A New Efficient Algorithm based on Odd-Even Transposition Sort Algorithm for Mergesort on Grid Computer

Zaidatul Aqmar binti Zakaria [#1], Mohamed Faidz Mohamed Said [#2]

[#] *Universiti Teknologi MARA*
70300 Seremban Negeri Sembilan, MALAYSIA

[1] zaiaqmar74@gmail.com

[2] faidzms@ieee.org

***Abstract* –This paper discussed about a new parallel sorting algorithm, which is imitative from the odd-even Merge Sort algorithm, titled "partition and concurrent merging" (PCM). To categorize all the components, the comparisons of individual elements must be generated professionally. A divide and conquer strategy Merge Sort are used to sort an array proficiently while making a smallest number of contrasts between the array components. Merge sort is known as one of the most effective ways to figure out the sorting problem. The research recommends that it is approachable to knowing optimizations. A fast variation of Merge sort also has been proposed significantly in this paper. So, it is quite efficient for using the Odd-Even Transposition sort algorithm by Merge Sort.**

Keywords: **odd-even transposition sort algorithm, mergesort, grid computer**

I. INTRODUCTION

Sorting is the supreme significant operation in the database systems. Meanwhile it can effect the whole system of performance and it is one of the greatest mutual applications in the computer science. Besides, the progression is going through to which organization of data according to their values [3]. Parallelism was applied to the execution of the administration operation's data to speed up the performance.

The study of some sorting algorithm is to come out with the most efficient sorting algorithm and the capacity to orchestrate a variety of components into a characterized request is the essential thing in Computer Science. Sorting typically is utilized with online stores, were the request of administrations or things were obtained figures out what requests can be filled and who gets their first request. Sorting is used for the database management system especially for the bank and financial systems [4]. This is for making them simple to track and rank the billions of exchanges that are experiencing in one day.

Numerous calculations give a similar effectiveness however they do not have a similar speed on a similar info [5]. The comparison for their criteria is will be shown based on:

- Calculations must be judged in light of their normal case, best case, and most pessimistic scenario proficiency.
- Their space requirement.
- Stability

There are various algorithms that offer the answer specially to sorting the arrays such as Bubble Sort, Selection Sort, Merge Sort, Insertion Sort and Quick Sort. All of these algorithms are programmatically correct but they are unable for arrays to cooperate with a huge number of elements and show time complexity for quadratic [4].

Computations must be judged in light of their ordinary case, best case, and most critical situation capability. To be sorted is appropriated thus creates a conveyed sorted succession was the expecting of the calculation [6]. This paper will depict around a utilization of the union sort inside a parallel handling environment. In the completely parallel model, it over and again split the sub records down to the point where it has single-component records. And after that it will combine these in parallel move down of handling tree until it gets the completely consolidated rundown at the highest point of the tree.

II. BACKGROUND

Merge sort, it is a partition and vanquish external the sorting calculation. The basic process was elaborate in merge sort is to split the list to be organized into two slighter sub-lists and recursively repeat this technique till the last one element is left in the sub-list. Next the many sorted sub-lists are merged back to form the higher level in the list that is parent list. This merging process will go on recursively till the sorted original list is reached. There are a portion of the hindrances in existing framework is:

- The high of the time taken to resolve the problem.
- The high of the number of iterations.

To overcome to these restrictions, another method will be utilized to explicitly decrease the time, and in the meantime, it can be used to lessen the quantity of emphases for some sum.

Merge sort is a stable divide and conquer algorithm that will solve all the sorting efficiently and it requires an extra array of the size. Although a lot of research work had to be carried out instead of improving the performance of Merge sort to sort the input array and it is not in place of sorting algorithm. But the research offers a better implementation of Merge sort that will save a lot of time and also the space. Proposed algorithm will provide tremendously compact and fast Merge function and also it retains the advantages of existing Merge sort variations [2].

The Odd Even Merge Sort, it is used to sort a bunch of the numbers in efficient way and it is different from all the existing sorting techniques. This sorting system is range to the consolidation sort. Really sorting is to orchestrate the components in the succession [3].

III. METHODOLOGY

The Odd-even transposition sort algorithm starting by distributing n/p sub-lists that mean p is the number of processors to all the processors. Each processor then chronologically sorts locally by sub-list. The algorithm then works by alternating between an *odd* and an *even* phase. Hence, the name *odd-even* was referring to even numbered processors (processor i) and communicate with the other odd numbered processors (processor i+1) [7].

