An Integrated Approach to Robust Proportional Responsiveness Differentiation *

Authors
Department of Computer Science
University of Colorado at Colorado Springs, Colorado Springs, CO 80933
{...}@cs.uccs.edu

Abstract

There is an increasing demand of providing proportional response time differentiation to various clients on Web servers. According to the foundations of queueing theory, the objective can be achieved by providing different processing rates to the client on the servers. At application level, process is often used as the resource allocation principal for achieving processing rates. However, an implementation of the approach has shown weak proportionality with large variance because it does not have control over the consumption of resources that the kernel consumes. In this paper, we integrate a feedback controller with the queueing-theoretical approach. The integrated approach allocates the certain number of processes to handle requests of different client classes according to the queueing-theoretical processing rate allocation scheme. The process allocations are then adjusted according to the difference between the target response time and the achieved one by using the proportional integral derivative control. We implement the integrated approach on Apache Web servers and the experimental results demonstrate that this application-level approach can enable Web servers to provide robust proportional responsive time differentiation.

1 Introduction

Due to the open and dynamics nature of Web applications, the last decade has witnessed an increasing demand for provisioning of different levels of quality of service (QoS) to meet changing system configuration and resource availability and satisfy different client requirements. This differentiated QoS provisioning problem was first formulated by the Internet Engineering Task Force in the network core. Differentiated Services (DiffServ) [6] is a major architecture, where the network traffic is divided into a number of classes. It aims to define configurable types of packet forwarding in network core routers, which can provide per-hop differentiated services for per-class aggregates of network traffic. The proportional differentiation model [9] is one of the most popular models. It states that certain class performance metrics should be proportional to their pre-specified differentiation weights, independent of the class loads. Due to its inherent differentiation predictability and proportionality fairness, the model has been accepted as an important DiffServ model and been applied in the proportional queueing-delay differentiation (PDD) in packet scheduling [9, 10, 15, 17] and proportional loss differentiation in packet dropping [12].

End-to-end service differentiation requires stateless packet scheduling in the core routers, in combination with stateful resource management in the network edges and end servers. There are recent efforts on DiffServ provisioning on servers [1, 2, 5, 7, 8, 14, 19, 20, 21]. In the server side, response time is a fundamental performance metric. Existing response time differentiation strategies are mostly based on priority scheduling in combination with admission control and content adaptation [1, 2, 5, 7, 8]. The authors in [8] adopted strict priority scheduling strategies to achieve responsiveness differentiation on Internet servers. The results showed that the differentiation can be achieved with requests of higher priority classes receiving lower response time than requests of lower priority classes. However, this kind of strategies cannot control the quality spacings proportionally among different classes. Time-dependent priority scheduling algorithms developed for PDD provisioning in packet networks can be tailored for PDD provisioning on Web servers [14]. However, they are not applicable for response time differentiation because the response time is not only dependent on a job's queueing delay but also on its service time, which varies significantly depending on the requested services.

In [19, 20], we proposed queueing-theoretical processing rate allocation strategies for server-side DiffServ provisioning with respect to slowdown, the ratio of a request's queueing delay to its service time. While simulation results match expectations, a challenging implementation issue is, how

*This research work was supported in part by a NISSC AFOSR Grant subaward.

to practically achieve the processing rate for various traffic classes on servers. This is not a trivial problem. In [18], we presented a processing rate allocation scheme based on queueing theory for proportional response time differentiation on Web servers. We then designed and implemented an adaptive process allocation strategy to achieve the processing rates allocated to the request classes. Figure **??** shows the experimental results. When the system load is between 30% to 80%, the expected differentiation ratios are achieved in average. However, the proportionality comes along with large variance. When the load goes up to 90%, the expected ratio is not achieved even in average. For example... There are three reasons:....

