpreet Kaur

Abstract- In this paper, the execution behaviours of different parallel sorting algorithms like odd-even transposition sort and parallel rank sort have been invested with multithreading. Multithreading in JAVA programming language provides a mechanism where parallel algorithms are implemented by far. The performance of implemented algorithms is evaluated on the basis of execution time. It has been found that parallel odd even Transposition algorithm is giving better performance as compared with rank sort algorithm.

Keywords: Parallel sorting algorithms, performance analysis, Multithreading.

I. INTRODUCTION

A sorting algorithm is an algorithm that puts elements of a list in a certain order. Sorting is one of the most important operations in database systems and its efficiency can influences the overall system performance drastically. To speed up the performance of database system, parallelism is applied to the execution of the data administration operations [1]. Sorting is the most common operations in parallel processing applications. For example, it is central to many parallel database operations and important in areas such as image processing, statistical methodology and so on. A number of different types of parallel sorting schemes have been developed. A number of researchers have worked addressing the parallel sorting algorithms as the central issue.

II. PARALLEL SORTING ALGORITHMS

A parallel algorithm is an algorithm in which several computations are carried on simultaneously. Here two parallel sorting Algorithms are implemented and evaluated. These are:

A. Parallel Odd-even transposition sort.
B. Parallel rank sort

2.1. Odd-Even Transposition

Odd-even sort is a relatively simple sorting algorithm. It is a comparison sort based on bubble sort with which it shares many characteristics. It functions by comparing all (odd, even)-indexed pairs of adjacent elements in the list and, if a pair is in the wrong order (the first is larger than the second) the elements are switched. The next step repeats this for (even, odd)-indexed pairs (of adjacent elements). Then it alternates between (odd, even) and (even, odd) steps until the list is sorted. Odd-even transposition sort a refinement of the bubble sort in which adjacent pairs, starting with an odd position, are sorted then adjacent pairs, starting with an even position, are sorted. The two phases alternate until sorting is completed. [2].

The parallel Odd-even transposition sort algorithm [3, 4] starts by distributing n/p sub-lists (p is the number of processors) to all the processors. Each processor then sequentially sorts its sub-list locally. The algorithm then operates by alternating between an *odd* and an *even* phase, hence the name *odd-even*. In the even phase, even numbered processors (processor i) communicate with the next odd numbered processors (processor i+1). In this communication process, the two sub-lists for each 2 communicating processes are merged together. The upper half of the list is then kept in the higher number processor and the lower half is put in the lower number [1].

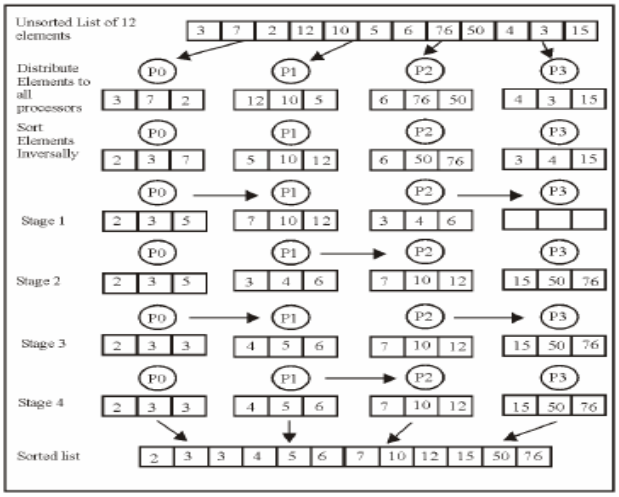

Figure 1: Parallel Odd-Even Transposition Sort

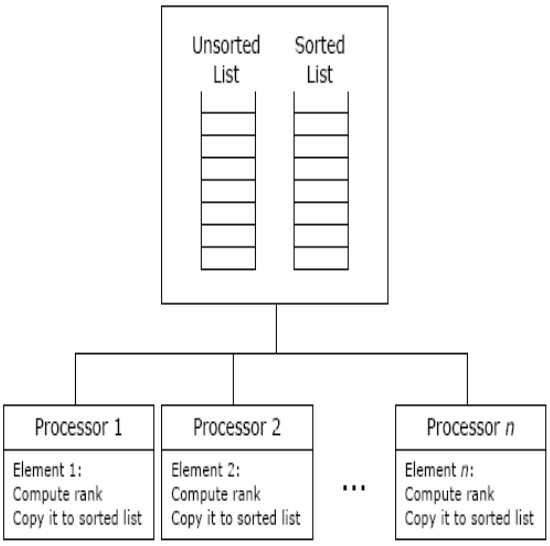

Figure 2: Parallel Rank Sort

Manuscript Received on November 2014.

Ms. Manpreet Kaur, Asst. Prof., in PG Department of Computer Applications, Khalsa College for Women, Amritsar, India.

