Parallel Implementation of Bitonic Sort

Presented For CSE633

Instructor: Dr. Russ Miller

Presented By:

Anushree Parmar

University at Buffalo The State University of New York

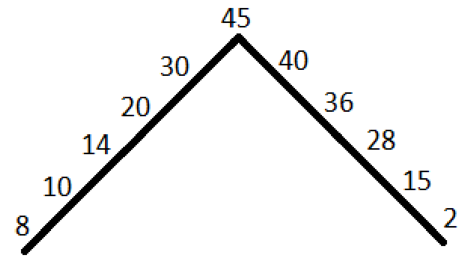
Why Bitonic Sort?

- No of comparisons in Bitonic sort are O(n Log ^{2}n)
- No of comparisons done by most of the algorithms like Merge Sort or Quick Sort take O(n Logn)
- Bitonic sort is better for parallel implementation

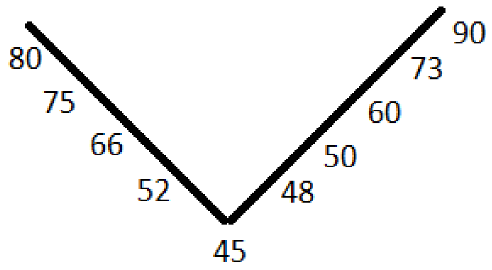
Bitonic Sequence

A sequence numbers is said to be *bitonic* if and only if

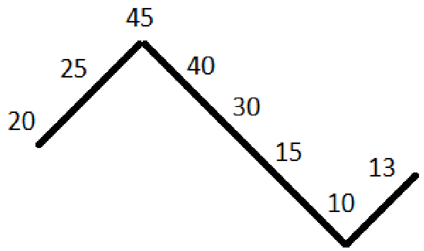
1. Monotonically increases and then monotonically decreases



