Performance Measurement and Queueing Analysis at Medium-High Blocking Probability of Parallel Connection Servers with Identical Service Rates

CHUNG-PING CHEN* and YING-WEN BAI**
Graduate Institute of Applied Science and Engineering, Fu Jen Catholic University*
Department of Electronic Engineering, National Taipei University of Technology*
Department of Electrical Engineering, Fu Jen Catholic University**
510 Chung Cheng Rd. Hsinchuang, Taipei County 24205
TAIWAN, R.O.C.
491598038@mail.fju.edu.tw, bai@ee.fju.edu.tw

Abstract: - In this paper we propose a performance measurement and queueing analysis for medium-high blocking probability where the service rate is less than equal to the arrival rate of parallel connection servers, and for which we can estimate the system response time. First, we calculate the system response time of the parallel network. Second, we simulate the queueing model of the parallel network and calculate the system response time. Third, we measure the system response time by using different numbers of ASP multiplication loops to represent the different service rates of the parallel network. Fourth, we take 30,000 data for imitation as the foundation and use linear regression to compare the measurements of both the system response time and the search service rates of the actual system under different service load conditions. Fifth, we compare the system response time of a single server with up to ten parallel connection servers and compute the discrepancy in their efficiency.

Key-Words: - Parallel Network, Web Servers, Service Rate, Parallel Connection Servers, Queueing Anallysis.

1 Introduction

When the network structure is more and more complicated and Web service loads become bigger and bigger, this result generates a high blocking probability of the Web service and lengthens the system response time, with unacceptable results. In the designs of server architecture many improved methods have been put forth, as stated in the following examples. Upgrading the equipment using a multiple connection method increases the service rate by making use of a load balance mechanism in order to strengthen the Web service capability [1]. A Gigabit Ethernet between the network interface controller and the corresponding CPU increases the network flow and reduces the network transmission time [2]. Also, the use of a preemption mode improves the utilization of the network and handles the high priority class information in advance. Then the low priority class makes use of the blank in the idle period of flow, thus reducing the whole blocking probability of the network [3].

From the aspect of a high blocking probability, a four-band mixture (FWM) of the wavelength partition multi-task network is selected to assign the small path first (SASPF) algorithm to select the path according to the design parameter Vm and to name a wavelength. If the Vm value is too low, there may arise a high blocking probability. If the Vm value is obviously too high, this will influence the FWM network performance [4]. Traffic Engineering uses a mixture of the short-holding-time queue of high blocking probability and the general blocked-calls-cleared queue of low blocking probability to control the latency time. The short-holding-time queue serves to improve the flow rate and to reduce the blocking probability of the cell phone and the allocation system [5, 6]. ISDN in broadband can be used for status reduction when the system generates a high blocking probability [7].

As for the service rate, the Code Division Multiple Access (CDMA) wireless network provides the closed form algorithm which controls the capacity, solves the problem by utilizing the reversal links for the capacity and generates a fairer allocation [8]. In the CDMA data network the power is constrained by discrete rate allocation (DRA). Verification is the problem of NP-complete by the selective rate reduction (SRR) scheme, a genetic algorithm (GA) and the improved genetic algorithm (IGA), all these together allowing the structure to provide distinct data services [9]. In dynamic operations, when the many program codes sharing the CDMA's service rate assign to each user a fair queueing (FQ) algorithm which controls the weight

and the power in time, this results in an improved network performance [10]. The CDMA 1 xEV system in the wireless data network is mainly used to provide high-speed data communication service. The channel-aware scheduling algorithms, if used in the channel fluctuation in the wireless data network, effectively improve the performance [11].

To predict the performance and efficiency of the network, the SHRiNK method is used to input the flow into the system that, according to the system parameters, is immediately extrapolated from the computing system [12]. The SNMP-based network is based on a measurement of the performance of the telecommunication domain [13]. Les Cottrell clicks to interact with a network's measurement. This testing format includes various network paths and flow rates [14]. Bikfalvi suggests taking the quality of service (QoS) parameter as the management foundation of the network measurement system and then starts experimenting by using the GNU/Linux platform. This procedure improves the communication information by utilizing the program in the key station. The dispersed type SNMP software provides the measurement while running a QoS measurement [15].

This paper uses a parallel method to improve the service rate and to reduce the problem of a high blocking probability, and, by making use of an experiment result, predicts the performance and efficiency of the network.

The server architecture can be enhanced by a multi-layer serial-parallel connection. This method analyzes the error margin of the system response time between the parallel queue and the physical measurement of the server system [16, 17]. We aim at a parallel queue to imitate the system response time of both the Web requests and the measurement and compare the error margin between the measurement and the analysis.

A single server measures the quality of service (QoS) of the response time, which can be divided into three kinds: Less than 0.1s, a low blocking probability, from 0.1-10s, a medium blocking probability, and greater than 10s, a high blocking probability [18].

As for the network system response times, their factors of influence which induce sorting are as follows [19]:

Network system response time = Network Time + Web Site Time + Time of DNS + Web Page Size + multifarious degrees of click.

