

Estimation of Execution Time and Speedup for Bitonic Sorting in Sequential and Parallel Environment

Megha Jain
 S.o.S In CS & IT,
 Pt. Ravi Shankar Shukla
 University, Raipur (C.G), India

Sanjay Kumar
 S.o.S In CS & IT,
 Pt. Ravi Shankar Shukla
 University, Raipur (C.G), India

V.K Patle
 S.o.S In CS & IT,
 Pt. Ravi Shankar Shukla
 University, Raipur (C.G), India

ABSTRACT

The Batcher's bitonic sorting algorithm is one of the best parallel sorting algorithms, for sorting random numbers in modern parallel machines. Load balancing property of bitonic sorting algorithm makes it unique among other parallel sorting algorithms. Contribution of bitonic sorting algorithm can be seen in various scientific and engineering applications. Research on a bitonic sorting algorithm has been reported by various researchers in order to improve up the performance of initial batcher's bitonic sorting algorithm. In this paper, time estimation of bitonic sort algorithm is done in both sequential as well as in parallel domain.

Keywords

Parallel Algorithm, Parallel Sorting algorithm, Bitonic Sort, Sorting Network, OpenMP.

1. INTRODUCTION

In computer science, sorting has been an interesting research area in both theoretical and practical significance. Sorting is the process of arranging items in some sequence. However, sorting is the very common activity in both sequential as well as in parallel systems. The various parallel sorting algorithm was introduced in the parallel computer era. The Bitonic sorting algorithm was introduced in 1968 by Ken Batcher. This algorithm is capable of sorting N elements in time complexity of $O(\log^2 N)$ with $O(N \log^2 N)$ cost complexity.

1.1 Parallel Algorithm

If there are N concurrent processes, if $N=1$ then the sequential algorithm runs on a uniprocessor machine, and if $N>1$ then parallel algorithm runs on parallel machines. Parallel algorithm is opposed of sequential algorithm. However, parallel algorithm can be executed on different processors by using parallel algorithmic techniques. Techniques such as divide and conquer, randomization, parallel pointer techniques etc.

1.2 Parallel Sorting Algorithm

Parallel Sorting Algorithms are used in parallel computers for sorting the numbers. Sorting in parallel machine is achieved by dividing and computing data separately among processors. It generally requires a fixed number of merging and exchange data. However, In parallel computer execution time decreases with the increase in the number of processors, but execution time is not just half then sequential machine.

There are several parallel sorting algorithms such as bitonic sort, radix sort [10] etc. Bitonic sort algorithm is efficient sorting algorithm among other parallel sorting algorithms. A load balancing feature of bitonic sort algorithm makes it unique among all other parallel sorting algorithm. With the load balancing feature, each processor under parallel machine, assign an equal number of keys.

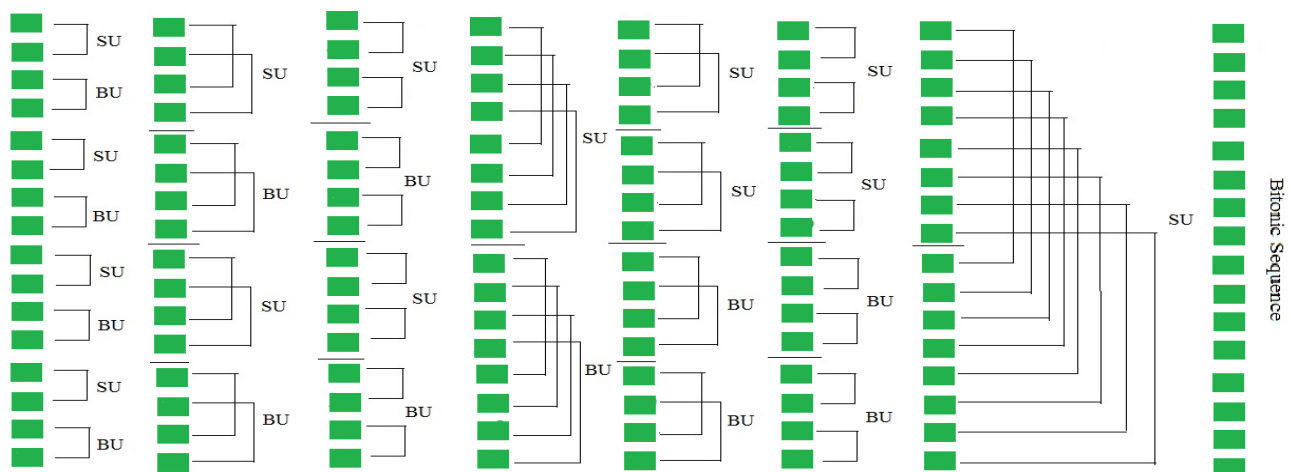


Figure 1: Bitonic sequence of 16 elements

2. BITONIC SORT

Bitonic sort is a comparison-based sorting algorithm. It converts a random sequence of numbers into a *bitonic sequence*, one that monotonically increasing and decreasing and then in the sorted sequence