In the current general-purpose operating systems, a process, or a thread within a process, is treated as the scheduling entity for an independent activity. It is also the entity for the allocation of resources, such as CPU cycles and memory space. Thus, process abstraction serves both as a protection domain and as a resource principal. However, in the operating systems, resource allocation and scheduling primitives do not extend to the execution of significant parts of kernel code. An application has no control over the consumption of resources that the kernel consumes on behalf of the application. As the result, resource principals do not always coincide with either processes or threads. For example, in a network-intensive application, the process is the correct unit for application isolation, but it does not encompass all of the associated resource consumption since the kernel generally does not control or properly account for resources consumed during the processing of network traffic. Because of the coincidence between protection domain and resource principal, applications lack sufficient control over resource scheduling and management on the server. This problem makes it difficult to enforce application-level process allocation strategies for proportional response time differentiation on Web servers.

There are efforts on the design of new resource management mechanisms at kernel level to support DiffServ provisioning efficiently. *Resource container* is a new operating system abstraction [4]. It separates the notion of a protection domain from that of a resource principal. A resource container encompasses all system resources that the server uses to perform an independent activity, such as processing a client HTTP request. All user and kernel level processing for an activity is charged to the appropriate resource container and scheduled at the priority of the container. Resource containers allow accurate accounting and scheduling of resources consumed on behalf of a single client request or a class of client requests. Thus, this new mechanism can provide fine-grained resource management for DiffServ provisioning when combined with an appropriate resource scheduler. However, while kernel-level mechanisms can provide efficient control over resource management, their weaknesses lie on the portability and deployment issues.

In this paper, we seek a practical application-level approach to providing robust proportional responsiveness differentiation. We design an integrated approach based on queueing theory and feedback control theory. The structure of the paper is as follows. Section **??** gives the processing rate allocation scheme for response time differentiation. Section **??** presents the design and implementation of the adaptive process allocation on Apache Web servers. Section **??** focuses on experimental results and performance evaluation. In Section 3, we review other related resource allocation and scheduling disciplines in the DiffServ areas. Section **??** concludes the paper.

2 Integrated Process Allocation with Feedback Control

A proportional differentiation model is to ensure the pre-specified QoS ratios between N ($N > 1$) classified traffic classes. The proportional responsive time differentiation model aims to control the ratios of the average response time of classes based on their pre-specified differentiation parameters $\{\delta_i, i = 1, \ldots, N\}$. Let $T_i(k)$ denote the average response time of requests of class i at sampling period k. Specifically, the model requires that the ratio of average response time between class i and j is fixed to the ratio of the corresponding differentiation parameters

$$\frac{T_i(k)}{T_j(k)} = \frac{\delta_i}{\delta_j} \qquad 1 \leq i, j \leq N. \tag{1}$$

There are three requirements of service differentiation provisioning.

1. *predictability*: higher classes should receive better or no worse service quality than lower classes, independent of the class load distributions.

2. *Controllability*: the system should have a number of controllable parameters that are adjustable for the control of quality spacings among classes.

3. *fairness*: requests from lower classes should not be over-compromised for requests from higher classes.

The proportional model essentially has the proportionality fairness. According to the requirement of the differentiation predictability, the higher classes should receive better service, i.e., lower response time. Without loss of generality, we assume that class 1 is the 'highest class' and set $0 < \delta_1 < \delta_2 < \ldots < \delta_N$.

Like others in [16, 21], we use Poisson process arrivals and exponentially distributed service times (an $M/M/1$ FCFS queue) for modeling the traffic. We note that there are other popular heavy-tailed distributions, such as Bounded

Pareto, for service time distributions [3, 20]. The processing rate allocation scheme derived by the M/M/1 queueing model can give the key insights about the differentiation problem and the feasibility of the process allocation strategy.

