Quantitative Service Differentiation: A Square-Root Proportional Model

Xiaobo Zhou[1] and Cheng-Zhong Xu[2]

[1] Department of Computer Science, University of Colorado at Colorado Springs, 1420 Austin Bluffs Parkway, Colorado Springs, CO 80918, USA
{zbo}@cs.uccs.edu
[2] Department of Electrical & Computer Engineering, Wayne State University, 5050 Anthony Wayne Drive, Detroit, MI 48202, USA
{czxu}@wayne.edu

Abstract. Due to the open and dynamics nature of ubiquitous computing environments and services, quantitative service differentiation is needed to provide controllable quality of service (QoS) levels to meet changing system configuration and resource availability and to satisfy different requirements of applications and users. A proportional differentiation model was proposed in the DiffServ context, which states that QoS factors of certain classes of aggregated traffic be proportional to their differentiation weights. While it provides compelling proportionality fairness to clients, it lacks of the support of a server-side QoS optimization with respect to the resource allocation. In this paper, we propose and promote a square-root proportional differentiation model for delay-sensitive Internet services. Interestingly, both popular QoS factors with respect to delay, queueing delay and slowdown, are reciprocally proportional to the allocated resource usages. We formulate the problem of quantitative service differentiation as a resource allocation optimization towards the minimization of *system delay*, defined as the sum of weighted responsiveness of client request classes. We prove that the optimization-based resource allocation scheme essentially provides square-root proportional service differentiation to clients. We then propose a generalized rate-based resource allocation approach. Simulation results demonstrate that the approach provides quantitative service differentiation at a minimum cost of system delay.

1 Introduction

Popular Internet services must be scalable to support a large number of concurrent client requests reliably, responsively, and economically [3]. These scalability and availability requirements pose great challenge on both processing power and networking capacity. Meanwhile, clients of Internet services are diverse with respect to service expectations and access devices. To meet changing system configuration and resource availability and to satisfy different application and client requirements, there is an increasing demand for provisioning of different

levels of quality of service (QoS) [2, 4, 7, 10, 12]. Service differentiation is to provide a certain level of QoS guarantee to a class of aggregate requests based on predefined service level agreements with the clients. It can provide degraded levels of QoS to client requests when a server is heavily loaded, but also adapt the service quality to meet the variety of client preferences and devices. By adjusting the level of QoS, service differentiation techniques are able to postpone the occurrence of request rejection as the server load increases. They achieve the scalability in terms of cost-effectiveness. In addition, service differentiation can also provide an incentive for different charging policies for most of today's QoS-sensitive Internet services.

The service differentiation architecture is also demanded in ubiquitous computing and communications. By blending computers into the ubiquitous networking infrastructure, ubiquitous computing provides people with a most wide range of communication and information access services. The services are ensured to be accessible anytime, anywhere on any device. Provisioning of such services is a challenge because of the diversity of access devices and access networks. Their capabilities to receive, process, store and display media-rich content vary greatly. A user who accesses streaming services on a cellular phone must expect different service qualities from users on a highend workstation with high bandwidth networking capacities. A service differentiation architecture supports such heterogeneous QoS requirements and preferences by adapting the stream quality to various devices and access patterns.

To provide quantitative QoS differentiation, the proportional differentiation model [4] was proposed which states that QoS metrics of certain classes of aggregated requests should be proportional to their differentiation parameters, independent of their workloads. It is accepted as an important relative differentiation model and is applied in the proportional delay and loss rate differentiation in packet forwarding and dropping [4]. It is also adopted for server-side service differentiation [7, 11]. While the proportional model provides compelling proportionality fairness to clients, it lacks of the support of a server-side QoS optimization with respect to the resource allocation. This is due to the fact that from the server's perspective, the QoS factor offered to a client is usually not proportional to the resource usage allocated.