2.1 Bitonic Sequence

A bitonic sequence of n elements ranges from x_0, \dots, x_{n-1} with steps (a) Existence of the index i , where $0 \leq i \leq n-1$, such that there exist monotonically increasing sequence from x_0 to x_i and monotonically decreasing sequence from x_i to x_{n-1} . (b) Existence of the cyclic shift of indices by which characteristics (a) satisfy [20]. After performing above stated step bitonic sequence occur. The Bitonic sorting algorithm works on a random sequence of power two i.e. 2,4,8,16 etc. The bitonic split operation is applied for producing two bitonic sequences on which merge and sort operation is applied as shown in Fig 1 refer from [22].

3. SORTING NETWORK

Sorting networks, also known as comparison networks for sorting. The comparison based network consists components like wires and comparators. The initial random sequence enters through input wires. These input wires are attached to the level of the comparator, where each comparator accepts 2 inputs from external source. Comparator after performing its job, i.e. compare it supply output to an external source. There two types of comparator as shown in Fig 2 below i.e. Fig 2a Increment Comparator and fig 2b Decrement Comparator.

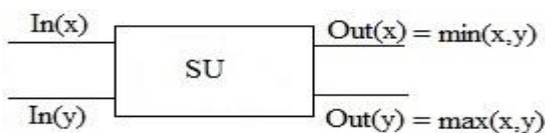


Figure 2a : Increment Operator

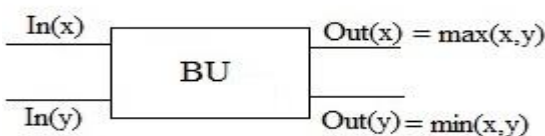


Figure 2b : Decrement Operator

Both Increment and Decrement Comparator take two inputs, i.e. In (x) and In (y) after comparison it produce output, i.e. Out (x) and Out (y). Increment Comparator performs ascending order with SU(small upper) comparator label while Decrement Comparator performs descending order with BU(big upper) comparator label.

3.1 Bitonic Sorting Network

The Bitonic sorting network is a compare and merge network. It works on comparing and merging technique. However, compare and merge operation is done in odd number pattern, i.e. 1-0-1-3-1-0 for 8 input size, 1-0-1-3-1-0-7-3-1-0 for 16 input size etc. A large network can be split up into several identical modules. It sort input by implementing bitonic merge. A Bitonic sorting network of K numbers consists of $\log K$ bitonic sorting stages, where i^{th} stage composed of $N/2^i$ alternating increasing and decreasing bitonic merges of size 2^i [20] as depicted in Fig 3.

4. OMP (OPEN MULTI PROCESSING)

Open MP is an Application Program Interface (API) that supports multi -platform shared memory multiprocessing programming. OMP is an implementation of multithreading. It is a method of parallelizing with fork and join technique. The master thread forks number of threads which execute code blocks in parallel. However, the task is divided among threads by working constructs. The basic component of Open MP is the constructs for thread creation, workload distribution, data environment management, thread synchronization, user-level runtime routines and environment variables.

5. LITERATURE REVIEW

K -way bitonic sort is another generalized version of Batchers' bitonic sort [1]. Hardware algorithms for sorting network of N elements with conflict-free memory accesses are known as batchers classic bitonic sorting network [2]. Optimization in the communication process of merge and split steps is also one of the improved technique for bitonic sorting [3]. Comparison of execution speed of the FPGA processing elements with the microprocessor processing elements, possible with bitonic sort algorithm [4].

