## List of Tables

2.1 Execution time in milliseconds of sequential count sort .......................... 30

2.2 Execution time in milliseconds of parallel count sort using uniform
test case ........................................................................................................... 31

2.3 Execution time in milliseconds of parallel count sort using sorted
test case ........................................................................................................... 32

2.4 Execution time in milliseconds of parallel count sort using zero test
case .................................................................................................................. 33

2.5 Execution time in milliseconds of parallel count sort using bucket
test case ........................................................................................................... 34

2.6 Execution time in milliseconds of parallel count sort using Gaussian test case .......................................................... 35

2.7 Execution time in milliseconds of parallel count sort using staggered test case .......................................................... 36

2.8 Speedup achieved by parallel count sort using different types of
test cases with $n=7500000$ ............................................................................... 37

2.9 Speedup achieved by parallel count sort using different types of
test cases with $n=10000000$ ............................................................................ 38

2.10 Speedup achieved by parallel count sort using different types of
test cases with $n=5000000$ ............................................................................... 39
2.11 Speedup achieved by parallel count sort using different types of test cases with \( n = 2500000 \) .................................................. 40

2.12 Speedup achieved by parallel count sort using different types of test cases with \( n = 1000000 \) .................................................. 40

2.13 Sequential and parallel execution time in seconds of merge and quick sort using the four cases of the dataset. ................................................. 44

2.14 Sequential and parallel memory in bytes of merge sort using the random dataset. .......................................................... 46

2.15 Sequential and parallel memory in bytes of quick sort ............... 47

2.16 Execution Time of various sorting algorithm in seconds ........... 48

2.17 Total memory occupied by various sorting algorithms using dataset 55

3.1 Execution time in sec of sequential OETSN using different types of test cases .......................................................... 64

3.2 Execution time in sec of parallel OETSN using different types of test cases .......................................................... 65

3.3 Speedup achieved by parallel OETSN using different types of test cases .......................................................... 66

3.4 Execution time in seconds of modified parallel OETSN using different types of test cases .......................................................... 72

4.1 Execution Time of Library Sort Algorithm in Microseconds Based on Gap Values .......................................................... 81

4.2 Total Memory in Bytes of Library Sort with Increasing Value of Gaps and Re-balancing Factor .......................................................... 85

4.3 Time taken by Library Sort Algorithm in Microseconds during Re-balancing .......................................................... 86

xv
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Execution Time of Library Sort Algorithm in Microseconds Based on Gap Values</td>
<td>94</td>
</tr>
<tr>
<td>5.2</td>
<td>Execution Time of Library Sort Algorithm in Microseconds Based on Gap Values</td>
<td>100</td>
</tr>
<tr>
<td>6.1</td>
<td>Execution Time in Microseconds of Bucket with Merge Sort</td>
<td>110</td>
</tr>
<tr>
<td>6.2</td>
<td>Execution Time in Microseconds of Bucket with Count Sort</td>
<td>110</td>
</tr>
<tr>
<td>6.3</td>
<td>Execution Time in Microseconds of Proposed Hybrid Approach</td>
<td>111</td>
</tr>
<tr>
<td>6.4</td>
<td>Auxiliary Memory Occupied by Bucket with Merge Sort in Bytes</td>
<td>115</td>
</tr>
<tr>
<td>6.5</td>
<td>Auxiliary Memory Occupied by Bucket with Count Sort in Bytes</td>
<td>116</td>
</tr>
<tr>
<td>6.6</td>
<td>Auxiliary Memory Occupied by Proposed Hybrid Sort in Bytes</td>
<td>117</td>
</tr>
<tr>
<td>7.1</td>
<td>Summary of the various articles</td>
<td>128</td>
</tr>
<tr>
<td>7.2</td>
<td>Execution time in seconds of sequential bubble sort</td>
<td>129</td>
</tr>
<tr>
<td>7.3</td>
<td>Execution time in seconds of parallel bubble sort</td>
<td>130</td>
</tr>
<tr>
<td>7.4</td>
<td>Speedup achieved by parallel bubble sort at $T=512$</td>
<td>130</td>
</tr>
</tbody>
</table>
List of Figures