Network Time = Node Latency + Transmission Time.

Web Site Time = Queuing Time + Service Time.

Service Time = CPU performance + performance of disk driver + network processing performance.

As far as the network transmission time is a result of the characteristics of the wire material and other equipment of the network, this aspect of the network transmission time doesn't fall within the range of this paper. We are only concerned with the handling time of the Website service time of the server and the handling time required by a packet waiting in a buffer. In order to identify a queue network theory, we use the total number of times of the ASP multiplication loop and change as the CPU service rate. When the multiplication loop increases, as the CPU transaction time also increases, the service rate of a server, therefore, can be adjusted.

The organization of this paper is as follows: In Section 2 we turn the server system of the parallel network into a queue model. In Section 3 we use as a tool the queue network simulation. The simulation of the single queue uses the result both for the 1-400 requests/sec in the service rate and for the arrival rate, which is the foundation of the linear regression algorithm identification. In Section 4 we physically configure 2-10 parallel connection servers of identical service rates. We run the linear regression algorithm to analyze the performance. In Section 5 we draw our conclusions according to the results of the experiment.

2 The Queueing Model for the Parallel Connection Servers

In the parallel connection servers we create one physical 1-10 stage server computer network environment and use an algorithm simplification to obtain a single equivalent network model. We use the network software simulation system response time gained by the equivalent model for the measurement. The system definition and model parameter are shown in Table 1. The unit of measurement here is msec.

Table 1 System definition and model parameters

Parameter	Description	Definition
λ	Web request rate	Requests/sec
μ_{P_n}	Service rate of the n^{th} server	Requests/sec
μ_{eq_n}	Equivalent service rate of the n^{th} parallel network	Requests/sec
P_n	Probability of the n^{th} parallel queue	
$E_{P_n}(T)$	System response time of the n^{th} queue	ms
$E_{eq_n}(T)$	Equivalent system	ms

	response time of the n^{th} parallel network	
ρ_{P_n}	Utilization of the n^{th} server	
B	The packet amount in the buffer (click amount)	

We use the idea of a parallel equivalent electric circuit as our analytical foundation and verify the performance by measuring, and we use approximate equations for the multiple stages of the parallel network. At the beginning we use queueing networks and Markov Chains [20] to analyze parallel networks with the following assumptions.

- All requests are first in first out first in the system.
- The total of the requests in the system is unlimited.
- The request can leave the system from another node.
- All service times are exponentially distributed.

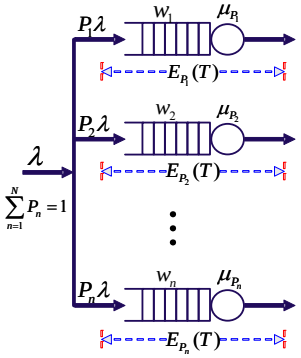

Fig. 1 Equivalent model of parallel connection servers

If a customer demands a packet, we set the arrival rate λ. As the packet arrives at the server set, it is assigned into each parallel node one after another. Because there is no need to detect the network status or operation, the assigning time is so short that we can neglect it. In Fig. 1 the packet, after getting into the node, will have one small segment of waiting time in the buffer, and then it is handled by the processor before it leaves the queue node.

To reduce the complexity of the measurement we install the measurement tool for the same hardware platform. Using 10 identical servers in the actual measurement, we link them to a hub of 16 ports which provides a parallel connection of 1-10 servers. For the arrival rate we use the server and the WebStress Tool to test the software simulation. The parallel servers act as the cluster server. As the parallel server has the same hardware platform, we obtain both the same service rate, $\mu_{P_1} = \mu_{P_2} = \ldots = \mu_{P_{10}}$, and the same arrival rate, $P_1\lambda_{P_1} = P_2\lambda_{P_2} = \ldots = P_{10}\lambda_{P_{10}}$. When the parameters have been identified, this raises the accuracy of measurement and reduces the degree of complication. The measurements of the hardware specification and connection are shown in Table 2 and Fig. 2.

Table 2 Specification of parallel connection servers

Type	CPU	Hardware	Tool/Web Server
Client	3.2GHz	**RAM:** 2G **OS:** Windows Advanced Server 2003 **Network:** LAN (100Mb/sec)	Webserver Stress Tool
Server 1	2.4GHz		IIS 6.0
Server 2			
Server 3			
Server 4			
Server 5			
Server 6			
Server 7			
Server 8			
Server 9			
Server 10			

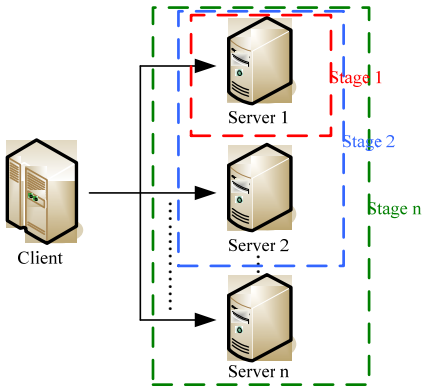

Fig. 2 Measurement of multiple parallel stages

3 The Simulation of the Linear Regression Algorithm

To analyze the error margin of a parallel network we have to check the accuracy of the previous equivalent equation. After modeling we make use of the software network simulation tool to support us during the simulation stage. For our simulation we use the Queuing Network Analysis Tool (QNAT) [21]; by means of simulation, it can create a closed or open queueing network. With the single queue simulation as our foundation, the service rate ranges from 1-300 requests/sec, while the arrival rate ranges from 1-100 requests/sec. This simulation result is the main basis of comparison for the service rate.