In this paper, we propose and promote a square-root proportional differentiation model for delay-sensitive Internet services, in which delay is the key QoS metric. We find that both popular delay factors, queueing delay and slowdown, are reciprocally proportional to the allocated resource usages according to the foundations of queueing theory. Note that slowdown measures the ratio of a request's queueing delay to its service time. It is known that clients are more likely to anticipate short delays for "small" requests and more willing to tolerate long delays for "large" requests [5]. The slowdown metric characterizes the relative queueing delay. We formulate the problem of quantitative service differentiation as a resource allocation optimization towards the minimization of system delay, defined as the sum of weighted delay of client request classes. We prove that the optimization-based resource allocation scheme essentially provides square-root

proportional differentiation to clients. We then propose a generalized rate-based resource allocation approach. Simulation results demonstrate that the approach provides quantitative service differentiation at a minimum system delay.

The structure of the paper is as follows. Section 2 reviews related work. Section 3 gives the square-root proportional differentiation model with a generalized resource allocation approach. Section 4 focuses on the performance evaluation. Section 5 concludes the paper with remarks on the implementation issues.

2 Related Work

The concept of service differentiation is not new. It was first invented for QoS-aware packet scheduling in the network core. The proportional differentiation model has been extensively studied in packet scheduling with respect to packet delay, packet loss, and connection bandwidth; see [4] for a representative approach. The work in [7] demonstrated that some approaches developed for proportional delay differentiation on networks can be tailored for its provisioning on Internet servers. While the model is compelling and important for Internet services due to its inherent proportional fairness and predictability properties, it lacks of the resource allocation optimization from the server's perspective.

On the basis of request classification, the objective of service differentiation on servers can be realized in five aspects: admission control, feedback control, content adaptation, resource allocation, and request scheduling. Admission control can provide QoS guarantees on delivered services, such as response time and bandwidth, to different classes. The work in [8] developed an admission control mechanism to provide differentiated bandwidth allocation to multiple service classes in a Web server. Feedback control operates by responding to measured deviations from the desired performance. The work in [9] proposed to integrate a queueing model with proportional integral control for relative request delay guarantees in Web servers. Multimedia objects are becoming a prevalent part of Web content. Content adaptation techniques are often used to deliver the media-rich objects in different formats, resolutions, sizes, color depths, and other quality control options for service differentiation purposes [12].

Priority-based request scheduling and rate-based resource allocation are general resource management approaches to ensuring the QoS of requests or preserving the quality spacings between two classes. There are efforts on priority-based request scheduling with admission control for response time differentiation [2]. Incoming requests were categorized into the appropriate queues and executed according to their priority levels [2]. The approach was shown to be effective for service differentiation in terms of queueing delays between lower and higher priority classes of traffic, but with no guarantee of quality spacings. In this paper, we propose and promote a square-root proportional differentiation model, which essentially provides the resource allocation optimization and proportionality fairness at the same time. We adopt a rate-based resource allocation approach in the simulations and performance evaluation.

3 A Square-root Proportional Model

Predictability and controllability are two basic requirements of quantitative service differentiation. Predictability requires that higher priority classes receive better or no worse service quality than lower priority classes. Controllability requires that a scheduler contain a number of controllable parameters that are adjustable for the control of quality spacings between classes. An additional requirement is fairness. Fairness is a quantitative extension of the predictability, which describes how much better QoS received by a class compared with that received by another class.

Proportional fair-sharing is compelling and important for various Internet services due to its inherent proportional fairness and predictability properties. Assuming incoming requests are classified into N contending classes, the proportional differentiation model aims to ensure the quality spacing between class i and class j to be proportional to certain pre-specified differentiation parameters δ_i and δ_j [4]. We consider delay as the key QoS factor in delay-sensitive Internet services. The quality factor of a request class is then represented by the inverse of its delay. That is,

$$\frac{q_i}{q_j} = \frac{\delta_j}{\delta_i} \qquad 1 \leq i, j \leq N,$$

where q_i and q_j are the delay factors of class i and class j, respectively. The model essentially considers the proportional fairness to the clients. From the viewpoint of the server, however, it lacks the support of a system-wide QoS optimization with respect to the resource allocation.

