Timing Performance Control in Web Server Systems Utilizing Server Internal State Information

Xue Liu, Rong Zheng, Jin Heo, Qixin Wang, Lui Sha
{xueliu, zheng4, jinheo, qwang4, lrs}@cs.uiuc.edu
Department of Computer Science, Univ. of Illinois at Urbana-Champaign

Abstract

How to effectively allocate system resource to meet the Service Level Agreement (SLA) of Web servers is a challenging problem. In this paper, we propose an improved scheme for autonomous timing performance control in Web servers under highly dynamic traffic loads. We devise a novel delay regulation technique called Queue Length Model Based Feedback Control utilizing server internal state information to reduce response time variance in presence of bursty traffic. Both simulation and experimental studies using synthesized workloads and real-world Web traces demonstrate the effectiveness of the proposed approach.

1. Introduction

Web server systems have become an integral part of our society. Web hosting companies often times host multiple client classes on a single Web server. One major problem that these companies face is how to meet the Service Level Agreements (SLAs) [14] with their clients without excessively over-provisioning resources. SLAs are usually expressed in the form of the maximum average response time guarantee, above which is not acceptable to the clients. From clients' perspective, response time in general consisted of two parts: the queueing and processing delay incurred at the server system, and the network delay to transport requests and server responses over the Internet. With advancement of the fiber optic and Web caching technologies, the latter is usually small compared to the service delay experienced in the Web server systems at the network edges. Therefore, it is of great theoretical and practical interests to provide delay service guarantees to clients in Web server systems.

Feedback control is an important technology for controlling mechanical systems. Recently, it has been applied in computing systems for performance control [1], [11]. However, how to provide smooth performance control over a wide range of workload conditions remains to be a challenging problem. In real systems, Web workloads are stochastic with parameters often varying significantly over time. Furthermore, a Web server's response to allocated resources is highly nonlinear. As a consequence, differential equation models used in classical control theory do not work well for Web server systems. On the other hand, Web servers are intrinsically queueing systems. How to leverage the power of queueing model and control theory for performance control draws great research interests.

Queueing Model Based Feedback Control architecture is proposed in [18]. At the core of the design is a model based feed forward predictor, which keeps the system state near an equilibrium operation point in presence of dynamic workloads. This essentially linearizes the system. Combined with a feedback PI controller to suppress the "residual errors", the Queueing Model Based Feedback Control architecture is shown to provide good mean delay regulation for Web server under single client classes.

However, Web traffic is known to be bursty and exhibits self-similar properties [5]. We observe that the performance of Queueing Model Based Feedback Control deteriorates in presence of bursty traffic. In this paper, we devise a new control approach, Queue Length Model Based Feedback Control to reduce the response time variance. It utilizes the server internal state information of instantaneous queue length that allows better handling of the transient behaviors caused by rapidly changing traffic loads.

The rest of the paper is organized as follows. Section 2 provides a brief review of related research work. In Section 3, we identify the problem of previous Queueing Model Based Feedback Control. In Section 4, the new timing performance control approach -- *Queue Length Model Based Feedback Control* is proposed. In Section 5 and 6, we evaluate the performance of the proposed scheme through simulation and experimental studies using both

synthetic and trace based workloads. Finally, we conclude the paper in Section 7.

2. Related Work

Feedback control theory was invented more than 50 years ago; since then many powerful tools and results have been obtained and deployed [8]. Recently, control-theoretic approaches have been applied to server performance control. In [1], Abdelzaher *et al*. build a feedback control loop for Apache Web server that enforces desired relative delays among different service classes via dynamic connection scheduling and process reallocation. In [12], a similar approach is used for Squid proxy server to guarantee cache hit-ratio by dynamically adjusting the disk space allocation. In [11], the parameters of an Apache Web server are dynamically allocated using a MIMO feedback controller. The goal is to keep the system's CPU and memory utilization stabilized at a desired reference value. These approaches view the server system as either a linear transfer function or a state space model and use linear feedback control scheme. Due to the fundamental difference between a mechanical system and computing system, these models cannot be built directly to reflect the internal dynamics of the server. Instead, they are usually constructed offline using model identification techniques under certain predefined workloads [10]. Due to the stochastic nature of the Web traffic (with temporally and spatially variations), models thus obtained are not accurate. Furthermore, since computing systems are highly nonlinear, these models are at best a linear approximation of the real system. This leads to the problem of poor robustness when directly applying classical control theory to controlling server's performance, especially in presence of dynamic traffic loads. To solve this problem, Lu et al. [13] propose an adaptive control technique for Web caching systems. The introduction of adaptive controller helps adjusting the model to the traffic dynamics to some extent. However, the model and control are still intrinsically linear; and the adaptive controller usually responses slowly to abrupt traffic changes.