Based on Fig. 3 we determine the arrival rate and the service rate, each ranging from 1 request/sec to 300 requests/sec, and obtain the simulation results shown in Fig. 4.

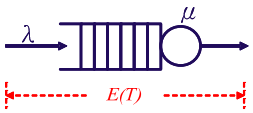

Fig. 3. A single equivalent queue

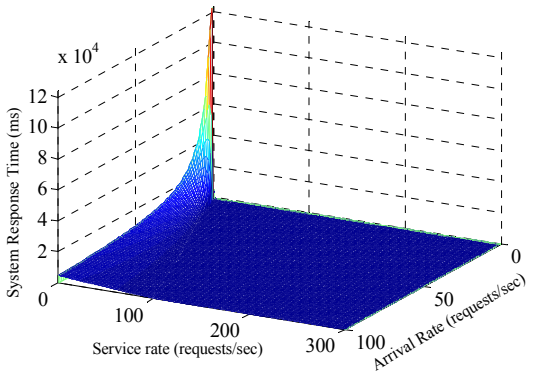

Fig. 4. The simulation results of a single equivalent queue

We use the WebServer Stress Tool [18] to measure the system response time of a single server and of parallel connection servers and obtain the data of measurement according to the linear regression of the simulation data to calculate the service rate of the parallel connection servers. From the comparison we gain the service rate, which has a minimum error margin [22].

4 The Performance Measurement of the Parallel Connection Servers

To verify the system performance of the parallel connection model of a real network we use a local area network as the measuring environment. We represent the parallel connection servers by means of the queue model that is divided into ten measurement stages. We use the WebServer Stress Tool measurement software to obtain the Milt-node parallel service rate of the ASP network. The service time is consists of the system response time of the parallel connection servers. We set up the Server1 IP address as 192.168.0.101, the Server2 IP address as 192.168.0.102 and so up to the Server10 IP address as 192.168.0.110.

The client port requests the network to run a Web page service. We can neglect the Web page access time and only concentrate on the parallel connection server performance; therefore the Web page provides just simple data in a mathematical operation. The first stage of the measurement is for the single server, and then, as shown in Fig. 2, the second stage increases with the number of parallel connection servers.

In the actual measurement, in order to both adjust the service rate and tranfer the simulation data of the server, we use the ASP Web page execution multiplication loop. The ASP Web page controls the service time of the server CPU through the execution multiplication loop. The multiplication loop has 1, 1 K, 10 K, 20 K, …, 190 K, 200 K, 300 K, …, 900 K, 1 M times and both calculates and compares the system response time for every case.

4.1 The Measurement Results of the Linear Regression Algorithm

To get the equivalent service rate of the server we compare the simulation result with the measurement result, by utilizing the linear regression method of the least square error margin value. We extract the parameter according to the steps shown in Fig. 5. By comparing the results of the measurement data and the simulation of the single queue data we obtain the minimum error margin after running linear regression [17]. By using Fig. 5 steps 1-5 with 1-10 sets of system response times of parallel connection servers we obtain the results shown in Fig. 6. The service rate of the parallel connection servers is at the lowest point in the diagram.

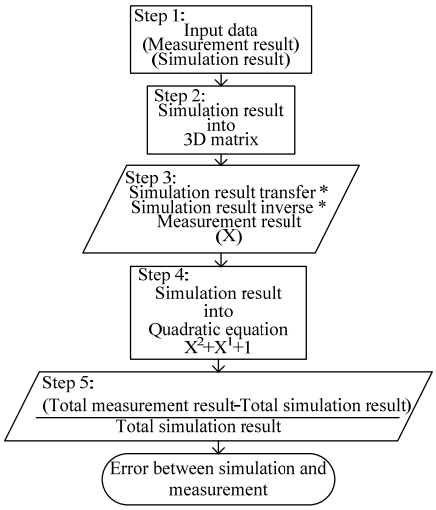

Fig. 5 The flowchart of the analysis program.

For the result of the linear regression's least square method, Fig. 6 shows in the y-axis the system response time by logarithms. It also shows that ASP=1 multiplication loop, and that the linear regression gets the value of the equivalent service rate of a single server for the 232 requests/sec and for two parallel connection servers for 225 requests/sec.