2.2. Parallel Rank Sort

Rank sort is a simple parallel sorting algorithm where each element of an array is compared with every other element of the array to see which is larger. The rank of an element is defined as the total number of elements less than the element. The final position of an element in the sorted array is just its rank. The parallel version of the Rank Sort algorithm can be easily obtained by computing the rank of each element from the list independently on a different processor. Processor number 1 can compute the rank of the first element by comparing it with every other element in the list. In the same time, processor number 2 can simultaneously compute the rank of the 2nd element from list and so on.[5] If there are n processors in the parallel system, each processor i can be assigned to compute the rank of the element number i in the list. Also, the parallel Rank Sort algorithm is using two arrays (unsorted and sorted lists) stored in the shared memory area. In such a way the lists will be available to all the processors from the system

III. RESULTS AND ANALYSIS

The JAVA programming language is used to develop the sorting algorithms. The performance of the sorting algorithms is evaluated on an ACPI uniprocessor PC with Intel® Pentium®4 CPU 1.80GHz with 512MB RAM. An array 5800 random integers is used to test the parallel algorithms.

Parallel Odd-Even Transposition Table1 shows the total execution time for the odd-even transposition sorting algorithm both for sorted and unsorted data. In figure3 and figure4 an illustration of the total execution time of the algorithm is displayed.

Parallel Rank Sort is the slowest algorithm for sorting elements because each processor needs its own copy of the unsorted list thus, in turn, raises a serious communication overhead. The parallel rank sort algorithm can be considered as a *memory intensive* algorithm. Table 2 shows the total execution time for the parallel rank sort algorithm for unsorted and sorted data.

Table 1: Total Execution time of odd-even transposition sort for unsorted and sorted data

Integers	Time (unsorted data)	Time (sorted data)
48	0	0
96	16	0
498	94	31
900	391	78
1200	859	94
1500	1515	95
1800	2657	125
2100	4078	135

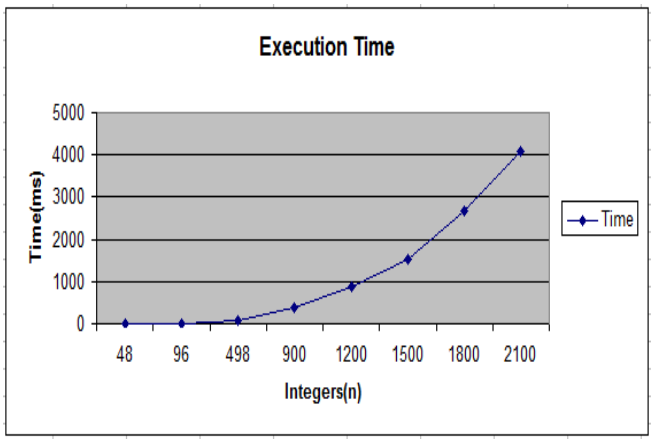

Figure 3: Total Execution Time of the Parallel Odd-Even Algorithm (Unsorted List)

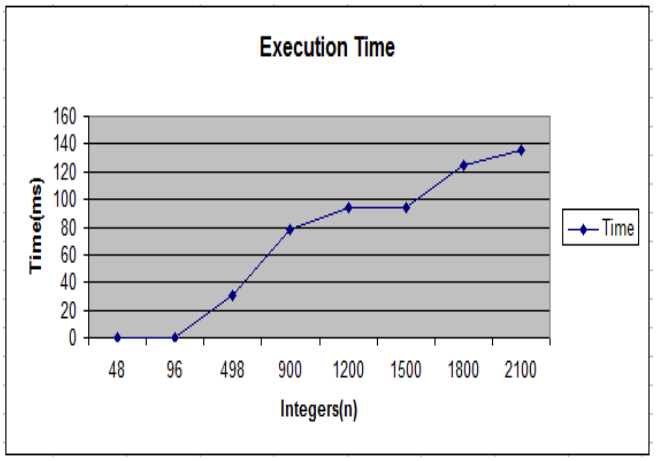

Figure 4: Total Execution Time of the Parallel Odd-Even Transposition Algorithm (Sorted List)

Table 2: Total Execution time of rank sort algorithm for unsorted and sorted data

Integers	Time (unsorted data)	Time (sorted data)
50	16	15
100	16	16
500	125	109
1000	218	203
2500	469	453
5000	1078	984
7500	1766	1563
10000	2562	2203

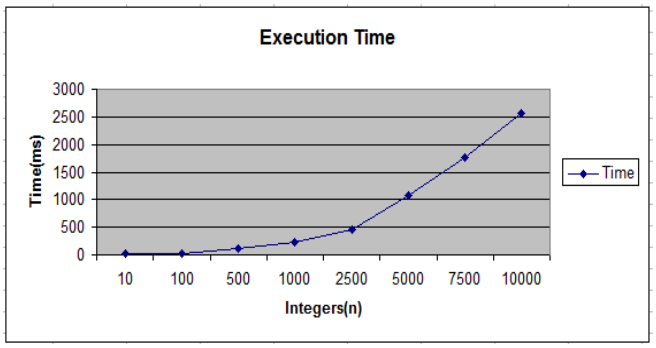

Figure 5: Total Execution Time of the Parallel Rank Sort Algorithm (Unsorted List)