2. Monotonically decreases and then monotonically increases

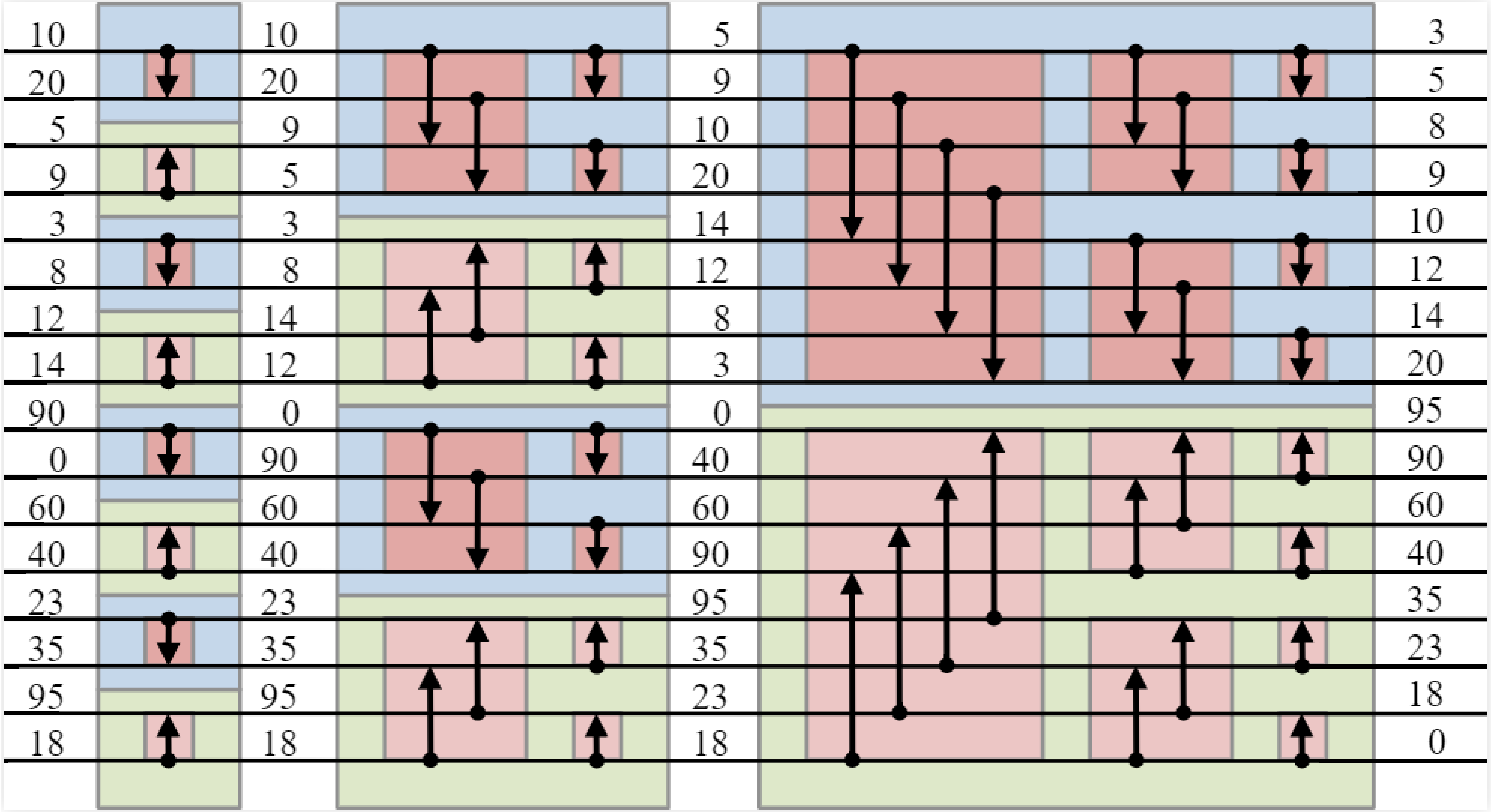


3. Can be split into two parts that can be interchanged to give either of the first two cases.

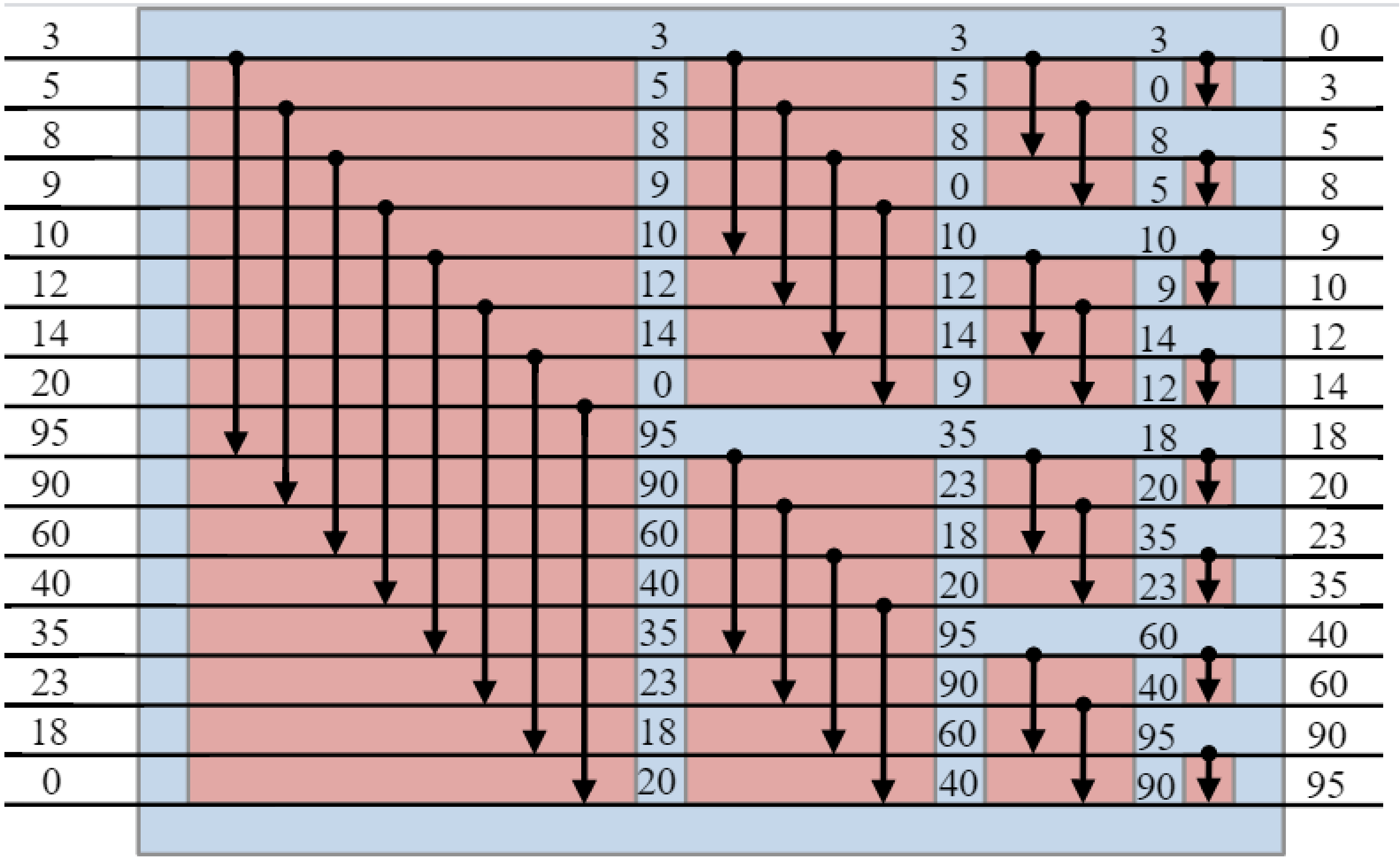


Rearrange to a bitonic sequence

<table>
<tr><td>$\oplus$ BM[2]</td><td rowspan="2">$\oplus$ BM[4]</td><td rowspan="4">$\oplus$ BM[8]</td><td rowspan="8">$\oplus$ BM[16]</td></tr>
<tr><td>$\ominus$ BM[2]</td></tr>
<tr><td>$\oplus$ BM[2]</td><td rowspan="2">$\ominus$ BM[4]</td></tr>
<tr><td>$\ominus$ BM[2]</td></tr>
<tr><td>$\oplus$ BM[2]</td><td rowspan="2">$\oplus$ BM[4]</td><td rowspan="4">$\ominus$ BM[8]</td></tr>
<tr><td>$\ominus$ BM[2]</td></tr>
<tr><td>$\oplus$ BM[2]</td><td rowspan="2">$\ominus$ BM[4]</td></tr>
<tr><td>$\ominus$ BM[2]</td></tr>
</table>



