CHAPTER 8

CONCLUSION AND FUTURE WORK

This chapter describes the various algorithms implemented in this research and indicates future directions that can be carried out from this research work.

8.1 CONCLUSION

In grid environment, many researchers have proposed various scheduling algorithms for improving the performance of the grid system. This research work introduces various scheduling algorithms which are more suitable for grid environment. This thesis began by studying and understanding several aspects of grid computing. In literature survey, various algorithms and methods have been identified and studied. Even though many researchers proposed various scheduling algorithms, it is found that there is no efficient and effective scheduling algorithm that gives a combined solution for many issues. This research proposes efficient and effective scheduling algorithms in which various issues such as user satisfaction, load balancing, fault tolerance are addressed by considering the parameters such as user deadline, makespan, resource utilization, failure rate and load state of the resources for scheduling.

First, prioritized user demand algorithm is designed which mainly concentrates on user deadline, expected execution time and makespan of the
jobs. The PUD algorithm is better with reduced makespan and increased hit rate than the other existing algorithms.

Secondly, user demand aware scheduling algorithm for data intensive tasks is proposed in which data transfer time is included while calculating expected completion time. Since the UDDA algorithm considers the data transfer time, the performance of the scheduler is improved. To reduce the data transfer time of jobs, grouping based user demand aware algorithm is implemented and tested. This work reduces the communication overhead in grid scheduling by grouping jobs and by considering user satisfaction.

Then, a hierarchical load balancing model with user demand aware grid scheduling algorithm is implemented which considers load of each resource and performs load balancing. In this work, load is shared effectively among all the resources and this minimizes the response time of the jobs and improves the utilization of resources in the grid environment. This algorithm considers the dynamicity of the resources/machines/ PEs, resource/machine heterogeneity, and task heterogeneity. By considering the user deadline of the job, the number of jobs completed within user deadline (hit rate) is increased and user satisfaction is improved.

Finally, a fault tolerant load balancing scheduling algorithm is introduced. It considers failure rate of the resources to reduce job failures and makespan. This algorithm also considers load of each resource while scheduling the jobs. Since it considers load and fault rate of the resources, it has minimized makespan, improved hit rate, reduced miss rate and better resource utilization.

Performance of the various proposed algorithms is compared with existing methods and the experimental results show that the proposed
algorithms such as PUD, UDDA, GUDA, HLBA and FTLB algorithms have better results over the other respectively.

8.2 FUTURE WORK

In future, some other user requirements such as cost for execution may be considered. In addition to that, other passive failure handling mechanisms such as check pointing may be considered. The jobs may arrive in a random manner. So, the dynamicity of the jobs may be considered for testing proposed algorithms. The proposed algorithms are tested with 16 resources and varied number of jobs up to 600. In future, the number of resources and jobs may be increased and tested as an extension of the proposed algorithms.

Future work can also explore modeling other characteristics such as I/O behavior, memory access pattern, cache effects and seek to build corresponding scheduling strategies that utilize these parameters to form efficient scheduling strategies.