Sorting of N^2 elements in row-major order on a $n * n$ mesh connected computer in $O(N)$ time [5]. However, with the combination of the bitonic merge sort algorithm with CCI (Compare and Conditionally Interchange) operations, an effort is made towards the improvement of parallel sorting algorithms for parallel machine [6]. Pleated cube-connected cycles (PCCC) implements optimal stable bitonic sorter in the synchronous VLSI model [7]. Bitonic sorter is the excellent approach for reconfigurable devices which provide enormous speedup [8]. Parallel and sequential bitonic, odd-even and rank-sort algorithms on different GPU and CPU analyzed with various performance metrics for better speed up [9]. However, a comparison between both bitonic sort and radix sort for sorting sequence on GPU and CPU environments for analyzing performance [10]. Interval Based Rearrangement [IBR] bitonic sorting algorithm is optimal when compared with an adaptive bitonic sorting algorithm for sorting sequences [21].

A bitonic sorting algorithm for one to many function were introduced with bit-level comparators, dynamic network and static network [11]. Comparisons among the links in bitonic sorting network made with Fault detection algorithm [12]. Simple wiring interconnection through parity technique raised with Construct-BSNF algorithm [13]. As, parity strategy is used for simplifying the communication from $O(N \log^2 N)$ to $O(N \log N)$ between communication links of the networks with the invention of the RBSN (Recirculation Bitonic Sorting Network). RBSN is used for minimizing communication among processing devices for solving problem [14]. Bitonic sorting algorithm efficiently sort N keys in $O(\log^2 N)$ time with $O(N \log^2 N)$ comparators [15].

M-algorithm proposed VLSI implementation in which bitonic sorter acts as a base [16]. Wafer Scale Integration is a modified version of bitonic sorting algorithm which uses a self reconfigurable algorithm for dynamic fault tolerance [17]. Packed Exponential Connections (PKC) efficiently implement bitonic sorting [18]. However, A-Rank sort was introduced for efficient performance of parallel machine which was compared with bitonic sort algorithm [19].

Increasing number of processors performance of bitonic sort algorithm can be increased. However, Remap-based algorithm is also one approach of optimization in parallel bitonic sort where data re maps of the smallest possible number are considered [20].

6. ESTIMATION OF EXPECTED TIME AND SPEEDUP OF BITONIC SORT ALGORITHM

6.1 Execution Time

Bitonic Sorting Algorithm was implemented in sequential and parallel environment with 2 and 4 threads. Execution time is calculated with the help of start time and end time function of OMP is tabulated in Table 1 for varying number of elements in the power of 2.

Table 1 : Execution Time estimation

Problem Size	Sequential	2 Threads	4 Threads
8	0.00046	0.00024	0.0002
16	0.00095	0.00034	0.00027
32	0.00074	0.0005	0.0004
64	0.00083	0.0006	0.0005
128	0.001	0.0005	0.0005
256	0.00065	0.0005	0.0003
512	0.00072	0.0005	0.0004
1024	0.00042	0.0002	0.0002

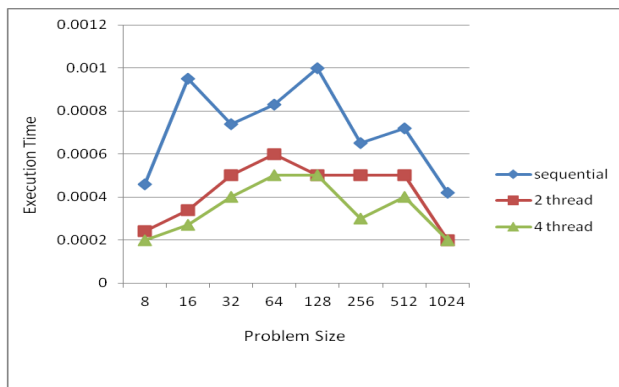


Figure 3 : Line Chart of Execution Time

Above experiments are performed on dual core machine with hyperthreading means each core is assigned 2 threads. It observed from above TABLE 1, execution time for sequential machine is higher compare to 2 threads and 4 threads parallel machines. Sequential execution time is highest for the problem size 128. However, sequential, 2 threads and 4 thread execution time are lowest for the problem size 1024. For the problem size 128 and 1024 the 2 thread machine and 4 thread machine have same execution time. After the problem size 128 the execution time for sequential machine goes on decreasing. However, the machine hangs was experience while running first time problem size 128. For the Table 1 line chart was depicted in Fig 3 for graphical observation.

6.2 Speedup

Speedup is the ratio of Execution time of sequential machine and Execution time of parallel machines. Speedup is also calculated separately for 2 and 4 number of threads and is tabulated in Table 2.