No of comparison levels 1 2 3



No of comparison levels

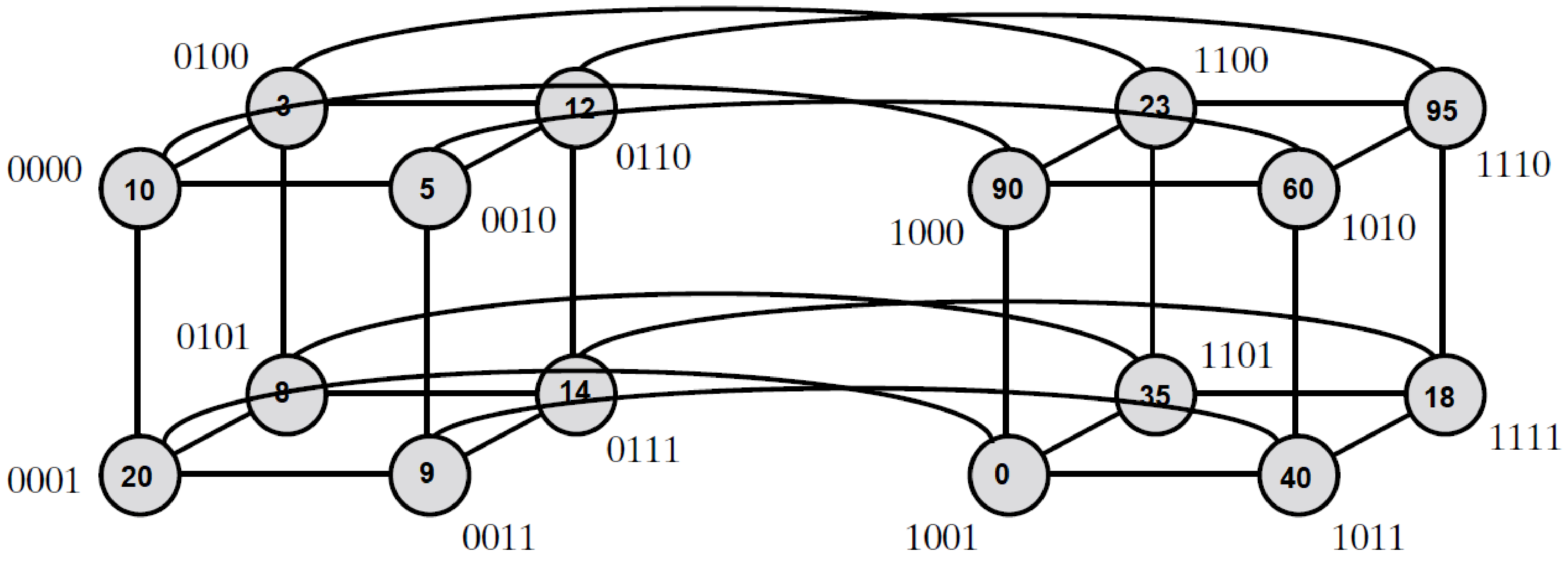
4

Algorithm

```
BitonicSort(a, low,high,direiction):
    if high> 1:
        k = high/2
        BitonicSort(a, low, k, 1)
        BitonicSort(a, low+k, k, 0)
        BitonicMerge(a, low, high, direction)
```

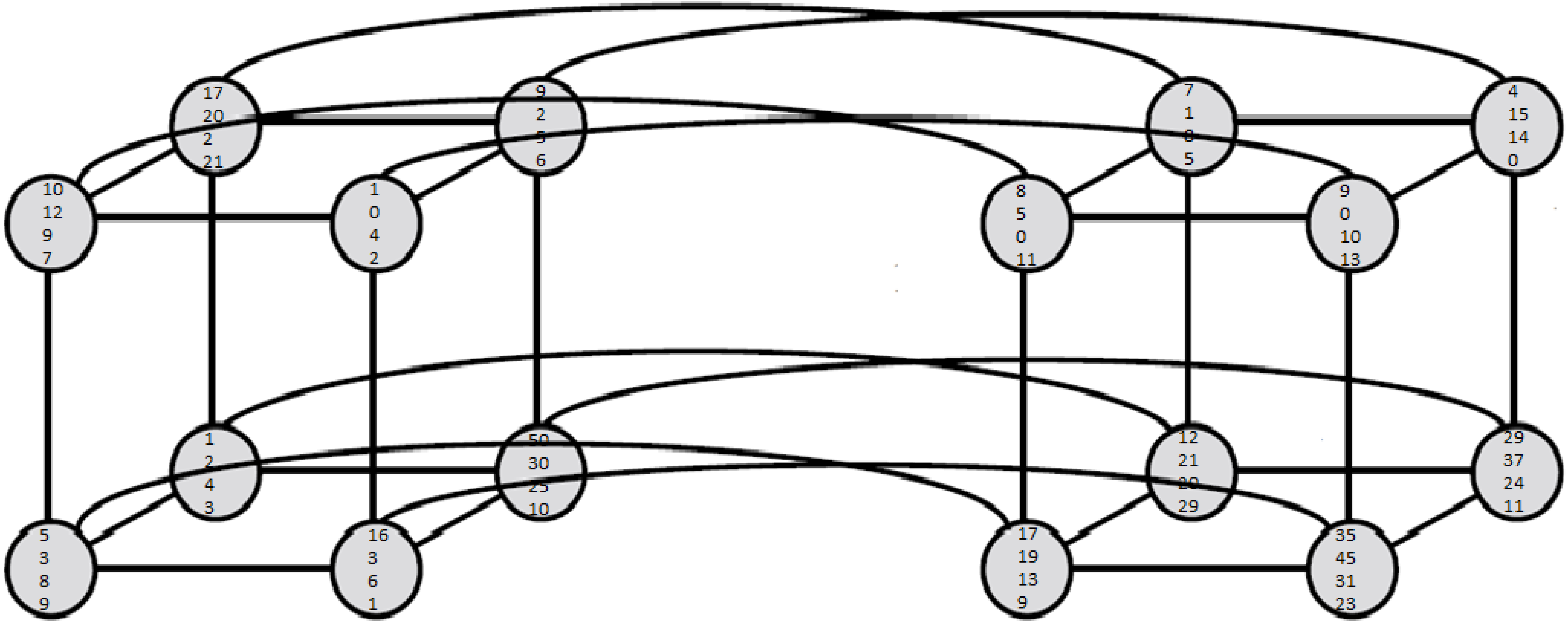
```
BitonicMerge(a, low,high, direction):
    if high > 1:
        k = high/2
        for i in range(low , low+k):
 // Based on direction swap the data
              a[i],a[i+k] = a[i+k],a[i]
              BitonicMerge(a, low, k, direction)
              BitonicMerge(a, low+k, k, direction)
```