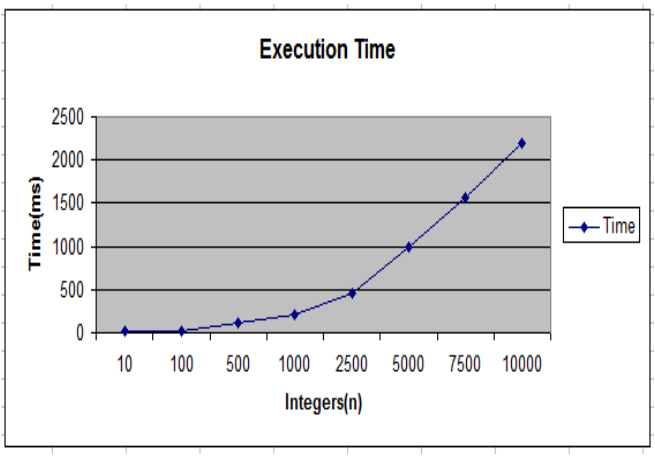

Figure 6: Total Execution Time of the Parallel Rank Sort Algorithm (Sorted List)

IV. CONCLUSIONS

Two parallel sorting algorithms have been developed and executed on multithreading. The parallel algorithms implemented are the odd even transposition sorting algorithm, the parallel rank sort algorithm. Table3 shows a comparison between the two parallel sorting algorithms from figure7, it has been analyzed that the parallel odd even transposition sorting algorithm is the fastest sorting algorithm for sorting up to 50000 random integers. It is found that parallel odd even transposition sort is 1.3 times faster than parallel rank sort

Table 3.Comparison of parallel sorting algorithms

Integers	Rank Sort	Odd-Even Sort
100	109	62
1000	219	63
5000	953	469
10000	1969	1125
15000	2734	1781
20000	3703	2562
25000	4657	3453
50000	5718	4141

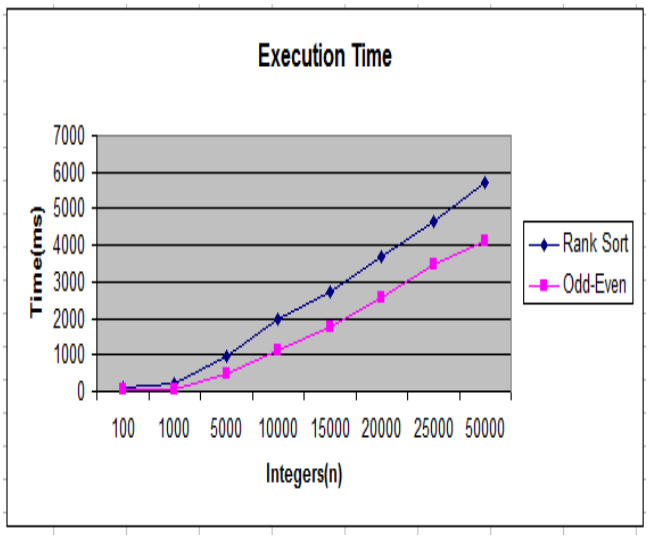

Figure 7: A Comparison of the Total Execution Time Required for Sorting 50000 Integers using Parallel Rank Sort and Odd-Even Transposition

REFERENCES

[1] Kalim Qureshi and Haroon Rashid,"A Practical Performance Comparison of Parallel Matrix Multiplication Algorithms on Network of Workstations.", IEEE Transaction Japan, Vol. 125, No. 3,2005.

[2] www.encyclopedia.com/doc/1O11-oddeventranspositionsort.html

[3] D. Bitton, D. DeWitt, D.K. Hsiao, J. Menon, A Taxonomy of Parallel Sorting, ACM Computing Surveys, 16,3,pp. 287-318.

[4] Song, Y.D., Shirasi, B. A Parallel Exchange Sort Algorithm. South Methodist University, IEEE.

[5] www.economyinformatics.ase.ro/content/EN5/alecu.pdf

[6] Parallel Rank Sort Assist Lecturer Felician ALECU Economy Informatics Department, A.S.E Bucharest, Economy Informatics.

[7] S.Lakshmivardan and S.K Dhall " Analysis and Design of Parallel Algorithms " McGraw-Hill 1990

[8] "Parallel Programming techniques and applications using networked workstations and parallel computer" Barry Wilkison and Michael Allen.

Ms. Manpreet Kaur, is Assistant Professor in Khalsa College for Women. She is M.Tech from Guru Nanak Dev University, Punjab and has published about 10 research papers in both International Conferences as well as Journals.