In [18], Queueing Model Based Feedback Control is proposed to achieve better delay regulation in Web servers. It consists of a feed forward predictor and a feedback controller. The feed forward predictor outputs a service rate allocation □q as predicted by a queueing-theoretical model. A feedback controller is introduced to further reduce the "residual" error. The Queueing Model Based Feedback Control works well under moderate workloads. However, as will be illustrated in Section 4.1, we observe that its performance deteriorates when the traffic becomes bursty. This motivates us to propose a new approach to handle such dynamic workloads and provide better performance control by utilizing queue length information.

3. Problems with Queueing Model Based Feedback Control

In this section, we motivate our proposed delay regulator by examining the deficiency of existing Queueing Model Based Feedback Control in handling bursty traffic loads.

Studies reveal that the Web traffic is bursty and exhibits self-similar properties [5]. It has been shown the burstiness of Web request traffic can be modeled using the Pareto On/Off distribution. In this model, packets are sent at a fixed rate during On periods, and no packets are sent during Off periods.

To evaluate the performance of different control approaches, we developed a Web server simulation package using the network simulator *ns-2* [17]. In the simulation, there are three adjustable parameters to control the "burstiness" of the Pareto On/Off traffic. The parameter *Burst_Time* corresponds to the mean length of On period of the traffic. The parameter *Idle_Time* corresponds to the mean length of the Off period and *Interval* is the interarrival time during On periods. In all the simulations, the Pareto shape parameter is set to 1.5 as evidenced from real traffic measurements [17].

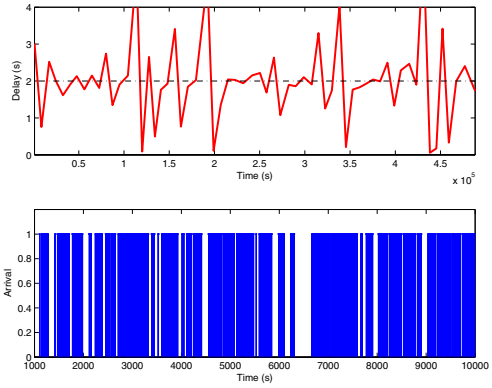

Figure 1: Performance of Queueing Model Based Feedback Control under Pareto On/Off Source

Figure 1 shows the delay experienced by a single client connection with *Burst_Time=1*, *Idle_Time =10* and *Interval=0.1* under the original Queueing Model Based Feedback Control. The reference delay is set to $D^{ref}=2$. Each point in the upper figure is an average of response times experienced by *1000* requests. The

corresponding mean and variance of the response time are *2.0041* and *1.2034* respectively.

From Figure 1, we observe that the delay fluctuates a lot around the reference value although the long-term average mean delay is close to D^{ref}. This is because the online estimation of the request rate λ is a time average of the instantaneous request rate (on the order of 500 requests in our implementation). For bursty traffic, the instantaneous request rate is larger than the long-term average rate λ during On periods. However, the feed forward predictor's output rate is based on λ and thus is lower than the instantaneous request rate. Therefore, the request queue will build up and clients will experience longer response time. On the other hand, during periods with sporadic requests, the instantaneous request rate is smaller than λ which leads to smaller response time. This observation motivates us to consider the use of server internal state information in the controller design to suppress large delay variations.

4. Queue Length Model Based Feedback Control

To have a better control of the transient behavior, we utilize queue length information in our new controller design. In fact, queue length is closely related to the delay of a request, which equals to the sum of service times of all requests queued ahead of it and the service time of its own.

First, we introduce a new *Queue Length Model Based Predictor* with an additional feedback term, i.e., the queue length measurements of server in the prediction of service rate allocation μ_q. The procedure of *Queue Length Model Based Predictor* is as follows:

Step 1: At each control invocation, we measure the current queue length $l_{current}$ and update the request rate estimate λ.

Step 2: Based on results from queueing theory, a targeted queue length $l_{targeted}$ is computed. For example, if *M/M/1*(or *G/G/1*) model is used to model the server, we have $l_{targeted} = \lambda D^{ref}$. The term $l_{targeted}$ gives the desired queue length in steady state under the current mean request rate λ and targeted delay reference.

Step 3: Let $\mu_q = (\frac{1}{D^{ref}} + \lambda) + K \times (l_{current} - l_{targeted})$ to be the new model output service rate. The first term $(\frac{1}{D^{ref}} + \lambda)$ is the same as the Queueing Model Predictor in the original approach. The second term $K \times (l_{current} - l_{targeted})$ represents the queue length feedback. K is a constant control gain, in practice, we can set $K = 1/D^{ref}$.

