Chapter - 6
CHAPTER 6

CONCLUSIONS AND CONTRIBUTIONS

6.1 Conclusion

The major findings and contributions of this research work and the potential directions for further research works are summarized in this chapter.

This Thesis work primarily focuses on scheduling the jobs in an optimize manner. The user submits the numbers of jobs, which in this thesis, are created in a simulated environment. These jobs are arranged in different combinations of scheduling algorithms. Once the combinations are applied, the best combination for the given job is determined and is submitted to the middleware for computation.

The following graphs will help to conclude the thesis work very clearly. There are two type of graphs mentioned here. One is for comparison among different scheduling algorithms (First Come First Serve, Shortest Job First, and Round Robin) and another is for combination of these scheduling algorithms.
Figure 6.1 Waiting Time Graph for FCFS, SJF, and RR

Figure 6.2 Waiting Time Graph for six Combinations
Figure 6.3 Turnaround Time Graph for FCFS, SJF, and RR

Figure 6.4 Turnaround Time Graph for six Combinations
Figure 6.5 Response Time Graph for FCFS, SJF, and RR

Figure 6.6 Response Time Graph for six Combinations
6.2 Summary of contributions

- This thesis demonstrates a new technique to select the best combination of job sequence from all the possible combinations using different combination of scheduling algorithms.

- Priority based multiple queue approach is used to solve the problem of selecting the best combination of job sequence. This increases the performance of a scheduler and in turn the Grid environment.

- Facilitates user to enter jobs according to his/her choice using a simulated environment. In the actual Grid environment, user may specify the priority of the job or scenario of the job.

- Priority based multiple queue scheduling algorithm uses first come first serve, shortest job first, round robin scheduling in order to find the best combination for a job sequence.

6.3 Improving performance of Web Servers

Improving the performance of Web servers has become a critical issue in coping with the increasing use of network-based services. The critical nature of many online transactions and distributed services mandate design of high performance Web servers since such servers are anticipated to be the bottleneck in hosting network-based services. To achieve this, several design issues is investigated in cluster-based Web servers.

6.3.1 Contributions

The major contributions of this thesis are summarized as follows. First, a coscheduled Web server model is proposed, which adopts coscheduling mechanisms between communicating processes of Web servers.
to reduce the response time of requests. The round trip time is reduced; the communicating processes of the requests are scheduled by the operating system as soon as possible. Also, the round trip time is reduced when the coscheduling algorithms reduce the context switches of processes. Four different Web server models are analyzed; PRESS over VIA, coscheduled PRESS with DCS, coscheduled PRESS with DCS and blocking, and Adaptive, in 16-node and 32-node configurations and found that response times can be reduced significantly, especially in lightly loaded systems.

Second, the impact of SSL offering is investigated in cluster-based application servers, and proposed a backend forwarding mechanism to provide higher throughput and even load distribution among the cluster nodes by exploiting the low overhead of user-level communication. In SSL, servers can be used the session reuse scheme; where a client’s previous session information is used to avoid the expensive authentication process. The backend forwarding mechanism uses a session aware distribution policy at a web switch to maximize the possibility that each server takes advantage of session reuse. In addition, it has an intelligent load balancing scheme that forwards client requests from a heavily loaded backend node to a lightly loaded node to improve utilization across all nodes. This policy uses the underlying user-level communication mechanism for fast communication. Our results indicate that the backend forwarding scheme is quite useful to enhance the performance of SSL-enable servers.

Third, to use the NIC cache is proposed as an extended cache, while a typical Web server uses only the main memory as its data cache. Two alternative schemes are examined; the exclusive and inclusive caching schemes. The exclusive caching scheme uses the NIC memory as an extended cache for the main memory cache. This scheme increases the cache
hit ratio, and thus reduces the number of disk accesses. The exclusive scheme is particularly effective when the number of Web data items is large and the popularity skewness is low, as in the UCB trace. The inclusive caching scheme maintains a copy of the most frequently accessed data items (or files) in the NIC memory. Thus, the NIC cache is a subset of the main memory cache. The main advantage of this scheme is to reduce the intra-cluster communication traffic and the DMA latency. The inclusive scheme is particularly effective when the size of Web content is large and the popularity skewness is high.

6.4 Further works and Directions

In future, the improvements can be done by also taking into account the dynamic behavior of the grid resources. There might be cases when resource required for a job, which is ready to be executed, is busy and job then has to be put on a waiting queue. In future waiting queue can also be implemented to take the above mentioned problem into account.

Secondly, the algorithm can be made more optimized to include more comparison parameters for comparing different queues. In this thesis only three parameters are considered i.e. waiting time response time and throughput. Other performance parameters e.g. turnaround time, CPU utilization etc. can also be added in future.

Thirdly, this thesis is carried out in a simulated environment. In future the work done here can be used in an actual Grid environment.

The work performed in this thesis can be used to optimize the scheduling algorithm for the Grid. This is because it provides the user friendly environment and choice to user to select his combination according to his choice.
A further extension to this work would be used to enhance the different scheduling algorithm to improve the performance, quality of service etc.

In the above scheduling algorithm, only CPU bound jobs are considered, in future the algorithm can be enhanced to include both CPU bound and Input bound jobs. Here it is assumed that, task scheduling is handling by local scheduler or resource management service, in future this part can be included in the algorithm to make it more efficient. In future, similar types of jobs group together and find the best combination of that group can be used to enhance the performance of the scheduling algorithm.

**To Improve Performance of Web Server will be suggested**

As the current trend of merging heterogeneous services into Web services continues, the requirements of Web servers would become increasingly more diverse compared to traditional Web servers. Additionally, security would become a highly critical issue in most enterprise Web servers. Thus, to extend our research is planned to meet the future demand imposed on Web servers.

First, our proposed scheme for SSL connections improves performance in a Web server cluster. Based on these results, we will expand our work to a real deployment. We will implement SSL-enabled content-aware and content-oblivious webs witches. Since there is no publicly available distributed Web server that provides SSL connections, to implement SSL functions are planned in a distributed Web server.

Second, Quality of Service (QoS) will be focused in cluster-based Web servers with respect to required contents, clients groups and service
availability. Providing QoS not only in network transmission, but also in the processing capacity of Web servers has become highly desirable. The previous studies on QoS in cluster based Web servers suggested that the nodes in a cluster are divided according to the required service load for each class. But this coarse grain partitioning may cause low utilization of Web servers and may not provide flexible adjustment on the current service requirements. Thus, we plan to propose a more adaptive and fine grain QoS scheme based on current local server’s utilization and global information. Unlike previous studies, service decisions will be made in the local node instead of in the central system of a cluster.

Third, to investigate the problems are planned with the storage systems used in cluster based Web servers, with an emphasis on improving performance of storage bounded services. With the increasing number of data-centric Web services, reliant and secure storage systems grow in demand. Therefore, the technologies are identified which are able to satisfy requirements from multiple dimensions, such as capacity, security, latency and fault tolerant performance.