In the following, we formulate a general resource allocation problem that aims to optimize a system-wide QoS to the server. We will prove that the derived allocation scheme also provides proportional service differentiation, in square root, to the clients. The basic idea of QoS provisioning is to divide the resource allocation procedure into a sequence of short intervals. In each interval, based on the measured resource utilization and the predicted workload, the available resource usages of the server are differently allocated to N task servers handling traffic classes. We assume the processing rate of a server can be proportionally allocated to the task servers. Each task server presents a processing unit that handles requests from the same class in a FCFS manner. The objective of the optimization is to minimize the system delay. Based on the delay factor (q_i) for individual request classes, the system delay is naturally defined as $\mathcal{G} = \sum_{i=1}^{N} \delta_i q_i$, the weighted delay of all request classes. We introduce the system delay as the system-wide QoS of a delay-sensitive Internet server. We formulate the resource allocation for quantitative service differentiation as the following optimization problem:

$$\text{Minimize} \quad \mathcal{G} = \sum_{i=1}^{N} \delta_i q_i \tag{1}$$

$$\text{subject to} \quad q_i = f(c_i, l_i), \tag{2}$$

$$\sum_{i=1}^{N} c_i < C. \tag{3}$$

(1) defines the objective function of the quantitative service differentiation. It is to minimize the system delay of the server. It implies that requests from higher classes get lower delay (higher QoS). The rationale is its feasibility, differentiation predictability and controllability. (2) gives the definition of q_i with the allocated resource usage (c_i) and the workload (l_i) of class i. We assume that the server has a single processing resource bottleneck. Although processing a request often needs to consume resources of different types, resource management usually focuses on the allocation of the most critical resource [2, 5]. (3) gives the constraint of the resource allocation, where C is the resource available during the current resource allocation interval, which is the server's processing capacity minus the overhead of resource usage collection and allocation in each interval. A processing rate allocation scheme is needed to determine the amount of the resource usages allocated to the task servers for handling requests so that the resource utilization is maximized.

For quantitatively predictable and controllable differentiation, we need to have a closed form expression of quality factor(s) with respect to resource allocation. We find that both popular delay factors, queueing delay and slowdown, are reciprocally proportional to the allocated resource usage (processing rate). We consider a general workload model on each task server. Let m_1, m_{-1}, and m_2 be the first moment (mean), the moment of its inverse, and the second moment of the service time distribution, respectively. According to Pollaczek-Khinchin formula [6], we have

Lemma 1. *Given an $M/G/1$ FCFS queue on a task server i, where the arrival process has workload l_i. Then, the delay factor q_i is reciprocally proportional to the allocated resource usage to server i (c_i). That is,*

$$q_i = f(c_i, l_i) = \frac{\propto l_i}{2(c_i - l_i)}, \tag{4}$$

where $\propto$ is a value determined by the service time distribution of the traffic workload.

Proof. . Let d_i and s_i be the queueing delay and slowdown of a job on the task server i, respectively. Let λ_i denote the arrival process of request class i. We have $l_i = m_1\lambda_i$. According to Pollaczek-Khinchin formula [6], we have

$$d_i = \frac{\lambda_i m_2}{2(c_i - \lambda_i m_1)} = \frac{m_2/m_1 l_i}{2(c_i - l_i)} = \frac{\propto l_i}{2(c_i - l_i)}, \qquad \propto = m_2/m_1. \tag{5}$$

$$s_i = d_i m_{-1} = \frac{m_2 m_{-1}/m_1 l_i}{2(c_i - l_i)} = \frac{\propto l_i}{2(c_i - l_i)}, \qquad \propto = m_2 m_{-1}/m_1. \tag{6}$$

It concludes the proof.