We divide the request processing rate of a Web server into N virtual servers. Each virtual server handles requests of one class in a FCFS manner. Let $\mu_i, 1 \leq i \leq N$ denote the normalized request processing rate of the virtual server i. We have

$$\sum_{i=1}^{N} \mu_i = 1. \tag{2}$$

Assume requests of class i in Poisson process arrive at virtual server i in a rate λ_i. It follows that the traffic intensity on the server $\rho_i = \lambda_i/\mu_i$. According to the foundations of queueing theory [13], when $\rho_i < 1$ ($\lambda_i < \mu_i$), we have the expected response time of requests in class i as

$$T_i = \frac{\rho_i}{\mu_i(1-\rho_i)} = \frac{1}{\mu_i - \lambda_i} \qquad 1 \leq i \leq N. \tag{3}$$

For feasible processing rate allocation, we must ensure that the system utilization $\sum_{i=1}^{N} \rho_i \leq 1$. That is, the total processing requirement of the N classes of traffic is less than the Web server's processing capacity. Otherwise, a request's response time can be infinite and responsiveness differentiation would be infeasible. Admission control mechanisms can be applied to drop requests from lower classes so that the constraint holds [8].

According to the definition of (3), the set of (1) in combination with (2) lead to

$$\mu_i = \lambda_i + \frac{1 - \sum_{i=1}^{N} \lambda_i}{\delta_i \sum_{i=1}^{N} 1/\delta_i}. \tag{4}$$

From this equation, we can observe that the remaining capacity of the server is fairly allocated to different request classes with respect to their differentiation parameters.

It follows that the expected response time of requests of class i, T_i, is calculated as:

$$T_i = \frac{\delta_i \sum_{i=1}^{N} 1/\delta_i}{1 - \sum_{i=1}^{N} \lambda_i}. \tag{5}$$

2.1 Adaptive Process Allocation

2.2 Feedback Control

...

3 Related Work

The proportional differentiation model was proposed in the network core [9]. It was first applied for DiffServ provisioning in packet scheduling and packet dropping, in which packet queueing delay and loss rate are key QoS factors, respectively. Many algorithms have been designed to achieve proportional delay differentiation (PDD) in the network routers. They can be classified into three categories: rate-based; see BPR [9] for example, time-dependent priority based; see WTP [10] and adaptive WTP [15] for examples, and Little's Law-based; see PAD [10] and LAD [17] for examples. The work in [14] demonstrated that some of the algorithms can be tailored for request scheduling for PDD provisioning on the server side. However, the algorithms are not applicable to proportional response time differentiation because response time is not only dependent on a job's queueing delay but also on its service time, which varies significantly depending on the requested services.

Priority-based request scheduling strategies have been investigated for response time differentiation on Internet servers [2, 5, 8, 11]. In [8], the authors addressed strict priority scheduling strategies for controlling CPU utilization on Web servers. Incoming requests were categorized into the appropriate queues with different priority levels for the corresponding services. Requests of lower priority classes were only executed if no requests existed in any higher priority classes. The results showed that response time differentiation can be achieved but the quality spacings among different classes cannot be guaranteed by strict priority scheduling. Therefore, this kind of priority-based scheduling strategies cannot achieve proportional response time differentiation on Web servers.

In [18], we proposed a queueing-theoretical processing rate allocation scheme for proportional response time differentiation and then designed a process allocation mechanism to achieve various processing rates. While the objective was achieved in the long run, overall, the proportionality was weak and the variance was large. In this paper, we design and integrate a PID feedback controller with the queueing-theoritical rate allocation. The integrated approach improves over the previous efforts in the sense that it can quantitatively control quality spacings between different classes and provide more robust proportional response time differentiation.

In [1], the authors utilized feedback control approaches to achieve overload protection and performance guarantees and differentiation on Web servers. The strategy was based on real-time scheduling theory which states that response time can be guaranteed if server utilization is maintained below a pre-computed bound. Thus, control-theoretical approaches, in combination with content adaptation strategies, were formulated to keep server utilization at or below the bound. Our approach is complementary to their work in the sense that our approach integrates the queueing theory and control theory for proportional response time differentiation.

References

[1] T. F. Abdelzaher, K. G. Shin, and N. Bhatti. Performance guarantees for Web server end-systems: a control-theoretical approach. *IEEE Trans. on Parallel and Distributed Systems*, 13(1):80–96, 2002.

