Chapter 8

Conclusion and Future Work

In this chapter, we have a propensity to gift the conclusions from this thesis and few recommendations and future works are also discussed. In section 8.1, we have presented the approaches and a conclusion for the work. Some of the long term works are planned in section 8.2.

8.1 CONCLUSION

The main aim of the proposed thesis is to analyse the possibilities of vectorizing the applications with limited resources. "***Code once and use it for multiple times***" is the vision of the complete application development process. At first, analysis is completed in order to know various performance challenges in parallel programming models. With the computed results, we have applied some of the merits of OpenCL execution trace in MATlAB code to perform in a better way. Vectorized search and optimization is applied in patient monitoring system and execution time is reduced with a speed up of 2X.

Environments like IoT applications require very quick response but lacks in processing. Systems acting as resource base for IoT based application should have good processing capabilities. The implementation of Vectorized AES algorithm shows maximum of 40X speed up compared with the sequential running time. Master slave based execution for CPU and GPU results are better than other techniques.

Applying multiple levels of Vectorization will work in a better manner. Comparative analysis of image compression algorithms against the sequential execution is tested and the results shows that this type of vectorization will have the same effort while scaling the processing power and henceforth execution time will also improve.

As a result of the analysis increasing partitioning and agglomeration and reducing the communications in vectorization process will result in good speedup.

Table 8.1 Summary of research

Vectorization method used in research	**Research gap exist in previous work?**	**Performance improvement after vectorization**	**Test applications used**	**Parallel computing tools used**
Problem1 in chapter 3: Direct software/ compiler based	No	Good (10X-20X)	1. Large size matrix multiplication 2. Query processing system for online shopping	OPenMp, CUDA, Matlab, OpenCL MPI
Problem2 in chapter 4: Divide and conquer-Execution trace based code analysis	Yes	Average (2X-9.9X)	Patient monitoring system with IoT	OpenCL .NET
Problem3 in chapter 5: Design challenges for vectorization with Matlab implementation	Yes	Good (10X to 20X)	Currency exchange prediction system	Matlab
Problem4 in chapter 6: Multi-level vectorization by utilizing CPU and GPU with execution trace	Yes	Good (10X-20X)	Image compression	OpenCL Matlab
Problem5 in chapter 7: Master/slave based vectorization	Yes	Very good <25X	AES encryption	Matlab toolkit

8.2 FUTURE WORK

There are three areas we need to concentrate in future vectorizations. One is with respect to amdhal's law, i.e. processors are getting updated frequently. It is the role of programmers to understand the processing before coding. Vectorization methods can be implemented with the programming constructs.

Second, multi core processing can be implemented directly with IoT processing so that decision making can be performed at the device itself.

Finally, testing the parallel algorithms in customized processing environment will give the exact output. So the proposed framework can be tested with customized environments.

This research work concentrated only on execution time for performance improvement. Impact of other factors like memory, cost etc. can be studied as future work.