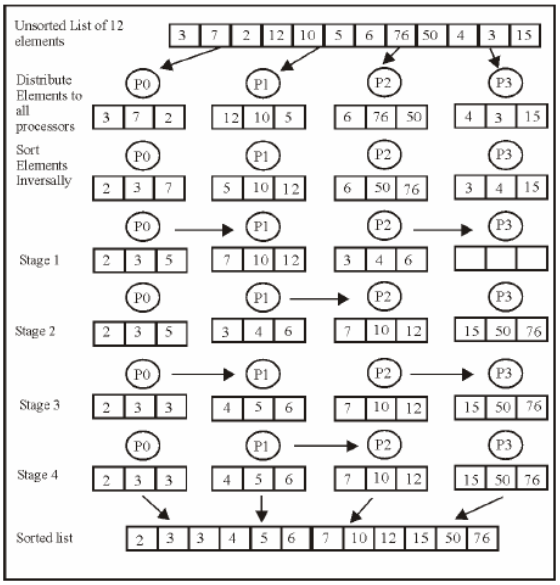

Figure 1. [7]

Figure 2 shows the odd-even transposition sort will relate the contiguous combine of the information in an exhibit that will be sorted and if a couple is found to the wrong request then those components are swapped. For the initial step is thought about and swapped if found in wrong request for odd record and the nearby even list components. The following stride, even file and the neighbouring odd list components are analysed, and they are additionally swapped on the off chance that they are found in wrong request. This stage procedure will ceaselessly be for (odd, even) and (even, odd) until no swapping of information components are required [8].

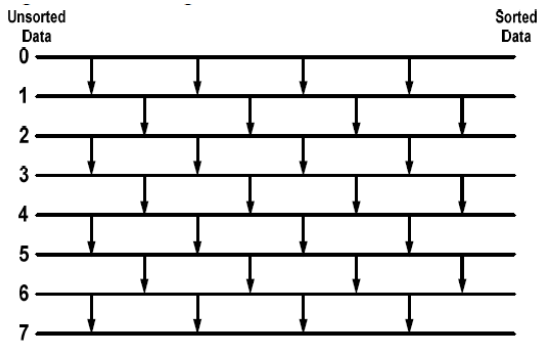

Figure 2. [8]

The knowledge was developed from the literatures that are being applied in designing of architectural models that are presented in Figure 3. It is divided into three basic components. That are repository, shuffling module, and sorting module. Evaluating performance of sorting algorithms was formulated to assist as benchmark of components to be developed. The data stored in the repository designed for evaluating the algorithms and the data may be ordered or unordered in nature [9].

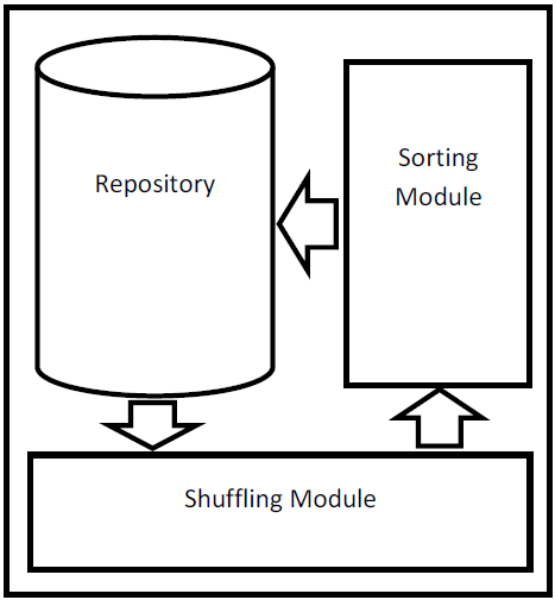

Figure 3. [9]

Divide and Conquer of algorithm is known as Merge Sort. The input array was partitioned into two parts, it calls itself for the parts and afterward blends the two sorted parts. Furthermore, the purpose of merg() is to use for merging the two parts. The key process of the two sorted sub-arrays into one is merge (arr, l, m, r) that undertakes arr[l..m] and arr[m+1..r] are sorted and merges [10]. But using

Merge Sort it also has the specialists and convicts. There are the list of them [11]:

Specialists:

- O(n log n) for time complexity.
- It can be utilized sorting for both inner and outer.

Convicts:

- In any event twice of the memory prerequisites of alternate sorts since it is recursive.
- Very high for the space complexity.