According to the same procedure we find the service rate for ten parallel connection servers to be 232, 232, 226, 228, 220, 222, 236 and 236 requests/sec. We check the result of the single queue simulation against a system arrival rate and find that it is 61 requests/sec in the equivalent arrival rate of the single server. In the same way we find that for two parallel connection servers it is 53 requests/sec, and for ten parallel connection servers it is 62, 61, 58, 59, 58, 58, 61 and 61 requests/sec. Having obtained the service and arrival rates from the linear regression, we use the formula $\rho = \frac{\lambda}{\mu}$ and find the utilization rate to be 0.24-0.27. The formula $E_{p_{eq_n}}(T) = \frac{1}{\mu_{p_{eq}} - \lambda_i} = E_{p_i}$ under $\mu >> \lambda$ in computing both the system response time and the actual system response time results in an error margin of 8.18%.

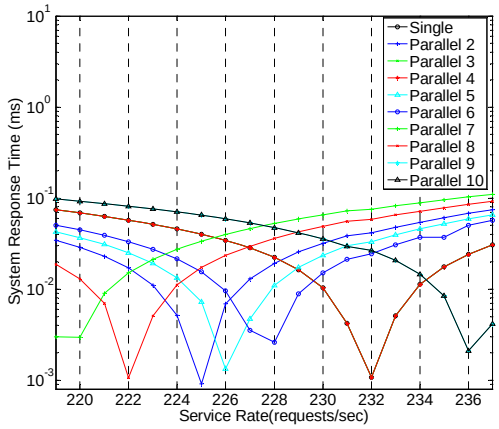

Fig.6 Error of the μ to the system response time of 1 loops

4.2 The Performance Measurement of 1-10 Stage Parallel Connection Servers

Of the measurement results given below, an ASP multiplication loop of 30K with a system response time of 0.1 sec is a Light Loading, a loop of 40K-160K with a system response time of 0.1-10secs is a Middle Loading, and a loop above 160 K with a system response time of 10 secs is a Heavy Loading. The parallel connection server system response time changes in three regions, as shown in Figs. 7-9.

Fig. 7 is the Light Loading of ASP = 10 K. From a single server to ten servers of the system response time there is no obvious change.

Fig. 8 shows the system response time of the Middle Loading of ASP = 90 K. The diagram observes a single server and two servers. A larger number of users extends the system response time. When a single server and two servers are parallel at 100 users, their system response times differ at 7918 ms. The system response time of a single server is about 3.38 times that of two servers, but below 20 users the system response time does not differ greatly.

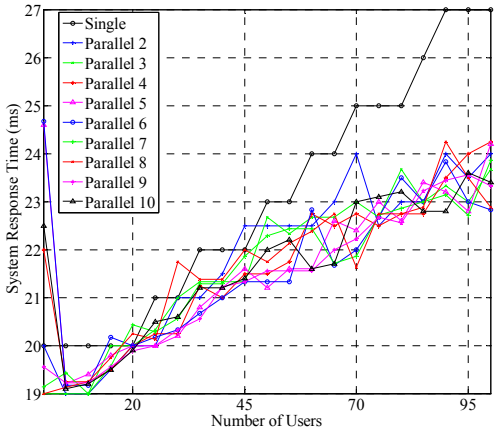

Fig. 7 System response time of the parallel connection servers with 10K loops

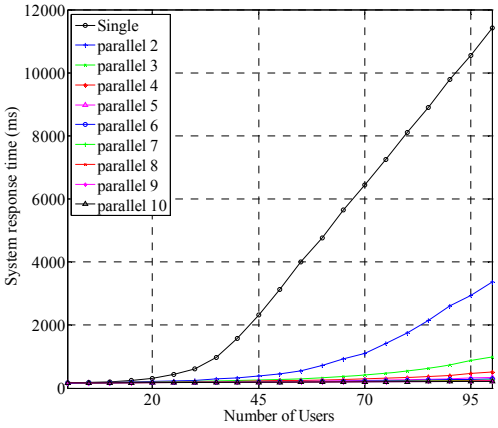

Fig. 8 System response time of the parallel connection servers with 90K loops

Fig. 9 is ASP=200 K, a Heavy Loading. The system response time of ten servers is 1/10 that of a single server, descending at 1/n times, where n is the number of parallel connection servers.

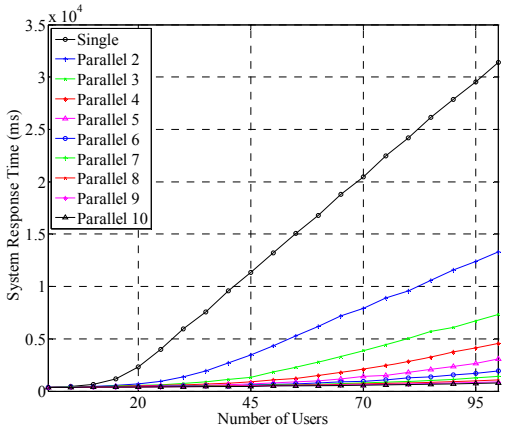

Fig. 9 System response time of the parallel connection servers with 200K loops

Fig. 10 is a Light Loading where the parallel number is 2-10. The diagram shows that the performance of the parallel connection servers declines at most by 30%.

