Chapter 8

Conclusions and Future Scope

8.1 Conclusion

In this thesis, the following algorithms have been tested using sorting benchmark and standard dataset with GPU computing.

1. GPU Merge Sort using CUDA hardware.
2. GPU Quick Sort using CUDA hardware.
3. GPU Count Sort using CUDA hardware.
4. GPU Bubble Sort using CUDA hardware.

In this thesis, we have also tested the various sorting algorithms on a standard dataset. The various algorithms are following.

1. Insertion Sort
2. Selection Sort
3. Bubble Sort
4. Heap Sort
5. Shell Sort
6. Quick Sort
7. Merge Sort
8. Radix Sort
9. Count Sort

The Following algorithms have been proposed.
1. Proposed Modified parallel OETSN algorithm.
2. Library sort algorithm with uniform gap distribution.
3. Library sort algorithm with non-uniform gap distribution.
4. Proposed Hybrid Sort Algorithm.

The performance measures have been done of all the listed algorithms in terms of space and time complexity. In the future, the performance measures can be tested of all the listed algorithms in terms of stability and adaptivity.

8.2 Future Scope

We can further classify other sorting algorithm like quick sort based on the number of elements in the bucket which will not only make the working faster of bucket sort but also will reduce the time.

We can still find a gap to use the knowledge about the data to implement the sorting algorithm. Future research will refine the performance of sorting algorithms using GPU architecture and Thrust library.

The parallel version of library sort using CUDA hardware can be designed in future. The GPU LNGD (Library sort using non-uniform gap distribution) can also be designed.

We have used the GPU computing using CUDA hardware having the compute capability 2.1 to test the algorithms. But, if the same algorithms has been used on the hardware having the compute capability 3.0, then it will give an added advantage of unified memory architecture. The performance of GPU algorithms can be enhance by using different CUDA hardware versions and using Thrust Library.