CHAPTER 7

CONCLUSIONS AND FUTURE ENHANCEMENT

7.1 CONCLUSIONS

Efficient application scheduling is critical for achieving high performance in a homogeneous cluster computing environment. Because of its prime importance, the task scheduling problem has been extensively studied and various heuristics have been proposed. This thesis starts with the implementation of the list scheduling heuristic which is scalable and provides low complexity O ((e) (p + log v)), which is less when compared to the other list scheduling algorithms reported. The simulation results confirm that SLTS is substantially better.

A hybrid strategy that combines a metaheuristic algorithm with a local heuristic leads to a better result compared to the GA. A hybrid approach that combines the excellence of both PSO and TS is proposed. The experimental results show that the proposed hybrid approach proves to be more effective in solving the tasks scheduling problem. The average makespan of the proposed hybrid metaheuristic approach is 36.66 while the average makespan of the GA is 38.04 and the PSO is 37.48 for different CCRs.

Most of the scheduling heuristics did not consider the contention for communication resources. Contention awareness is achieved through link contention by using the appropriate topology graph. The concepts of Tabu search and Simulated Annealing are embedded into the classical list scheduling algorithm through the edge scheduling concepts. The experimental results show that the proposed approach BSA-TS is 44% (CCR=0.1), 77% (CCR=1) and 88% (CCR=10) superior to the DLS in terms of the schedule length due to a more appropriate choice of node priority. The quality of the schedule generated by any algorithm depends upon the communication cost and the schedule. The communication cost of BSA-TS is almost 50% less than the DLS and BSA algorithms and the BSA-SA is 18% while the schedule length produced by the proposed BSA-TS approach is 3 times better than the classical list scheduling heuristic.

Memetic Algorithms (MA) constitutes an extremely powerful tool for handling combinatorial optimization problems. By hybridizing the population based evolutionary searching ability of the GA with the local improvement abilities of Hill Climbing (HC) and Tabu Search (TS) to balance exploration and exploitation is then investigated. The proposed approach MA-TS is compared with the MA-LS and MA-SA as well as with the GA. The improvement in schedule length produced by the MA-TS is 1.3% when compared to the MA-LS, 3.85% when compared to the proposed SLTS, 6.25% when compared to the CPOP and 11% when compared to the EFT and the same as that of the MA-SA approach. Experimental results show that the proposed approach gives a better solution compared to the other two because TS avoids premature convergence compared to HC.

To deal with the ever increasing data volumes especially for handling Web documents and request logs, a high level abstraction is required for such data-intensive computing. The MapReduce programming model is the suitable solution for tackling such situations. Also, to handle massive data parallelism with the available PC Clusters, a MapReduce framework has been developed by the Google Engineers. The performance of this model purely depends upon the task scheduler. A Resource Aware Scheduler which is flexible enough to react to the dynamic changes of the CPU load and the data locality of tasks in the nodes of the cluster is developed with the support for preemption. It is the distinct work of this thesis. The execution time of the MapShuffle Reduce (MSR) Framework is 26.72% when compared to the MapReduce (MR) Framework for running 4 jobs in 4 nodes of the cluster for word count problem dataset of 20 MB.

Experimental results show that the proposed metaheuristic approaches give better solutions compared to the existing heuristics in terms of makespan, efficiency, speedup and scalability. The developed Resource Aware Scheduler considers preemptive tasks with the awareness of the computing resources available for execution using profile information.

7.2 FUTURE ENHANCEMENTS

The future work of this research will be to incorporate the processor involvement into the scheduling models and to explore support for both data and task level parallelisms. Configuration of data nodes for the Resource Aware Scheduler can be automated by allowing the system to find the availability of nodes in run time. Timers can be used for reliable message transfer between the Job Tracker and the Task Tracker. The Resource Aware Scheduler can still be enhanced to support multiple jobs at a time.