Theorem 1. *The optimal resource allocation scheme to (1) essentially provides square-root proportional QoS differentiation. That is,*

$$\frac{q_i}{q_j} = g(l_i, l_j)\sqrt{\frac{\delta_j}{\delta_i}}, \tag{7}$$

where function $g(l_i, l_j)$ describes the workload relationship of two classes.

Proof. The optimization of (1) $\sim$ (3) is essentially a continuous convex separable resource allocation problem. According to foundations of resource allocation theory, its optimal solution occurs when the first order derivatives of the objective function over variables c_i are equivalent. Specifically, the optimal solution to (1) occurs when

$$-\frac{\propto \delta_i l_i}{(2(c_i - l_i))^2} = -\frac{\propto \delta_j l_j}{(2(c_j - l_j))^2} \qquad 1 \le i, j \le N. \tag{8}$$

It follows that

$$\frac{c_i - l_i}{c_1 - l_1} = \sqrt{\frac{\delta_i l_i}{\delta_1 l_1}}, \quad 1 \le i \le N. \tag{9}$$

Together with the constraint (3), the set of equations (9) leads to a linear equation system. It follows the generalized rate-based resource allocation approach as

$$c_i = l_i + \frac{(C - \sum_{j=1}^{N} l_j)\sqrt{\delta_i l_i}}{\sum_{j=1}^{N} \sqrt{\delta_j l_j}}. \tag{10}$$

The first term of (9) ensures that the task server will not be overloaded. The second term means that the remaining capacity of the server is proportionally allocated to different classes according to their scaled arrival rates and differentiation parameters.

Accordingly, the delay factor of a request class is calculated as

$$q_i = \frac{\propto l_i \sum_{j=1}^{N} \sqrt{\delta_j l_j}}{2(C - \sum_{j=1}^{N} l_j)\sqrt{\delta_i l_i}}. \tag{11}$$

It follows that

$$\frac{q_i}{q_j} = \sqrt{\frac{l_i}{l_j}}\sqrt{\frac{\delta_j}{\delta_i}}. \tag{12}$$

(12) shows that the allocation approach has the property of square-root proportional fairness as well, where $g(l_i, l_j) = \sqrt{l_i/l_j}$. It concludes the proof.

Thus, this square-root proportional allocation scheme meets a two-fold objective: one is the (square-root) proportional service differentiation provisioning; the other is the optimization of the resource allocation with respect to the system-wide QoS metric.

Remark. Recall that $l_i = m_1\lambda_i$. According to (12), the predictability of service differentiation holds if and only if $\sqrt{\lambda_i/\lambda_j} \leq \sqrt{\delta_i/\delta_j}$ for $1 \leq j < i \leq N$. Otherwise, the predictability, that is, $q_i \leq q_j$ if and only if $i > j$, will be violated. One solution is temporary weight promotion in [13]. When it is applied in this context, based on the current session arrival rates and the number of visits to a state in sessions, the scheduler temporarily increases weight δ_i in the current resource allocation interval so that the predictability of the relative service differentiation holds. In this context, the allocation scheme becomes heuristic.