In essence, this new queue length based feed forward predictor adjusts the estimated service rate μ_q not only based on the request rate estimation but also

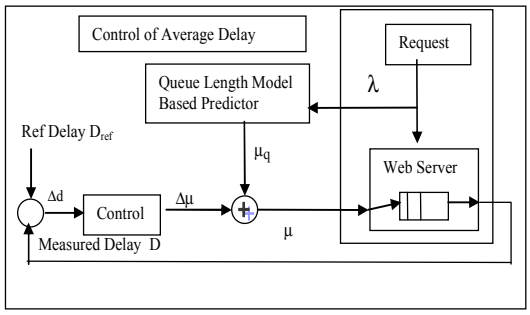

Figure 2: Queue Length Model Based Feedback Control Architecture

on the degree of server queue built-up. From discussions in Section 3, if the request rate in the current control interval is larger than the mean request rate estimation λ, we have $l_{current} > l_{targeted}$. Therefore, the second term is positive in Step 3, which helps to clear up the queue and thus reduces the client experienced response time.

The proposed *Queue Length Model Based Feedback Control* regulator is shown in Figure 2. In this architecture, the Queue Length Model Based Predictor outputs a service rate allocation μ_q based on the online measurement of request rates, reference delay D^{ref} and current queue length $l_{current}$. The feedback controller calculates a service rate adjustment $\Delta\mu$ according to the difference between delay reference D^{ref} and measured delay d in each control interval. Lastly, the sum of μ_q and $\Delta\mu$, i.e, μ is used to determine the system resource quota to be applied to the server.

5. Effect of Queue Length Model Based Feedback Control

We implemented the Queue Length Model Based Feedback Control in our simulation package. To verify the design of the proposed controller, we experiment with the same Pareto On/Off traffic (with parameters *Burst_Time* = 1, *Idle_Time* = 10 and *Interval* = 0.1) as in Section 3. Figure 3 demonstrates the controlled server performance using the new Queue Length Model Based Feedback Control. The variance of the client response time reduces to 0.0051 as compared with 1.2034 using the previous Queueing Model Based Feedback Control approach. In addition, the average response time of both approaches are very close to D^{ref}

= 2. We see using the new controller, even with this extremely bursty traffic, good delay regulation can be achieved.

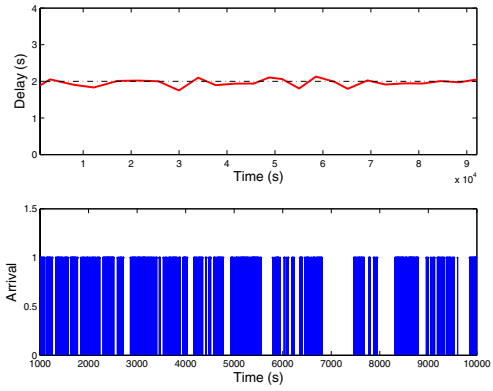

Figure 3: Performance of Queue Length Model Based Feedback Control under Pareto On/Off Source

6. Experimental Evaluation

We have implemented Queue Length Model Based Feedback Control on Apache web server 2.0.7 with Linux kernel 2.4.20. In our implementation, Apache and Linux kernel is modified to provide state information used in the control system.

6.1. Implementation

The web server is instrumented with three additional components, i.e., a Monitor, a Controller and an Actuator to implement the proposed control scheme. The Monitor is responsible for collecting state information (measurements) at the server and sends them to the Controller module. The Monitor processes and stores the state information and then invokes the controller, which in turn determines the next service rate to be applied to the server. Two types of controllers are implemented, i.e., Queueing Model Based Feedback controller and Queue Length Model Based Feedback controller. To isolate the effect of different controller designs, we did not apply the PI controller shown in Figure 3. We demonstrate that, even without PI controller, the performance of Queue Length Model Based Feedback achieves significant performance improvement compared to the Queueing Model Based Feedback Control.

6.2. Experimental Results

All experiments are conducted on a test bed consisted of two PCs connected through 100Mbps Ethernet. The client machine is equipped with a 1.7GHZ Intel Pentium IV processor and 512MB RAM. httperf [16] is used as synthetic generator on the client. We modified httperf to generate realistic workload from real Apache access log (i.e. web traces explained below). The server machine has a 333MHZ Intel Pentium II processor and 256MB RAM, which runs Apache 2.0.7 and Linux kernel 2.4.20.

Experiments are performed using the World Cup 98 Web trace [2]. For the World Cup 98 trace, inter-arrival time is calculated between consecutive requests and scaled by different scale values among 1.0, 0.5, 0.3, 0.2, such that

Adjusted_Interarrival_Time
*= Interarrival_Time_from_the_Trace * Scale_Value*

In all experiments, the reference delay is set to 0.1 (seconds).

We compare the performance of Queueing Model Based Feedback Control and Queue Length Model Based Feedback Control under the day 6 of the world cup trace series. The mean and variance of delay measurements are summarized in Table 1.