Table 2 : Speedup estimation

Problem Size	Speedup with 2 Threads	Speedup with 4 threads
8	1.9	2.3
16	2.8	3.5
32	1.5	1.9
64	1.4	1.7
128	2	2
256	1.3	1.2
512	1.4	1.8
1024	2.1	2.1

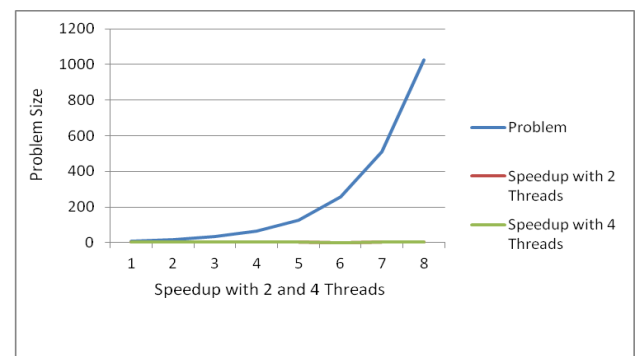


Figure 4 : Line chart of Speedup

TABLE 2 and Fig 4 represents a speedup performance metrics analysis. It is observed that with the increase problem size increases the speedup with 2 threads is less or same in most of cases except for problem size 256. Speedup with 4 threads more or same in maximum cases except for problem size 256. However, speedup with 2 threads and 4 threads for problem size 128 and 1024 remain almost same.

7. CONCLUSION

Parallel computer made up of numbers of processors known as Multiprocessor machine. Core are inbuilt in each processor for processing instruction such parallel computer also known as Multii-Core. The dual core processor and quad core processor with hyperthreading are available. Sorting in such cores machine with hyperthreading computer is achieved by distribution of a set of data. We have analysis the performance of a bitonic sorting algorithm from 2^3 up to 2^{10} . From experiment, it is concluded that parallel machine takes less computation time as compared to sequential machine.

7.1 Future Work

In this paper, as we have analysis the performance of a bitonic sorting algorithm from 2^3 up to 2^{10} .

However, in future further experiments can be made for large elements.

8. ACKNOWLEDGMENTS

This work is supported by School of Studies in Computer Science & I.T, PT. Ravi Shankar Shukla University, Raipur, India.

