MODIFIED DIFFERENTIATED WEB SERVICE APPLICATION LEVEL TCP CONNECTION MODEL

M.ARAMUDHAN* V.RHYMEND UTHAIARAJ **
Research scholar*, Asst.professor**
College of Engineering, Ramanujan Computing center
Anna University, Chennai – 25

Abstract: In current web service model, there is no priority among the requests while being processed by servers and transmitted over the network. Yoon-Jung Rhee and Tai-yun-kim [1] proposed an application level TCP connection management mechanism for web servers to differentiate the service as members and non-members of the web site using static IPaddress and setting different holding time for TCP connections. In existing model, server differentiated the service after TCP establishment where as in the proposed model service differentiation starts before TCP establishment and assign the holding time immediately after entered in to the server log file. In proposed model, the long idle holding time of TCP connection is abruptly terminated and increase reliability. The proposed frame model improves member's performance compared to the existing model and effectively provides quality of service (QoS) even in the absence of operating system and network support. For each priority separate queues are used for processing in the basic differentiation service. In the proposed model, single queue along with scheduling algorithm is used to differentiate the service. Average response time of the member site is considerably improved by 11-17 %. Prototype implementation for the models has been developed and performance is compared.

Keywords: Differentiated service, Persistent connection, prioritized request, scheduling algorithm

1. Introduction

In web transaction, clients send requests to servers; servers process them and send appropriate responses to the clients. Each request utilizes the resources of the servers. Even discarding requests consume the resources of the server. Concurrent transactions with a server compete for the resources in the network and end systems. In request based admission model, servers will allocate one socket connection per request. In e-commerce applications each transaction consists of number of requests and responses. Discarding any one-request may lead to the failure of the complete transaction and wastage of server resources. Therefore, server should allocate resources for all requests of the particular transaction that already accepted and fulfill the requirements of the users. Managing quality of service will become more important in enterprise web services as there is a need to cater various categories of users accessing services in different contexts and expecting different service levels [7].

In traditional model, there is no priority among transactions. Due to the explosive growth of Internet, user accesses on the popular web sites are exponentially increased. Hence, popular servers suffer from deficiency of resources such as network interface card, physical memory and disks [2]. In some cases, not all transactions are equally important to the clients or to the server and some applications need to treat them differently. The web site wishes to offer better service to the members of the site than non-members. The web service model treats all transactions equally according to the best- effort service. The service, which is differentiated, based on the priority such as web object (HTML/inline images), members and non-members, foreground and background tasks, mobile or PC is called differentiated service.

The latest version HTTP /1.1(Hyper Text Transfer Protocol) reduces latency and overhead by using same TCP connection for multiple requests [1] The service, which uses pre-established connection for transaction, is called persistent connection. HTTP does not specify the explicit connection closing time but suggests time out value beyond which an inactive connection should be closed. Latest HTTP uses a certain fixed holding time model. In existing TCP application management mechanism, the holding time is allotted to the members and non-members when the users log into the server. Members got extra holding time than non-members. Server resources are idle when the client does not send any requests during the allocated holding time. Current latency problems are caused not only by networks problem, but also by overloading servers having limited resources.

The proposed idea of the work is to differentiate the service before TCP connection in order to improve the member's performance, avoid long idle time of the users to increase the reliability of the model and use single queue with scheduling algorithm to differentiate the service in Apache web server (User level) and application model. This mechanism provides effective results in the absence of kernel and network mechanisms.

The rest of the paper is structured as follows. Section 2 describes problem in the existing mechanism. Section 3 presents modified application level TCP connection mechanism and Apache web server. Section 4 describes simulation of the proposed model. Finally, Section 5 provides the conclusions.

2. Problems in the Existing mechanism

In existing model, the server differentiated the requests after the establishment of the TCP connection. The requests are prioritized using static IP address. The IP addresses of members are stored in the database [3]. Server starts finding the incoming request is member or not after users log into the server [1]. The time taken for differentiating the request is considerably increased when the size of member database is large. Clients do not send further requests to the server in the assigned holding time decreases the reliability of the existing model. The problems in the existing model are sorted out in the proposed model. The following are considered in the modified application level TCP connection model

- Strict priority: This model schedules all member requests before non-member requests, even when non-members are waiting.
- Preemptive processing: Non-Member requests are stopped while the members entering the server during the overload condition.
- Dynamic capacity: The connection with the server is terminated abruptly when Members and non-Members are inactive for certain time in its holding time (like twice round trip propagation time).

Traditional Apache web server handling the requests in best effort model is shown in figure 1.

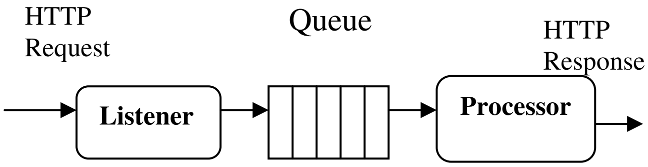

Figure 1: Apache Web server request handling using Best effort model

Listener listens on port 80 and accepts new connection. Accepted connections are placed in the queue and forward to the processor in FIFO manner. This mechanism is not suitable for differentiated service. Sook-HyanRyu and Kim [5] proposed a differentiated model for handling the requests is shown in figure 2.