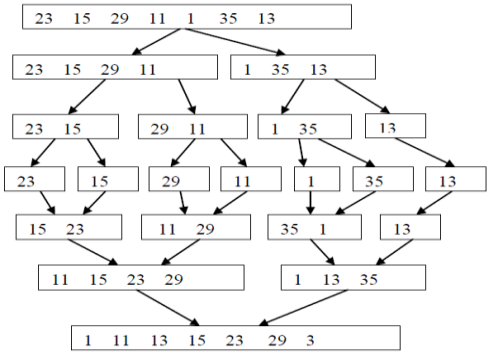

Figure 4. [11]

The information technologies of development was categorized by the expansion of applications which requires a lot of parallel sorting of data sets in real time [12]. Merge Sort parallelizes better, constant sort, and it is more effective for conduct slow to access sequential media. Merge Sort commonly have been selected as a top choice to sort a linked list [13]. Table 1 show the execution time and number of elements are as follow:

Table 1. [13]

Number of elements	Running time (ms)
10000	728
20000	1509
30000	2272

Next, a new fast parallel sorting algorithm will be termed as PCM (*parallel and concurrent merging*). After the underlying segment venture of the information succession was done, the algorithm will work in the two stages. The first stage sort is in parallel that all the sub-sequences will be using the Quicksort and the second stage is an iterative development. The merges and also the sorts will sub-sequence by pairs [14].

IV. SUMMARY

The aim of this paper research is to introduce the Merge Sort algorithm and make the confirmation about improvements and predecessors for Merge Sort itself. This research work proposes that the algorithm is significantly faster than the current best algorithm. The space requirement only needs 50 % of what is required by the classical Merge sort. Merge Sort have shown that it is 32 times quicker for arrays and higher than 100 components. So, it has proved that Merge Sort is significantly more productive than the others. For the array sizes that are below and greater than 100, the CPU runtimes were observed to be factually unique around 95% certainty interim. Furthermore, Merge Sort has also been proved to be well in the poorest case for array sizes above 100. This announcement can be clarified by the overhead basic for the creation and converging of all the exhibits along the consolidation step. An upcoming upgrading to the Merge Sort algorithm must find an effective technique to minimize this overhead.

REFERENCES

[1] M. T. V. S. V. Mr.Bobba Veera Mallu, Mr.K.Srinivasulu Achari, "Novelty Approach of “Odd-Even Transposition Technique”," vol. 3, 2013.

[2] D. Abhyankar, "A Fast Merge Sort," *International Journal of engineering Research and Applications*, vol. 1, pp. 131-134.

[3] V. Pravalika, L. K. R. Babu, R. DileepKumar, and M. T. Reddy, "Odd Even Merge Sort based on INDEX."

[4] R. Hibbler, "Merge Sort," ed: Dept. of Computer Science. Florida Institute of Technology. Florida, USA, 2008.

[5] P. Sareen, "Comparison of sorting algorithms (on the basis of average case)," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, pp. 522-532, 2013.

[6] K. Manwade, "Analysis of Parallel Merge Sort Algorithm," *Analysis*, vol. 1, 2010.

[7] H. Rashid and K. Qureshi, "A practical performance comparison of parallel sorting algorithms on homogeneous network of workstations," *WSEAS Transactions on Computers*, vol. 5, pp. 1606-1610, 2006.

[8] D. Prasad, M. Y. M. Yusof, S. S. Palai, and A. H. Nawi, "Sorting networks on FPGA," in *Proceedings of the 10th WSEAS international conference on Telecommunications and informatics and microelectronics, nanoelectronics, optoelectronics, and WSEAS international conference on Signal processing*, 2011, pp. 29-31.

[9] D. Aremu, O. Adesina, O. Makinde, O. Ajibola, and O. Agbo-Ajala, "A Comparative Study of Sorting Algorithms," *African Journal of Computing & ICT*, vol. 6, 2013.

[10] G. Kocher and N. Agrawal, "Analysis and Review of Sorting Algorithms," *Analysis*, vol. 2, 2014.

[11] M. J. S. S. Miss. Pooja K. Chhatwani, "Comparative Analysis & Performance of Different Sorting Algorithm in Data Structure," vol. Volume 3, p. 8, November 2013.

[12] O. S. Ivan Tsmots, Volodymyr Antoniv, "Parellel Algorithm and Structure for Implementation of Merge Sort.pdf," vol. 5, p. 10, March 2016.

[13] K. S. Al-Kharabsheh, I. M. AlTurani, A. M. I. AlTurani, and N. I. Zanoon, "Review on Sorting Algorithms A Comparative Study," *International Journal of Computer Science and Security (IJCSS)*, vol. 7, pp. 120-126, 2013.

[14] E. Herruzo, G. Ruiz, J. I. Benavides, and O. Plata, "A New Parallel Sorting Algorithm based on Odd-Even Mergesort," pp. 18-22, 2007.