[2] J. Almeida, M. Dabu, A. Manikutty, and P. Cao. Providing differentiated levels of services in Web content hosting. In *Proc. ACM SIGMETRICS Workshop on Internet Server Performance*, pages 91–102, 1998.

[3] M. Arlitt, D. Krishnamurthy, and J. Rolia. Characterizing the scalability of a large Web-based shopping system. *ACM Trans. on Internet Technology*, 1(1):44–69, 2001.

[4] G. Banga, P. Druschel, and J. Mogul. Resource containers: A new facility for resource management in server systems. In *Proc. USENIX Symposium on Operating System Design and Implementation*, 1999.

[5] N. Bhatti and R. Friedrich. Web server support for tiered services. *IEEE Network*, 13(5):64–71, 1999.

[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. *IETF RFC 2475*, 1998.

[7] S. Chandra, C. S. Ellis, and A. Vahdat. Differentiated multimedia Web services using quality aware transcoding. In *Proc. IEEE INFOCOM*, pages 961–968, 2000.

[8] X. Chen and P. Mohapatra. Performance evaluation of service differentiating Internet servers. *IEEE Trans. on Computers*, 51(11):1,368–1,375, 2002.

[9] C. Dovrolis, D. Stiliadis, and P. Ramanathan. Proportional differentiated services: Delay differentiation and packet scheduling. In *Proc. ACM SIGCOMM*, 1999.

[10] C. Dovrolis, D. Stiliadis, and P. Ramanathan. Proportional differentiated services: Delay differentiation and packet scheduling. *IEEE/ACM Trans. on Networking*, 10(1):12–26, 2002.

[11] L. Eggert and J. Heidemann. Application-level differentiated services for Web servers. *World Wide Web Journal*, 3(2):133–142, 1999.

[12] Y. Huang and R. Gu. A simple fifo-based scheme for differentiated loss guarantees. In *Proc. IWQoS*, 2004.

[13] L. Kleinrock. *Queueing Systems, Volume II*. John Wiley and Sons, 1976.

[14] S. C. M. Lee, J. C. S. Lui, and D. K. Y. Yau. Admission control and dynamic adaptation for a proportional-delay DiffServ-enabled Web server. In *Proc. ACM SIGMETRICS*, 2002.

[15] M. K. H. Leung, J. C. S. Lui, and D. K. Y. Yau. Adaptive proportional delay differentiated services: Characterization and performance evaluation. *IEEE/ACM Trans. on Networking*, 9(6):908–817, 2001.

[16] K. Shen, H. Tang, T. Yang, and L. Chu. Integrated resource management for cluster-based Internet services. In *Proc. of USENIX OSDI*, pages 225–238, December 2002.

[17] J. Wei, C.-Z. Xu, and X. Zhou. A robust packet scheduling algorithm for proportional delay differentiation services. In *Proc. of IEEE Globecom*, 2004.

[18] X. Zhou, Y. Cai, G. K. Godavari, and C. E. Chow. An adaptive process allocation strategy for proportional responsiveness differentiation on Web servers. In *Proceedings of IEEE 2nd Int'l Conf. on Web Services (ICWS)*, July 2004.

[19] X. Zhou, J. Wei, and C.-Z. Xu. Modeling and analysis of 2D service differentiation on e-Commerce servers. In *Proc. of IEEE 24th Int'l Conf. on Distributed Computing Systems (ICDCS)*, pages 740–747, March 2004.

[20] X. Zhou, J. Wei, and C.-Z. Xu. Processing rate allocation for proportional slowdown differentiation on Internet servers. In *Proc. IEEE 18th Int'l Parallel and Distributed Processing Symposium (IPDPS)*, April 2004.

[21] H. Zhu, H. Tang, and T. Yang. Demand-driven service differentiation for cluster-based network servers. In *Proc. IEEE INFOCOM*, pages 679–688, 2001.