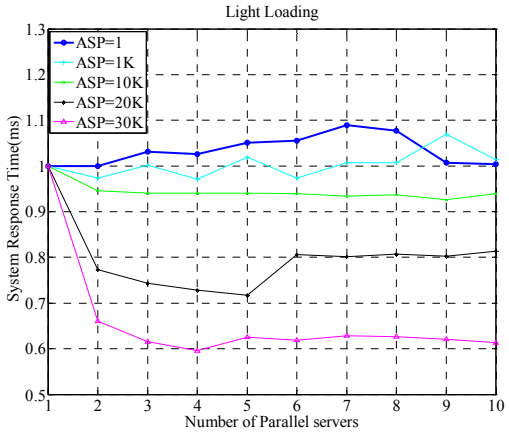

Fig. 10 System response time of the ASP=01-30K

Fig. 11 is a Middle Loading; when the parallel number is two, the efficiency is highest, with the system response time reduced by 75%, but this increases again with a parallel number of four, without a similar level of effect.

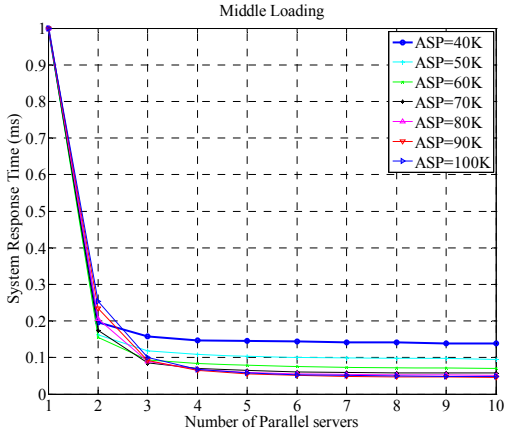

Fig. 11 System response time of the ASP=40K-100K

Fig. 12 is a Heavy Loading, where 2-10 servers are parallel, and the system response time is reduced by 50%.

In Fig. 13 the system response time of the Light Loading changes from a single server to ten parallel connection servers. Observe that in the diagram as the system response time has not changed greatly, the parallel system has no economic efficiency

Fig. 14 shows the response time of the Middle Loading. The system response time of a single server is between 592-4960 ms. When the four servers are parallel, the effect of the system response time gradually slows down; when ten servers are parallel, the system response time is reduced to 82-239 ms.

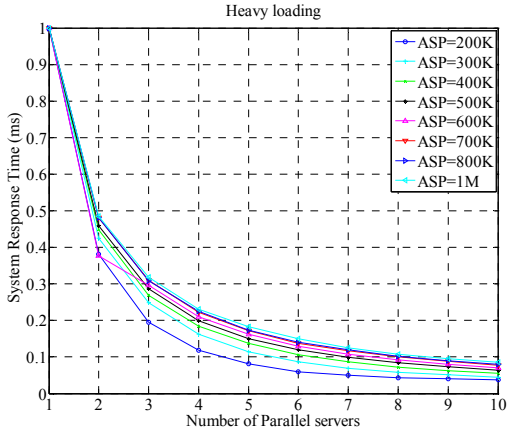

Fig. 12 System response time of the ASP=200K-1M

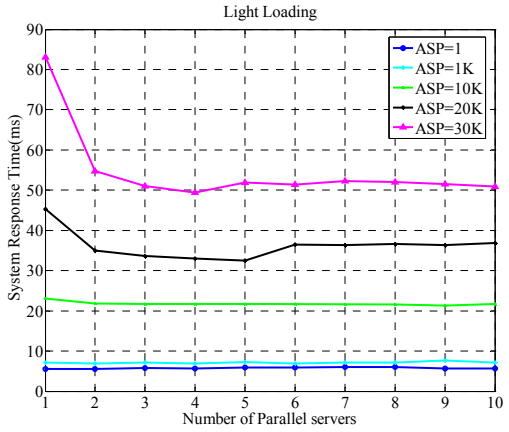

Fig. 13 With Light Loading of parallel connection servers

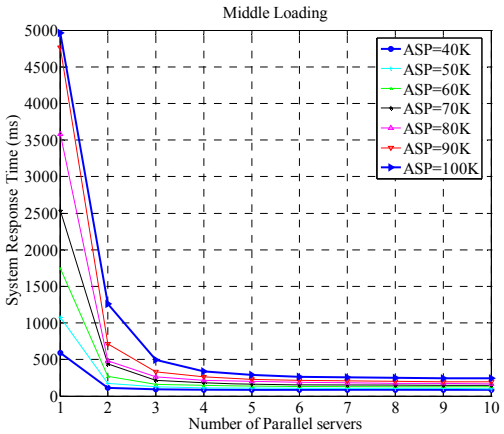

Fig. 14 With Middle Loading of parallel connection servers

Fig. 15 shows the variation in the system response time of a Heavy Loading, with the system response time of the server at 13776-83978 ms. When two servers are parallel, the system response

time is immediately reduced to 5241-41522 ms and descends to about 50% of the biggest possible range. Ten parallel system response times of servers are about 514.9-8623 ms.

