ABSTRACT

The main objective of this research work is to improve the effective handling of Web requests in distributed environment through the enhancement of services such as differentiated services at the application level, load balancing, and real-time scheduling. All methodologies have been used at the server side to enhance the value of distributed environment services. Scheduling, Load balancing and Heterogeneity are the crucial issues in the performance of Web applications in distributed environment.

TCP connection management mechanisms at application level for Web servers provide two different levels of Web services, namely, member and non-member services, by setting different holding times for the services. The incoming requests are differentiated using IP address that are stored in the database at the server and later authenticated by the Web server. The proposed approach starts user level differentiation ahead of entering the requests into the server log and assigns appropriate holding times to every client immediately after establishing a TCP connection. The rigid policy in the system carries with it the danger of service starvation, in which case a non-member request waits indefinitely for its turn due to a continual flow of member requests. In this proposed system, a processing priority weight is predefined for each queue to avoid service starvation and the initial weight depends on the priority of the queue. The number of requests processed at a time is the dependent on weight for each queue. When the number of requests previously in the queue is lower than normal, then the predefined weight to
that queue is reduced. When the number of requests in the queue is higher than normal, then the predefined weight to that queue is increased. Variations in the throughput of the incoming request categories are managed and adjusted using a customized tuning algorithm based on the current weight of the queues.

If all the incoming requests are belonging to similar categories, then the performances of the clients degrade due to the overheads of differentiating the requests. A novel architecture is proposed for handling all the incoming requests of identical categories and this was found to improve the effective handling of the requests. A proposed architecture takes some requests from the member queue and puts in non-member queue to retain the QoS.

The LDMA and the LTPT are two load balancing strategies proposed for distributing/processing the requests among the clustered Web servers in LAN environment. Moving a large number of requests between Web servers requires a high bandwidth and it is difficult if a node's status quickly changes in relation to the time needed to move requests. The LDMA is based on all-to-all communication without gathering load information and request transfer policies among the LAN clustered Web servers. Mobile agents are used for decision making among the clustered Web servers for processing the incoming requests. The LDMA framework uses the concept of “ranked Web-servers,” i.e., each server is assigned a rank based on the priority that is given for processing a request. Decision-making is made by the interaction of mobile agents between the Web servers. All Web servers are organized in a mesh topology.
The LTPT is a classical centralized decision making algorithm and is seldom used in practice because it requires that the processing times be known in advance. The incoming requests are forwarded to the server that has required least total time for processing the previous requests in the queue. The concept of "Ranked Web server" is used in the LTPT in order to avoid server ties during the processing of the requests.

The WLDMA is a new dynamic load balancing strategy which redistributes the requests to the less loaded Web server in WAN. In the WLDMA, each server processes client requests independently and interacts with other servers periodically to share the workload. In the existing WAN load balancing schemes, job reallocation is done when the workload on a server exceeds locally fixed threshold value. In the WLDMA, job reallocation is based on the load of the other servers in the WAN. The performance metrics such as throughput, average response time, network traffic and load deviation are used for evaluating the proposed load balancing schemes.

Most of the real-time scheduling algorithms are open loop algorithms. In these algorithms, scheduling decisions are based on the worst-case estimation of the task parameters. The requests in the queue for longer time than threshold value are dropped in best effort model Web server. These requests are effectively handled by the proposed sequential and differentiated Web server architecture that provides better response time and task miss ratio. This architecture is composed of a connection scheduler, a monitor, a controller and a server process. The controller calculates new relative deadline values for the requests depending on their position in the server queue and
reallocates the server process subsequently. Earliest Deadline First (EDF) is refined as a closed loop scheduling algorithm. Based on the simulation studies, closed loop real-time scheduling offers a better performance over the open loop scheduling. The proposed approaches for service differentiation at the application layer, load balancing and real-time scheduling provides better performance to the Web requests compared to the established paradigm in distributed environment.