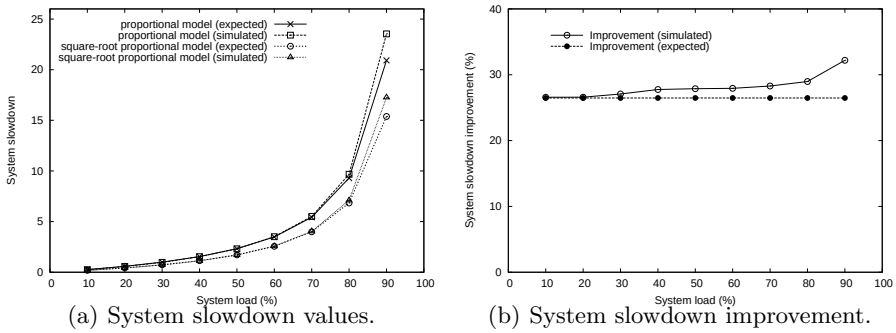
4 Performance Evaluation

To evaluate the rate-based resource allocation approach on quantitative service differentiation, we built a simulator. The processing rate allocation can be implemented by the use of many different mechanisms, as we are going to discuss in Section 5. In this work, we built a simulator because that without being affected by the methods of implementations, simulation can effectively evaluate the performance of the rate allocation schemes by itself. The simulator consists of a number of request generators, waiting queues, a load estimator, a processing rate allocator, and a number of task servers. We conducted the simulations with requests generated by using GNU scientific library. We consider *Bounded Pareto* distribution, a typical heavy-tailed distribution, for the service time distribution of workloads. Given a distribution, the first moment m_1, the moment of its inverse m_{-1}, and the second moment m_2 can be represented in closed forms [11]. Due to the space limitation, we show the results of slowdown differentiation only, while the results of queueing delay differentiation are similar. Each result reported is an average of 100 runs, unless otherwise specified.

4.1 Impact of Square-root Proportional Model on System QoS

Fig. 1 shows the results of system slowdown with the increase of server load by the use of the square-root rate allocation approach proposed in this paper and the proportional rate allocation scheme proposed in [11]. The predefined differentiation weight ratio of high priority class to low priority class ($\delta_1 : \delta_2$) is 4:1. The workload ratio of the two classes ($l_1 : l_2$) is 1:4.

Fig. 1(a) illustrates the effectiveness of the generalized rate allocation approaches by comparing the achieved system slowdown with the calculated slowdown. We found that simulation results closely agree with the expected results under various load conditions, before the server is heavily loaded ($\leq 70\%$). The gap between the simulated results and the expected results increases as the server load increases. This is due to the variance of both inter-arrival time distributions and the heavy-tailed service size distributions. When the workload is above 80%, the server observes more backlogged jobs. Slowdown of some backlogged jobs could be very large, which increases the variance of the simulation results. Second, we find that the square-root proportional rate allocation approach can reduce system slowodown significantly compared to that received by



(a) System slowdown values. (b) System slowdown improvement.

Fig. 1. System slowodown and its improvement due to different differentiation schemes.

the proportional approach. The improvement is shown in Fig. 1(b) with both predicted and simulated values. We also want to note that when two classes have the equal workload, the proportional approach indeed achieved the same system slowdown as a non-differentiation approach did (the results are not shown due to the space limitation). The results demonstrate the superiority of the square-root proportional model in optimizing resource allocation for minimizing the system slowdown. Also, we note that the improvement degree of system slowdown is dependent on the workload ratio and differentiation weight ratio of the classes.

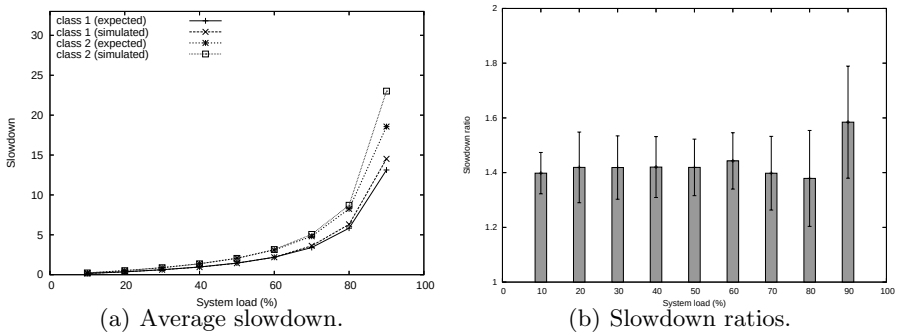
4.2 Differentiation Properties of Square-root Proportional Model

We then show the differentiation predictability and controllability of the rate allocation approaches. The predefined differentiation weight ratio of two classes ($\delta_1 : \delta_2$) is set to 2:1. The workload ratio of two classes ($l_1 : l_2$) is set to 1:1.