Parallel Execution



Parallel Algorithm Implementation

- Generate the data randomly
- N – Amount of data in each processor
- n – No. of processors
- Sequentially sort data in each processor using sorting algorithms like Merge Sort
- Compare the sorted sequences from each processor the way compared in Bitonic Sort
- Recursively repeat the same process
- Time Complexity - $O(N \log N) + O(N \log^2 n)$

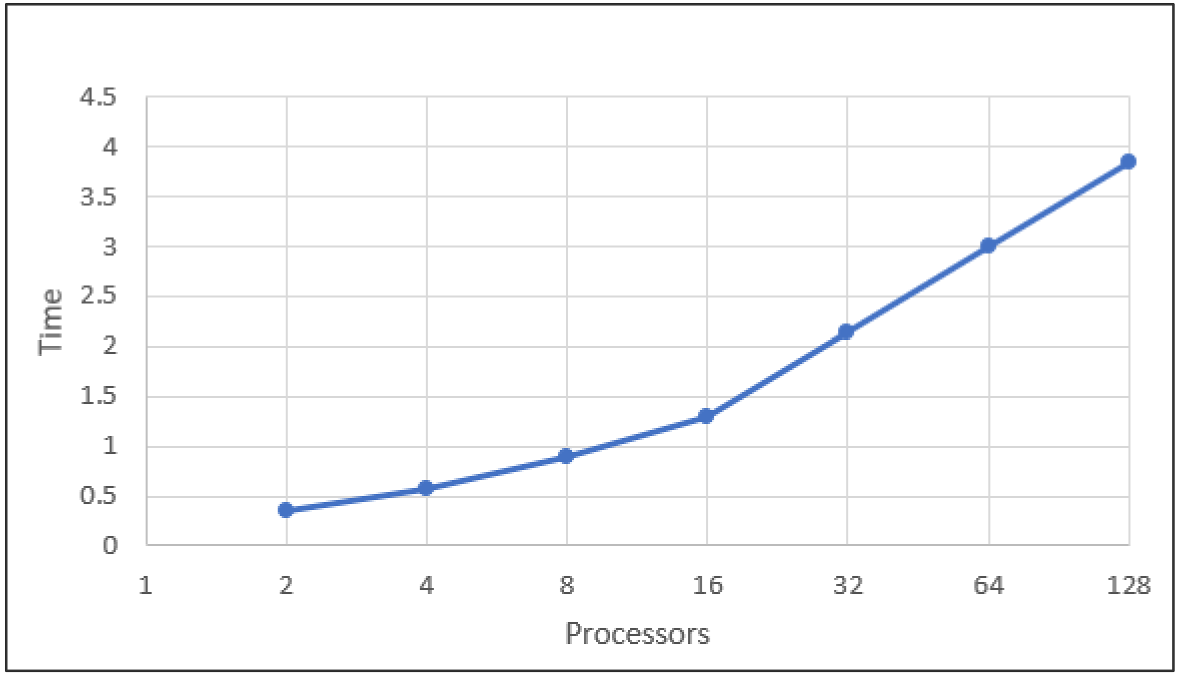

17 20 2 21
9 2 5 6
7 1 5
4 15 14 0
10 12 9 7
1 0 4 2
8 5 0 11
9 0 10 13
1 2 4 3
50 30 25 10
12 21 29
29 37 24 11
5 3 8 9
16 3 6 1
17 19 13 9
35 45 31 23