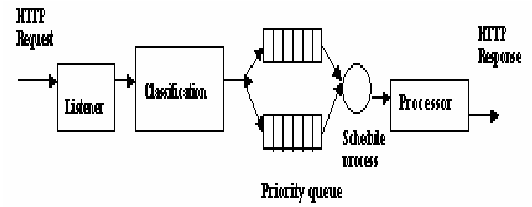

Figure 2: Differentiated service in Apache Server

Listener listens on port 80 and accepts new connection. Accepted requests are classified and placed on the appropriate queues. The number of priority queues implies number of differentiation levels. A schedule process selects the next request to the processor based on the scheduling policy.

3. Modified application level TCP connection Model

The service is classified into members and non-members based on IPaddress of the machine. Members pay fee for the service and non-members may get the services at free of cost. In modified model, server starts finding priority of requests after socket establishment and assigns holding time at the earliest after TCP connection. This model provides benefit only when the members in the database are large. Idle holding connections are abruptly terminated with different time for members and non-members in order to increase the reliability. Figure 3 shows the modified application level TCP connection model.

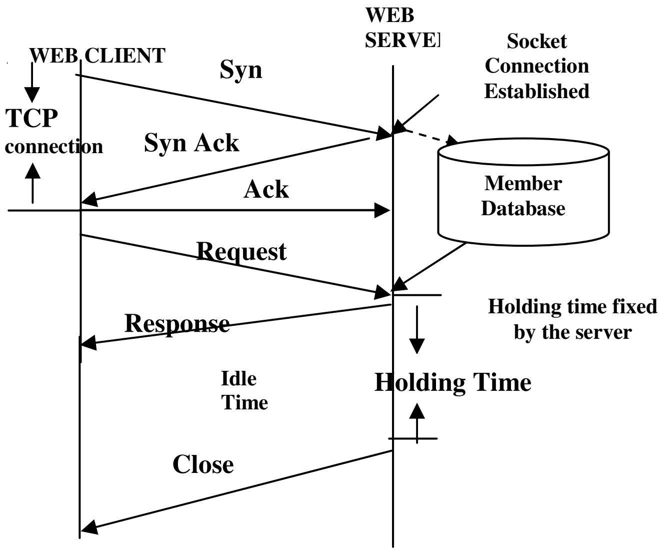

Figure 3: Modified Application level Connection Management Mechanism

In IEEE site, holding time for idle client is 20 minutes. But, the concept is implemented in kernel level. A single queue with scheduling algorithm is proposed for differentiated service in apache server instead of having priority queue. Figure 4 shows the modified differentiated service in apache server along with scheduling algorithm to differentiate the service and provides QoS.

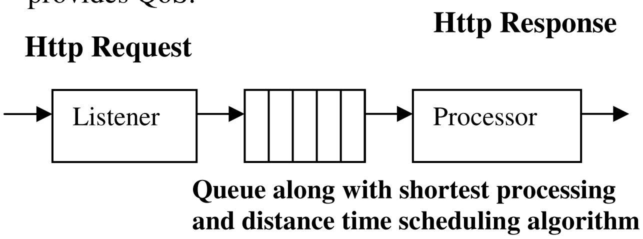
Http Request
Http Response
Listener
Processor
Queue along with shortest processing and distance time scheduling algorithm

Figure 4: Differentiated service using single queue along with scheduling algorithm

4. Prototype Implementation of the Proposed Model

The proposed idea is tested using prototype model instead of application level. Both existing and modified models are implemented in JAVA with some limitation and performance is compared. The test bed contains server and several clients. The server has an IBM system with x86 Family 6 models 8 stepping 10 AT/AT Compatible and 128 RAM running JAVA for Windows 2000. Table 1 shows time taken for differentiation of service after TCP connection and placed in the queue. Table 2 shows time taken for differentiation of service before TCP establishment and placed in the queue.

Size of Database (Number)	**No: of Clients**	**Average Latency (ms)**
20	5	301.23
	10	333.45
	15	356.65
50	5	306.3
	10	335.45
	15	368.93
100	5	372
	10	378
	15	390.4

Table 1: Differentiation take place after TCP establishment

The single sequential server model is implemented which is different from multithreaded or multiprocessor model. Sequential server receives a request at a time. The implementation has done in two ways

(i) Based on the priorities, the requests are placed in the appropriate queues and processed on priority.
(ii) Using single queue for all requests and scheduling rule to differentiate the services and improve over all QoS.