From Table 1, we observe that the proposed Queue Length Model Based Feedback Control can regulate the delay very well. The response time is very close to reference delay 0.1. Furthermore, the variance of response time is small. In comparison, without PI controller, the average response times of Queueing Model Based Feedback Control are greatly affected by the burstiness and load of the traffic. When traffic is light, average response time is lower than reference delay. When the traffic is more heavy and bursty, the average response time exceeds the reference delay. For almost all traffic loads, the variance of response time incurred by the Queue Length Model Based Feedback Control is at least an order of magnitude less than that of Queueing Model Based Feedback Control.

7. Conclusions

We proposed a new Web server timing control scheme called Queue Length Model Based Feedback Control. Compared with previous approaches, the new scheme can significantly reduce response time variance under a wide range of workload conditions including bursty traffic. This is achieved by utilizing the server internal queue length measurements. Extensive simulation study shows that the new scheme can provide smooth performance control and better track SLA specifications in Web server systems.

Table 1: Performance Comparison Using World Cup 98 Traces

Inter-arrival x	Queueing Model Based		Queue Length Model Based	
	Mean(RT)	Var(RT)	Mean(RT)	Var(RT)
1.0	0.049883 *s*	0.000390	0.100835 *s*	0.000750
0.5	0.049777 s	0.005358	0.101715 s	0.000475
0.3	0.089887s	0.005249	0.105355 s	0.000274
0.2	1.149359 s	0.285477	0.109762 s	0.000337

8. Acknowledgments

This research is sponsored in part by NSF CCR 02-09202, by ONR N00014-02-1-0102, and by MURI N00014-01-0576.

9. References

[1] T. F. Abdelzaher, C. Lu, "Modeling and Performance Control of Internet Servers, Invited Paper", *39th IEEE Conference on Decision and Control*, Sydney, Australia, December 2000

[2] M. Arlitt and T. Jin, *1998 World Cup Web Site Access Logs*, August 1998. Available at http://www.acm.org/sigcomm/ITA/

[3] G. Banga, P. Druschel, and J. C. Mogul, "Resource Containers: A new Facility for Resource Management in Server Systems", In *Third USENIX Symposium on Operating Systems Design and Implementation*, New Orleans, Louisiana, February 1999

[4] D. Blumenfeld, *Operations Research Calculations Handbook*, CRC press, June, 2001

[5] M. Crovella, A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Cause", *Proceedings of SIGMETRICS* 1996

[6] N. Gandhi, S. Parekh, J. Hellerstein, D.M. Tilbury, "Feedback Control of a Lotus Notes Server: Modeling and Control Design", *American Control Conference*, 2001

[7] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*, Wiley-Interscience, New York, NY, April 1991

[8] K. J. Astrom, B. Wittenmark, *Computer-Controlled Systems: Theory and Design*, 3rd edition, Prentice Hall, November, 1996

[9] R. Kumar, K. Juvva, A. Molano, and S. Oikawa, "Resource Kernels: A Resource-centric Approach to Real-Sime systems", In *Proc. of the SPIE/ACM Conference on Multimedia Computing and Networking*, Jan. 1998

[10] L. Ljung, *System Identification Toolbox* , Mathworks, April, 2001

[11] Y Diao, N. Gandhi, J. Hellerstein, S Parekh, D. M. Tilbury, "Using MIMO Feedback Control to Enforce Policies for Interrelated Metrics With Application to the Apache Web Server", Network Operations and Management, 2002

[12] Y. Lu, A. Saxena, T. F. Abdelzaher, "Differentiated Caching Services: A Control-Theoretical Approach", *International Conference on Distributed Computing Systems*, Phoenix, Arizona, April 2001

[13] Y. Lu, C. Lu, T. Abdelzaher, G. Tao, "An Adaptive Control Framework for QoS Guarantees and its Application to Differentiated Caching Services", IWQoS, Miami Beach, FL, May 2002

[14] D. Menasce, V. Almeida, *Capacity Planning for Web Services: Metrics, Models, and Methods*, Prentice Hall, 2001

[15] C. Mercer, S. Savage, and H. Tokuda, "Processor Capacity Reserves: Operating System Support for Multimedia Applications", In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems,* pages 90-99, May 1994

[16] D. Mosberger and T. Jin, "httperf—A Tool for MeasuringWeb Server Performance", In *Proceedings of WISP '98*, Madison, Wisconsin, USA, June 1998

[17] The Network Simulator - *ns-2*. Available at http://www.isi.edu/nsnam/ns/

[18] L. Sha, X. Liu, Y. Lu, T. Abdelzaher, "Queueing Model Based Network Server Performance Control", *IEEE Real-Time Systems Symposium*, Phoenix, Texas, December, 2002