ABSTRACT

A widely encountered problem in web servers over the Internet is the long response time. It is possible to reduce the response time of requests at a web server by simply changing the order in which we schedule the requests. Recently, the Shortest-Remaining-Processing-Time (SRPT) has been proposed for scheduling requests in web servers. The SRPT assumes that the response time of the requested file is strongly proportional to its size. However, depending only on the size of the file for determining the priority of the request is not enough, since it doesn't take into consideration the client-server interaction through the Internet, where web servers are mainly used. In the Internet, the clients are geographically dispersed which presents high diversity in path bandwidth, round-trip time and packet loss characteristics. To account for these parameters, this thesis proposes a new scheduling policy for processing static HTTP requests in web servers that better estimates the response time. We call this policy, Shortest-Remaining-Response-Time (SRRT).

Our approach benefits from the TCP implementation to capture useful scheduling information about the interaction between the server and the client through the network. The SRRT prioritizes requests based on a combination of the current Round-Trip-Time (RTT), TCP window size and the size of what remains of the requested file. The requests which have the shortest estimated remaining response time receive higher priorities.

The implementation is done at the kernel level for controlling the order in which socket buffers are drained into the network. Our experiment uses the Linux operating system and the Apache web server. In the experiment the requests are generated by the Scalable URL Request Generator (SURGE) workload generator, and the WAN is represented by Network Emulation (netem).

We compare SRRT to SRPT and processor-sharing (PS) policies. SRRT and SRPT show an improvement over PS. However, the SRRT shows the best improvement in the mean response time. SRRT gives an average improvement of about 7.5% over SRPT for both 10Mbps and 100Mbps links and under all loads. For 10Mbps link, the maximum improvement of SRRT over SRPT is 13.2%. While for the 100Mbps link the maximum improvement is 11.6%.

This improvement comes at a negligible expense in response time for long requests. We found that under 100Mbps link, only 1.5% of long requests have longer response times. The longest request under SRRT has an increase in response time by a factor 1.7 over PS. For 10Mbps link, only 2.4% of requests are penalized, and SRRT increases the longest request time by a factor 2.2 over PS.