Results

1 Million Data per processor

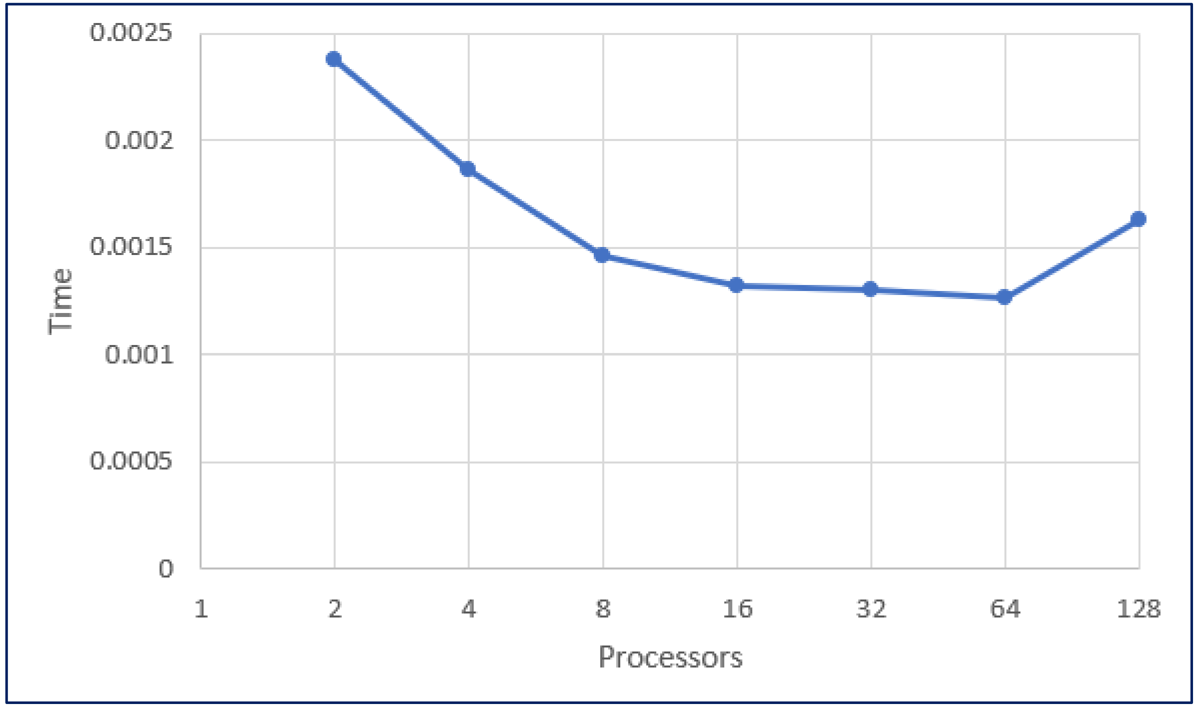
Time Complexity - O(N Log N) + O(N Log ^{2}n)

No. of Processors	Data	Time
2	2000000	0.351248
4	4000000	0.573388
8	8000000	0.888249
16	16000000	1.299627
32	32000000	2.144949
64	64000000	2.991943
128	128000000	3.853609