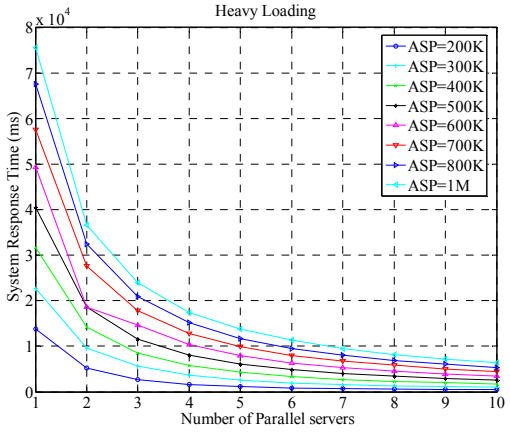

Fig. 15 With Heavy Loading of parallel connection servers

Fig. 16 shows all measurement data, the system response time uses the Y axis logarithm to express the characteristics of the response time. In Fig. 16, by utilizing a Light Loading, the parallel connection can't reduce the system response time by much. With a Middle Loading the parallel connection reduces the system response time by up to four times. If the number of parallel connections increases, the system response time can be reduced to 1/2n - $1/n^2$. With a Heavy Loading the system response time is decreased by about 1/n of a single server.

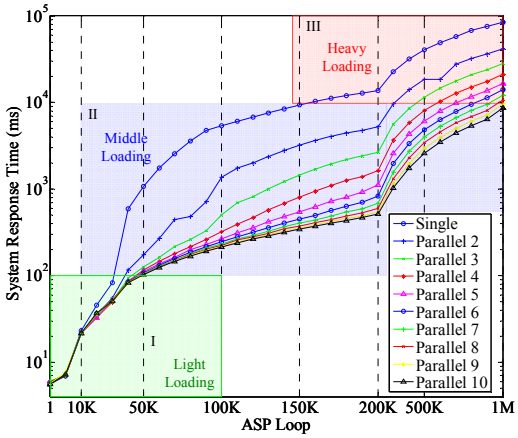

Fig. 16 The system response time is in various regions of the service rates

The key factors that influence the system response time in queue theory are: arrival distribution with the arrival rate λ, service distribution with the service rate μ and the buffer size. When a packet arrives, it stays in buffer and waits for a server. The system response time comprises the waiting time and the service time of the buffer.

With a Light Loading the system response time tends to be near the service time, and if there is no packet waiting in the buffer, no matter how many parallel servers there are available, the system response time varies little. This response time is comprised of the network access time and the DNS time. We use the following equation:

Light Loading, $\mu >> \lambda$, a customer waiting time of less than 0.1 sec

System response time $E_{P_1} = \frac{1}{\mu_1 - \lambda_1}$

System response time of parallel connections

$$E_{eq_n}(T) = P_1 E_{P_1}(T) + P_2 E_{P_2}(T) + \ldots + P_n E_{P_n}(T) \quad (1)$$

$$P_1 + P_2 + \ldots + P_n = 1$$

With a Middle Loading a large number of packets is waiting in the buffer but is unlikely to fill it up. Therefore by adding a server we increase the service rate by n times and reduce the number of packets in the buffer by 1/n times. Thus we reduce the system response time by $1/n^2$ times. When the number of parallel connection servers increases, the number of packets in the buffer is reduced by 1/2, 2/3, 3/4 … , the least of these packets being 1/2 B, and the system response time is gradually reduced from a Middle Loading to a Light Loading. It no longer has the $1/n^2$ efficiency. Our equation can be explained as follows:

Middle Loading, $\mu \approx \lambda$, packet waiting time is 0.1-10 secs $E_{P_1} = \frac{B_1}{2\mu_1}$

If the parallel connection is n, $\mu_1 = \mu_2 = \ldots = \mu_n$, $B_1 = B_2 = \ldots = B_n$

Buffer size for single server is 1

Service rate $\mu_{eq_n} = n\mu_1$, if the buffer was not filled up, so $B_{eq_n} = \frac{1}{n}B_1 \ldots \frac{1}{2}B_1 \ldots \frac{n}{n+1}B_1$

System response time just before a Middle Loading is

$$E_{eq_n}(T) = \frac{B_{eq_n}}{2\mu_{eq_n}} = \frac{\frac{1}{n}B_1}{2n\mu_1} = \frac{B}{2n^2\mu} = \frac{1}{n^2}E_{P_1} \quad (2)$$

After a Middle Loading

$$E_{eq_n}(T) = \frac{B_{eq_n}}{2\mu_{eq_n}} = \frac{\frac{1}{2}B_1}{2n\mu_1} = \frac{B}{4n\mu} = \frac{1}{2n}E_{P_1} \quad (3)$$

Without a Heavy Loading, if the parallel connection number is n, if the buffer is full the service rate increases n times. Therefore the total time is reduced by only 1/n times. Our equation is as follows:

Heavy Loading, $\mu \ll \lambda$, customer waiting time greater than 10 secs. $E_{P_1} = \frac{B_1}{\mu_1}$

If the parallel connection number is n, $\mu_1 = \mu_2 = ... = \mu_n$, $B_1 = B_2 = ... = B_n$

Service rate $\mu_{eq_n} = n\mu_1$, but the buffer is still full, so $B_{eq_n} = B_1$.