Size of Database (Number)	**No: of Clients**	**Average Latency (ms)**
20	5	294
	10	366.25
	15	388.33
50	5	311.23
	10	366.75
	15	404.96
100	5	386
	10	392.92
	15	396.64

Table 2: Differentiation take place before TCP establishment

Size of Data	Windows 98	Windows 2000	Linux
20k	2968	2578	2196
40k	3108	2678	2277
60k	3609	2968	2689
100k	7328	6100	4320
120K	8087	7158	5203

Table 3: Time taken by different Operating system for transferring data using TCP

Each client side class is made to generate requests at the rate of 5,10,15 requests per second through multithreading. Database size is varied with number having 20, 50 and 100 respectively. Linux operating system provides better data transfer using TCP is shown in table 3.Table 4 shows an average response time of request having no policy using FIFO. The holding time is 50s. The performance is slightly better in single queue along with weighted shortest processing time scheduling algorithm (WSPT). To schedule a set of jobs on a single machine given that all jobs are available at time =0. The WSPT schedules a set of jobs by decreasing order using a formula, which includes priority weight, processing time. In future, distance time is also incorporate in this scheduling. WSPT incorporates weight factor, thereby allowing differentiation of requests with various priority weight. Table 5 shows average response time of request using single queue. In priority scheduling, FIFO is used to process the request after placed in the appropriate queue. In single queue, all the requests are placed in a queue and process order based on priority weight, processing time and distance time of the request. The performance is slightly better in single queue with weighted shortest processing and distance time scheduling algorithm.

No. of Clients	Size of Database (Number)	Average Response Time	
		Differentiation before TCP connection (ms)	Differentiation after TCP connection (ms)
5	50	318.6	340.6
10	50	377.2	399.2
15	50	404.4	421.1
5	100	394.6	415.6
10	100	392.2	434.2
15	100	409	441.06

Table 4: average response time of request using FIFO scheduling.

No. Of Clients	Size of Database (Number)	Average Response Time	
		Differentiation before TCP connection (ms)	Differentiation after TCP connection (ms)
5	50	318.2	329.4
10	50	369.9	383.7
15	50	379.2	404.8
5	100	388.28	392.24
10	100	388.9	407.90
15	100	398.34	416.06

Table 4: average response time of member request using multiple queues with priority scheduling

Table 5 shows average response time of the member request of the site using single queues. The performance of member is increased 11 % to 17%. In prototype implementation clients issue 65% of member requests and the remainder issue requests of default class. Our experiments were conducted to access HTML files in server. Current scenario offers large sizes of multimedia data such as audio, video and images that require reliable service. It is difficult to meet accessing with in the holding time. The proposed mechanism may degrade the server's performance when connections with member clients increase rapidly. To overcome this, for every n requests of a higher priority m requests of a lower priority should also processed.

No. Of Clients	Size of Database (Number)	Average Response Time	
		Differentiation before TCP connection (ms)	Differentiation after TCP connection (ms)
5	50	314.9	324.21
10	50	359.9	376.71
15	50	376.82	401.34
5	100	386.41	390.26
10	100	387.54	395.55
15	100	395.43	407.32

Table 5: average response time of the member request using single queue along with shortest processing Time scheduling algorithm.

In practical, web server creates multiple service session threads in order to serve multiple clients simultaneously. Perhaps the greater number of session threads worse the performance of services. In our experiment service is processed one at a time and the average amount of time that each client takes to invoke and receive results for maximum 15 clients. In real life thousands of requests could be directed to server in which case proposed idea as a performance bottleneck. Also, some load balancing mechanism can reduce the severity of this impact; it is still open issue for further research.

5. Conclusion and Future work

It is impossible for a web server having limited resources to process all the requests from the clients with high quality of service. The metric for QoS is latency, which is reduced by starts finding differentiation before TCP establishment. It gives yielding only when the member database is large. Idle client connections are abruptly terminated and increase reliability. Single queue along with scheduling algorithm is used for differentiation instead of priority queues. It shows improved performance in the timeliness. Linux operating system provides better performance in TCP data transfer. The future work is to implement the proposed frame model using Apache web server in application level and develop scheduling patches. In this model strict priority is enforced. In future, for every m requests of higher priority n requests of lower priority should process policy will incorporated. The single queue along with WSPT scheduling gives slight better performance than basic differentiation service.

References:

[1] Yoon-Jung Rhee, Eun-Sil Hyunz and Tai-Yun Kimy, "Connection management for QoS service on the web" Journal of Network and Computer Applications (2002) 25, 57-68

[2] J. Almedia, M. Dabu, A. Manikntty & P. Cao "Providing Differentiated Levels of Service in Web Content Hosting." In Proc. 1998 Workshop on Internet Server Performance. Madison, Wisconsin.

[3] Bhatti N, Fredrich R. Web Server support for Tiered services. IEEE Network 2000,13(5); 64- 71

[4] Apache HTTP Server Project, 1998. http://www.apache.org/

[5] Sook-Hyun Ryn, Jae-Young Kim and James Won-ki-Hong Towards Supporting Differentiated Quality of Web Service (downloaded material)

[6]W.R.Stevens, "TCP/IP Illustrated Vol I Reading, MA Addison –Wesley. 1994

[7]Amit Sharma, Hemant Adarkar and Shubhahis sengupta, "Managing QoS through prioritization in web services " proceedings of the Fourth International conference on Web information systems Engineering workshops, 2004