9. REFERENCES

- [1] Toshio nakatani, Shing-tsaan huang, Bruce W. Arden, and Satish K. Tripath. " K-Way Bitonic Sort ". IEEE Transactions on computers, VOL. 38, NO. 2, FEBRUARY 1989, page no. 283-288, 0018-9340.
- [2] Stephan Olariu, Member, IEEE, M. Cristina Pinotti, Member, IEEE Computer Society, and Si Qing Zheng, Senior Member, IEEE. " An Optimal Hardware-Algorithm for Sorting Using a Fixed-Size Parallel Sorting Device ". IEEE Transactions on computers, VOL. 49, NO. 12, December 2000, page no. 1310-1324, 0018-9340.
- [3] Yong Cheol Kim, Minsoo Jeon, Dongseung Kim Andrew Sohn. " Communication-Efficient Bitonic Sort on a Distributed Memory Parallel Computer* ". Parallel and Distributed Systems, 2001. ICPADS 2001. Proceedings. Eighth International Conference of 2001, page no. 165-170, 0-7695-1 153-8.
- [4] John Harkins, Tarek El-Ghazawi, Esam El-Araby, Miaoqing Huang. " Performance of Sorting Algorithms on the SRC 6 Reconfigurable Computer ". Field-Programmable Technology, 2005. Proceedings. 2005 IEEE International Conference on 11-14 dec. 2005, page no. 295-296, 0-7803-9407-0.
- [5] David Nassimi and Sartaj sahini. " Bitonic Sort on a Mesh-Connected Parallel Computer ".IEEE transactions on computer, VOL. c-27, NO. 1, January 1979, page no. 2-7, 0018-9340.
- [6] Xie Hongwei Xue Yafeng. " An improved parallel sorting algorithm for odd sequence ". 2008 International Conference on Advanced Computer Theory and Engineering. ICACTE 08, on 20-22 Dec. 2008, page no. 356-360, 978-0-7695-3489-3.
- [7] Gianfranco bilardi, Student member, IEEE, and FRANCO P. PREPARATA, FELLOW, IEEE . " An Architecture for Bitonic Sorting with Optimal VLSI Performance". IEEE TRANSACTIONS ON COMPUTERS. VOL c-33, NO. 7, JULY 1984, page no. 646-651.
- [8] J. Angermeier, E. Sibirko, R. Wanka, and J. Teich . " Bitonic Sorting on Dynamically Reconfigurable Architectures* "Parallel & Distributed Processing Workshops and PhD Forum, 2011 IEEE International Symposium on 16-20 May 2011, page no. 314-317, 1530-2075.
- [9] Fiaz Gul Khan, Omar Usman Khan, Bartolomeo Montrucchio, Paolo Giaccone . " Analysis of Fast Parallel Sorting Algorithms for GPU Architectures' ". 2011 Frontiers of Information Technology, on 19-21 Dec, page no. 173-178, 978-1-4673-0209-8.
- [10] Zehra YILDIZ, Musa AYDIN, Güray YILMAZ . " Paralleization of Bitonic sort and radix sort algorithms on many core GPUS ". Electronics, Computer and Computation(ICECCO), 2013 International Conference on 7-9 Nov. 2013, page no. 326-329, 978-1-4799-3343-3.
- [11] Majed 2. Al-Hajery and Kenneth E. Batcher " ON The Role Of K-Bits Bitonic Sorting Network In Multicast Routing* ". Parallel and Distributed Processing, 1994. Proceedings. Sixth IEEE Symposium on 26-29 Oct 1994, on 706-714 page, 0- 8186-6427-4/94.
- [12] Jae-Dong Lee and Kenneth E. Batcher. " A Bitonic Sorting Network with Simpler Flip Interconnections". Parallel Architectures, Algorithms, and Networks, 1996. Proceedings, Second International Symposium on 12-14 Jun 1996, page no. 104-109, 1087-4089.
- [13] Hongin Choi, Kenneth E. Batcher. " Fault Detection in Bitonic Sorting Networks". Parallel and Distributed Processing. 1995. Proceedings. seventh IEEE Symposium on 25-18 Oct 1995, page no. 266-270, 1063-6374.
- [14] Jae-dong Lee and Kenneth E. Batcher. " Minimizing Communication of a Recirculating Bitonic Sorting Network ". Parallel Processing, 1996. Vol 3. Software, In the Proceeding 1 996 International Conference on Parallel Processing. At 12-16 Aug 1996, page no. 251-254 Vol 1, 0190-3918.
- [15] Jae-Dong Lee and Kenneth E. Batcher. " Minimizing Communication in the Bitonic Sort ". Parallel and Distributed Systems, IEEE Transactions on, Vol. 11, NO. 5, May 2000, page no. 459-474, 1045-9219.
- [16] Stanley J. Simmons." A Bitonic-Sorter based VLSI implementation of the M-Algorithm". IEEE Pacific Rim Conference on Communications, Computers and Signal Processing June 1st - 2nd, 1989, page no. 337-340.
- [17] Susumu Horiguchi Issei Numata Masayuki Kimura. " Self-Reconfigurable Algorithm of WSI Sorting Network ".In the Proceeding I991, Third International Conference on Wafer Scale Integration, on 29-31 Jan 1991, page no. 249-255.0-8186-9126-3.
- [18] Donna Quammen and Pearl Y. Wang " Bitonic Sorting on 2D-PEC: An Algorithmic Study on a Hierarchy of Meshes Network ". Parallel Processing Symposium, 1994 . Proceedings., Eighth International, on 26-19 Apr 1994, page no. 418-423, 0-8186-5602-6.
- [19] Taoufik Dachraou, Lata Narayanan *" Fast Deterministic Sorting on Large Parallel Machines ". Parallel and Distributed Processing, 1996., Eighth IEEE Symposium on 23-26 Oct 1996, page no. 273-280, 0-8186-7683-3.
- [20] Mihai Florin Ionescu and Klaus E. Schauer . " Optimizing Parallel Bitonic Sort ". Parallel Processing Symposium, 1997 . Proceedings., Eleventh International in 1-5 Apr, page no. 303-309, 1063-7133,0-8186-7793-7.
- [21] Hagen Peters, Ole Schulz-Hildebrandt, Norbert Luttenberge." A novel sorting algorithm for many-core architectures based on adaptive bitonic sort ". 2012 IEEE 26th International Parallel and Distributed Processing Symposium, page no. 227-237.
- [22] Megha Jain, Sanjay Kumar, V.K.Patle , " Bitonic Sorting Algorithm", IJCA, Volume 113-No.13, March 2015.
- [23] Sorting Networks. <https://mitpress.mit.edu/sites/default/files/Chapter%2027.pdf>.