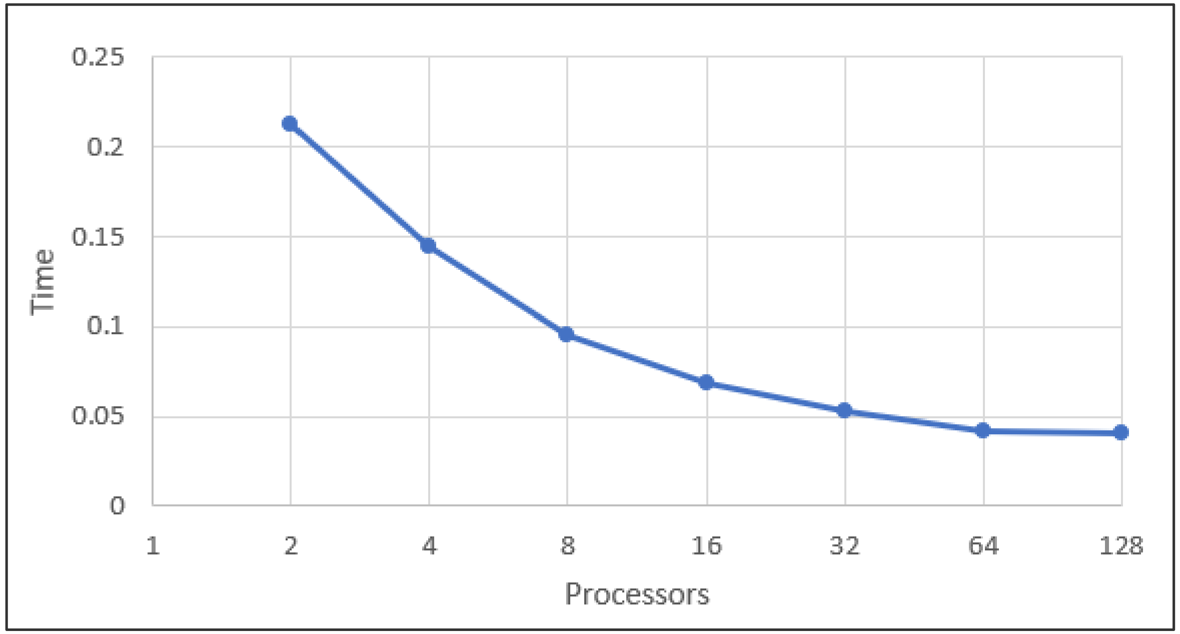
Constant Data Size – 10,000 Data

No. of Processors	Time(s)
2	0.002375
4	0.001864
8	0.001466
16	0.001323
32	0.001307
64	0.001269
128	0.001633



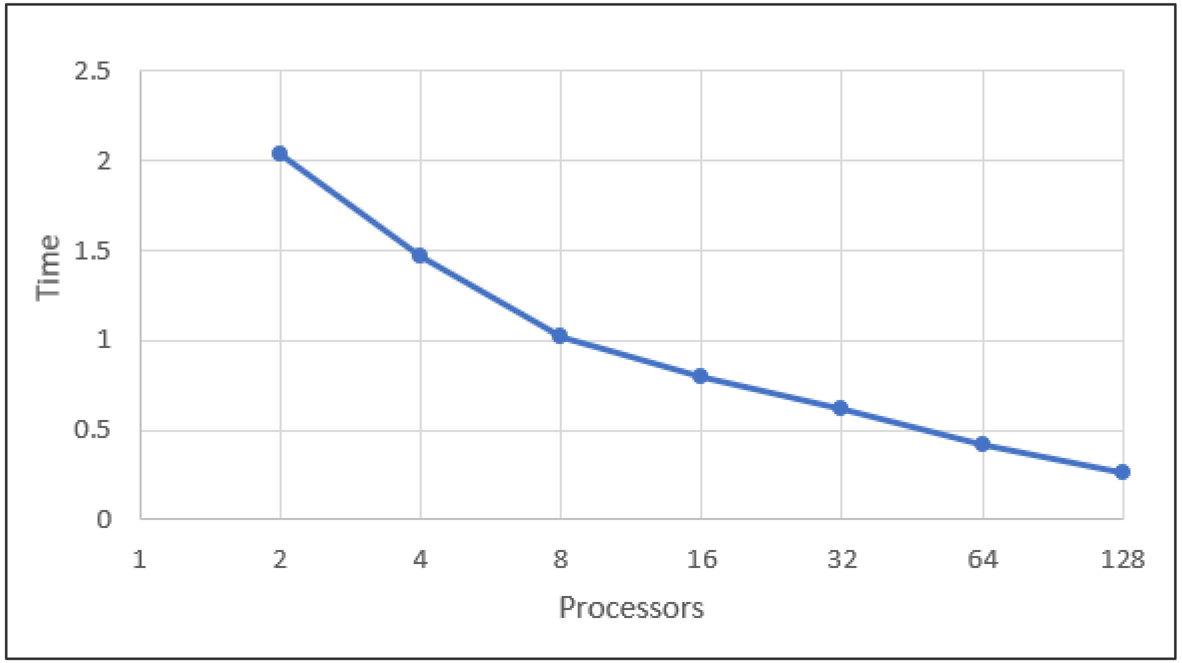
Constant Data Size – 1 Million Data

No. of Processors	Time(s)
2	0.212092
4	0.1446
8	0.095244
16	0.069043
32	0.053
64	0.041478
128	0.040321