The system response time of a Heavy Loading is

$$E_{eq_n}(T) = \frac{B_{eq_n}}{\mu_{eq_n}} = \frac{B_1}{n\mu_1} = \frac{1}{n}E_{P_1}. \quad (4)$$

In addition to Light Loading, Middle Loading and Heavy Loading, the system response time of the best efficiency is shown in Fig. 17. The ratio of the system response time is $\frac{E_n(T)}{E_1(T)}$ of the servers. In the diagram, with ten parallel connection servers, ASP=180K reduces the system response time by 27.8 times. With a Heavy Loading, ASP=1M, the highest, the system response time is only reduced by 10 times, and with a Light Loading, ASP=40K, the system response time is reduced by only 1.5 times.

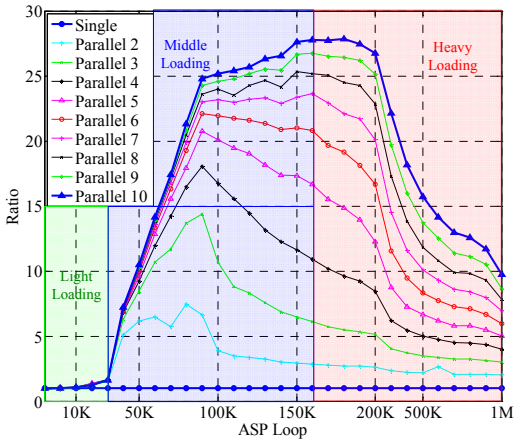

Fig. 17 System response time and service rate of parallel connection servers

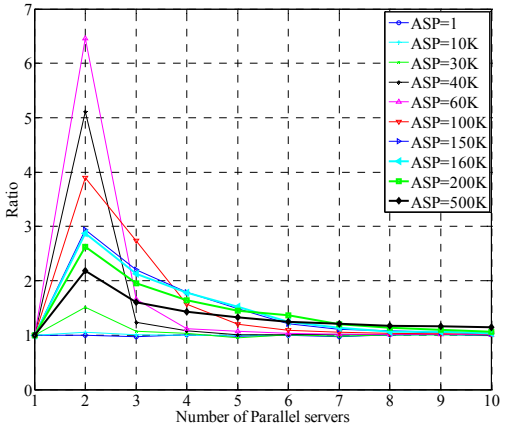

Fig. 18 System response time and number of parallel connection servers

If every server has the same service rate, then two parallel servers have the best performance. Fig. 18 shows the ratio of the system response time, $\frac{E_n(T)}{E_{n+1}(T)}$. In this diagram, at ASP=60K the system response times with parallel connection servers are 6.46 times that of a single server. If we compare the parallel performance of two and three parallel connection servers, ASP=100K is the highest, 2.74 times, and the lowest, of five and six parallel servers is ASP=20K, only 0.89 times. Ten parallel connection servers result in an average of about 1.4 times.

5 Conclusion

When we want to reduce the system response time, parallel connection servers can be useful with a Middle Loading and a Heavy Loading. In Fig. 16, with a Light Loading and a system response time of 100 ms, the parallel connection servers can't reduce the system response time by very much. But with a Middle Loading, where the system response time is 100 ms to 10 secs, the parallel connection servers significantly reduce it up to the $1/n^2$. If the number of parallel connection servers increases with a Middle Loading, and the buffer is full, this situation will not result in a linear decrease, because the service rate still can't empty the buffer, and the system response time inclines to $n/(n+1)$ by $1/n^2$. We therefore, can't immediately change to a Light Loading, because the efficiency of the system response time gradually declines to 1/2n. With a Heavy Loading the system response time is higher than 10 secs, and the parallel connection servers just present the inverse ratio of 1/n. Also, in Fig. 18, with two parallel servers, the efficiency of the system response time is the highest possible, when based on a balance sharing.

With a Middle Loading and a Heavy Loading the parallel queue improves the system response time, but with a Light Loading it doesn't. As we acquire an understanding of the many characteristics of parallel connection servers, we shall in the future investigate more complicated connection servers to predict the characteristics of their system response times.

References:

[1] Ying-Wen Bai, Chia-Yu Chen, and Yu-Nien Yang, "A Two-Pass Web Document Allocation Method for Load Balance in Multiple Grouping of a Web Cluster System," *ICON'04, 12th IEEE International Conference on Networks,* Vol. 1, 2004, pp. 177-181.

[2] Nathan L. Binkert, Lisa R. Hsu, Ali G. Saidi, "Performance Analysis of System Overheads in

TCP/IP Workloads," *Proceedings of the 14th International Conference on Parallel Architectures and compilation Techniques,* 17-21 Sept., , 2005, pp.218-228.