1.2 Min-heap .......................................................... 8
1.1 Max-heap .......................................................... 8
1.3 Example of shell sort ............................................. 10
1.4 Example of shell sort ............................................. 11
1.5 Example of shell sort ............................................. 11
1.6 Split of $n$ on two consecutive levels ....................... 15
1.7 Split of $n$ on two consecutive levels ....................... 15
1.8 Example of Radix Sort .......................................... 17
1.9 Example of Radix Sort .......................................... 18
1.10 Example of Radix Sort ........................................ 18
1.11 Example of Radix Sort ........................................ 19
1.12 Example of Radix Sort ........................................ 19
1.13 Example of Bucket Sort ...................................... 21
1.14 CUDA support the various languages ....................... 23
1.15 GPU Architecture .............................................. 24
1.16 Scalable Programming Model ................................. 25
2.1 Execution time comparison between parallel and sequential count sort using uniform test case ........................................ 32
2.2 Execution time comparison between parallel and sequential count sort using sorted test case ........................................ 33
2.3 Execution time comparison between parallel and sequential count sort using zero test case ........................................ 34
2.4 Execution time comparison between parallel and sequential count sort using bucket test case ........................................ 35
2.5 Execution time comparison between parallel and sequential count sort using gaussian test case ................................... 36
2.6 Execution time comparison between parallel and sequential count sort using staggered test case ................................... 37
2.7 Speedup achieved by parallel count sort using different types of test cases with \( n=7500000 \) .................................................. 38
2.8 Speedup achieved by parallel count sort using different types of test cases with \( n=10000000 \) .................................................. 39
2.9 Speedup achieved by parallel count sort using different types of test cases with \( n=5000000 \) .................................................. 39
2.10 Speedup achieved by parallel count sort using different types of test cases with \( n=2500000 \) .................................................. 40
2.11 Speedup achieved by parallel count sort using different types of test cases with \( n=1000000 \) .................................................. 41
2.12 Example of parallel merge sort ........................................ 42
2.13 Execution time comparison between sequential and parallel merge sort ................................................................. 45
2.14 Execution time comparison between sequential and parallel quick sort .................................. 45
2.15 Memory comparison between sequential and parallel merge sort . 46
2.16 Memory comparison between sequential and parallel quick sort . 47
2.17 Execution time of insertion sort ......................................................... 49
2.18 Execution time of selection sort ..................................................... 49
2.19 Execution time of bubble sort ....................................................... 50
2.20 Execution time of heap sort ............................................................ 50
2.21 Execution time of shell sort ............................................................ 51
2.22 Execution time of count sort .......................................................... 51
2.23 Execution time of quick sort .......................................................... 52
2.24 Execution time of merge sort ......................................................... 52
2.25 Execution time of radix sort .......................................................... 53
2.26 Memory occupied by various sorting algorithms .......................... 55

3.1 (a) Increasing Comparator (b) Decreasing Comparator. .............. 58
3.2 Example of OETS network ............................................................. 58
3.3 Flowchart for the proposed modified parallel OETSN ................. 62
3.4 Speedup achieved by parallel OETSN using uniform test case . . . 67
3.5 Speedup achieved by parallel OETSN using Gaussian test case . . 67
3.6 Speedup achieved by parallel OETSN using zero test case ............ 68
3.7 Speedup achieved by parallel OETSN using staggered test case . . 68
3.8 Speedup achieved by parallel OETSN using bucket test case . . . . 69

xix
3.9 Speedup achieved by parallel OETSN using sorted test case

4.1 Execution time of random data using value of gaps

4.2 Execution time of nearly sorted data using value of gaps

4.3 Execution time of reverse sorted data using value of gaps

4.4 Execution time of sorted data using value of gaps

4.5 Memory occupied by library sort

4.6 Re-balancing of library sort using random dataset

4.7 Re-balancing of library sort using reverse sorted dataset

4.8 Re-balancing of library sort using sorted dataset

4.9 Re-balancing of library sort using nearly sorted dataset

5.1 Execution time comparison between LUGD and LNGD using random data

5.2 Execution time comparison between LUGD and LNGD using nearly sorted data

5.3 Execution time comparison between LUGD and LNGD using reverse sorted data

5.4 Execution time comparison between LUGD and LNGD using sorted data

5.5 Re-balancing execution time comparison between LUGD and LNGD using random data

5.6 Re-balancing execution time comparison between LUGD and LNGD using nearly sorted data

5.7 Re-balancing execution time comparison between LUGD and LNGD using reverse sorted data
5.8 Re-balancing execution time comparison between LUGD and LNGD using sorted data .................................................. 102

6.1 Example of Bucket Sort ............................................. 105

6.2 Distribution of buckets .............................................. 106

6.3 Flowchart of Proposed Hybrid Sort .............................. 108

6.4 Execution time comparison of uniform test case ............... 111

6.5 Execution time comparison of bucket test case ............... 112

6.6 Execution time comparison of gaussian test case ............ 113

6.7 Execution time comparison of sorted test case ............... 113

6.8 Execution time comparison of staggered test case .......... 114

6.9 Execution time comparison of zero test case ................. 114

6.10 Memory comparison of uniform test case ...................... 118

6.11 Memory comparison of bucket test case ....................... 118

6.12 Memory comparison of gaussian test case .................... 119

6.13 Memory comparison of sorted test case ....................... 120

6.14 Memory comparison of staggered test case ................... 120

7.1 Speedup achieved by parallel bubble sort using uniform test case 131

7.2 Speedup achieved by parallel bubble sort using gaussian test case 131

7.3 Speedup achieved by parallel bubble sort using zero test case 132

7.4 Speedup achieved by parallel bubble sort using staggered test case 132

7.5 Speedup achieved by parallel bubble sort using bucket test case 133

7.6 Speedup achieved by parallel bubble sort using sorted test case 133

xxi
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.7</td>
<td>Execution time comparison of parallel and sequential bubble sort using uniform test case</td>
<td>134</td>
</tr>
<tr>
<td>7.8</td>
<td>Execution time comparison of parallel and sequential bubble sort using gaussian test case</td>
<td>135</td>
</tr>
<tr>
<td>7.9</td>
<td>Execution time comparison of parallel and sequential bubble sort using zero test case</td>
<td>135</td>
</tr>
<tr>
<td>7.10</td>
<td>Execution time comparison of parallel and sequential bubble sort using staggered test case</td>
<td>136</td>
</tr>
<tr>
<td>7.11</td>
<td>Execution time comparison of parallel and sequential bubble sort using bucket test case</td>
<td>136</td>
</tr>
<tr>
<td>7.12</td>
<td>Execution time comparison of parallel and sequential bubble sort using sorted test case</td>
<td>137</td>
</tr>
</tbody>
</table>