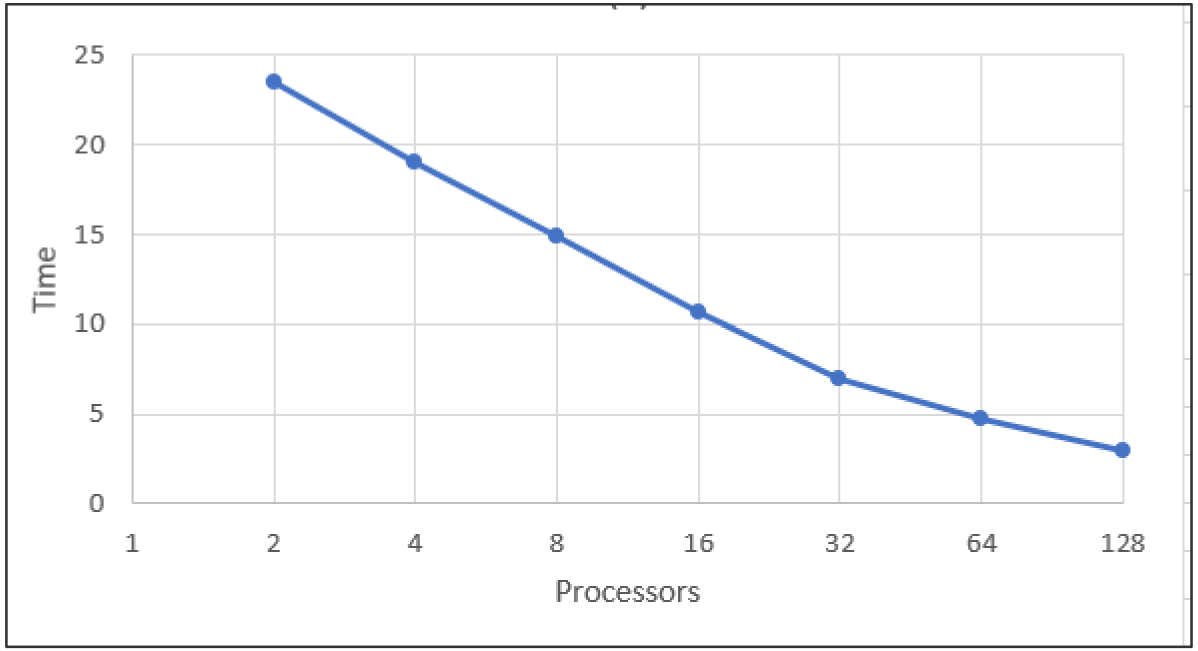
Constant Data Size – 10 Million Data

No. of Processors	Time(s)
2	2.029992
4	1.468713
8	1.024255
16	0.80198
32	0.621896
64	0.418867
128	0.262654



Constant Data Size – 100 Million Data

No. of Processors	Time(s)
2	23.49861
4	18.99897
8	14.87071
16	10.65893
32	6.959202
64	4.742998
128	2.910114



Observations

- When data is kept constant per processor, time increases with a factor of Log ^{2}n.
- Increasing the number of processors, increases the communication overheard which outweighs the benefit of reducing computation per processor.
- For input size used, using more than 16 or 32 processors is not practical.

References

- Algorithms Sequential and Parallel: A Unified Approach by Russ Miller and Laurence Boxer
- http://www.cs.utah.edu/~hari/teaching/paralg/slides/lec06.html#/3/13
- https://www.geeksforgeeks.org/bitonic-sort/
- https://en.wikipedia.org/wiki/Bitonic_sorter
- https://ubccr.freshdesk.com/support/solutions/articles/13000026245-tutorials-and-training-documents

Thank You.