[3] Zhen Zhao, Bryan Willman, Steven Weber and Jaudelice C.de Oliveira, "Performance analysis of a parallel link network with preemption," *Proceedings of International Conference on Information Sciences and Systems,* 22-24 March, 2006, pp.271-276.

[4] S.C. Tan, F.M. Abbou, and H.T. Ewe, "Selective assign shortest path first algorithm for routing and wavelength assignment in the presence of four wave mixing" *Published in IET Communications*, 2009, Vol. 3, Iss. 7, pp. 1097–1102.

[5] Martin M. Peritsky, "Traffic Engineering of Combined Mobile Telephone and Dispatch Systems." *IEEE Transactions on Communications*, Vol, COM-21, NO. 11, November 1973, pp.1307-1309.

[6] Martin M. Peritsky, "Traffic engineering of combined mobile telephone and dispatch systems." *IEEE Transactions on Vehicular Technology*, Vol. VT-22, NO. 4, November 1973, pp.223-225.

[7] Chin-Tau Lea and Anwar Alyatama, "Bandwidth quantization and states reduction in the broadband ISDN," *IEEE/ACM Transactions on Networking*, Vol. 3, NO. *3,* June 1995, pp. 352-360

[8] Arash Abadpour, Attahiru Sule Alfa, and Anthony C. K. Soong, "Closed Form Solution for Maximizing the Sum Capacity of Reverse-Link CDMA System with Rate Constraints," *IEEE Transactions on Wireless Communications*, Vol. 7, NO. 4, April 2008, pp.1179-1183

[9] Mainak Chatterjee, Haitao Lin, and Sajal K. Das, "Rate Allocation and Admission Control for Differentiated Services in CDMA Data Networks," *IEEE Transactions on Mobile Computing*, Vol. 6, NO. 2, February 2007, pp. 179-191.

[10] Anastasios Stamoulis, Nicholas D. Sidiropoulos, and Georgios B. Giannakis , "Time-varying fair queueing scheduling for multicode CDMA based on dynamic programming," *IEEE Transactions on Wireless Communications*, Vol. 3, NO. 2, March 2004, pp. 512-523.

[11] Sem Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks." *IEEE/ACM Transactions on Networking*, Vpl. 13, NO. 3, June 2005, pp. 636-647.

[12] Rong Pan, Balaji Prabhakar, Konstantinos Psounis, and Damon Wischik, "SHRiNK: a method for enabling scaleable performance prediction and efficient network simulation." *IEEE/ACM Transactions on Networking,* Vol. 13, NO. 5, October 2005, pp. 975-988.

[13] Shufen Liu, Lu Han, Xinjia Zhang, and Kai Nie, "Study of network performance measurement based on SNMP." *Proceedings of the The 8th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2004)*, Xiamen, China, May 26-28, 2004, pp.119-123.

[14] R. Les Cottrell, Connie Logg, Mahesh Chhaparia, Maxim Grigoriev, Felipe Haro, Fawad Nazir, Mark Sandford, "Evaluation of Techniques to Detect Significant Network Performance Problems using End-to-End Active Network Measurements," *Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006),* Vancouver Convention and Exhibition Center, Vancouver, Canada. April 3-7, 2006, pp. 85-94.

[15] Alexandru Bikfalvi, Paul Patras, Cristian Mihai Vancea, Virgil Dobrota, "The Management Infrastructure of a Network Measurement System for QoS Parameters." *Proceedings of 2006 International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2006),* Dubrovnik, Croatia, September 29 - October 1, 2006, pp.1-5.

[16] Ying-Wen Bai and Yu-Nien Yang, "An Approximate Performance Analysis and Measurement of the Equivalent Model of Parallel Queues for a Web Cluster with a Low Rejection," *Proceedings of the 14th IEEE International Conference,* 2006, pp.1-6.

[17] Chung-Ping Chen, Ying-Wen Bai and Yin-Sheng Lee, "Approximate Analysis and Measurement of Equivalent Model for Tree Connection of Web Server Systems," *Proceedings of the 19th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2007),* Cambridge, MA, USA, Nov. 19-21, 2007, pp.91-96.

[18] http://www.paessler.com/webstress/

[19] Keith W. Ross, James F. Kurose, "*Computer Networking: A Top-Down Approach Featuring the Internet*" Reading, MA : Addison-Wesley, 2001.

[20] Gunter Bolch, Stefan Greiner, Hermann de Meer, and Kishor S. Trivedi, "*Queueing Networks and Markov Chains*" Wiley-Interscience, 1998.

[21] H. T. Kaur, D. Manjunath, and S. K. Bose, "The Queueing Network Analysis Tool (QNAT),"

Proceedings of International Symposium on Modeling, Analysis and simulation of Computer and Telecommunication Systems, 2000, pp. 341-347.

[22] Chung-Ping Chen, Ying-Wen Bai and Yin-Sheng Lee, "Performance Measurement and Queueing Analysis with Low Blocking Probability of Serial Connection Networks," *Proceedings of the I²MTC 2008 – IEEE International Instrumentation and Measurement Technology Conference,* Victoria, Vancouver Island, Canada, May 12-15, 2008, pp.2216-2221.