Fig. 2 shows the quantitative slowdown differentiation when the server workload changes from 10% to 90%. The results were due to the square-root proportional rate allocation approach. Fig. 2(a) shows a per-class view of request slowdown. It shows that the expected results closely agree with the simulated results when the workload is moderate. This is due to the fact that the rate allocation approach is guided by the predictive queueing model. When the system load is close to system capacity, say at 90%, the rate allocation approach generates poor predictability. This can be explained by the fact that as the system load is close to its capacity, the impact of the variance of inter-arrival times on slowdown dominates. This mitigates the controllability of the queueing-theoretical rate allocation approach. Fig. 2(b) shows the achieved slowdown ratio of two classes with the 95% confidence interval measured with 20 runs. As the workload is close to the server capacity, the predictability is not desirable.

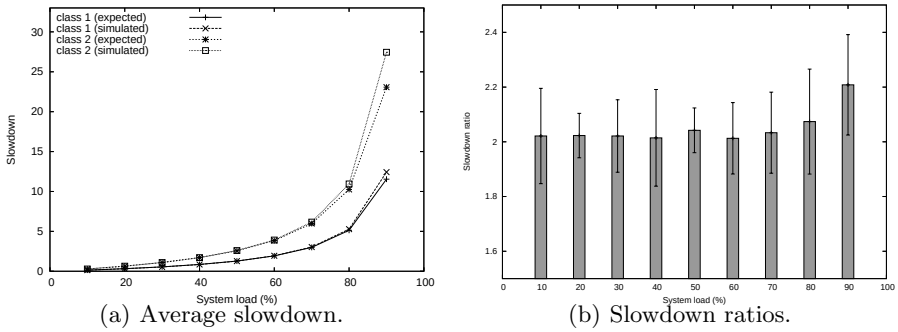
We then show the differentiation results by the use of the proportional rate allocation approach. Fig. 3 depicts the achieved per-class request slowdown with the 95% confidence interval measured with 20 runs at different system workload conditions. It has the basic similar shape as Fig. 2. Results show that the rate allocation approach is able to achieve the proportional differentiation as well.

From these results, we find that the generalized rate allocation approaches can achieve the objective of providing quantitative service differentiation to classes with different differentiation parameters under various system load conditions in long timescales.



(a) Average slowdown. (b) Slowdown ratios.

Fig. 2. Two-class differentiation due to the square-root proportional model.



(a) Average slowdown. (b) Slowdown ratios.

Fig. 3. Two-class differentiation due to the proportional model.

We conducted a wide ranger of sensitivity analysis. We varied the number of request classes, the arrival rate ratios of the classes, and the differentiation weight ratios of the classes. While we do not have space to present all of the results, note that we did not reach any significantly different conclusion regarding to the quantitative service differentiation achieved by two models.

5 Conclusion and Future Work

In this paper, we proposed a square-root proportional service differentiation model for ubiquitous computing environments and services. It was derived from the resource allocation optimization towards minimizing the system slowdown of an Internet server. We proved that the optimization-based allocation scheme essentially provides square-root proportional fairness to clients. Simulation results demonstrated that the derived rate allocation approach can provide quantitative service differentiation at a minimum of system delay.

A challenging issue left is how to allocate resources to meet an allocation of processing rate. In the theoretical analysis, we adopted the concept of task server to represent the processing rate that can be allocated to a traffic class. In practice, it can be a process on an individual Web server, a thread on a multi-thread server, a processor in a parallel machine, and a node in a server cluster. Process abstraction serves both as a protection domain and as a resource principal in current general-purpose operating systems. However, because an application does not have the control over the consumption of resources that the kernel consumes on behalf of the application, resource principals do not always coincide with processes. There are efforts on OS support for service differentiation, as exemplified by resource containers [1]. The implementation of the rate allocation schemes on servers deserves further study and evaluation.

Acknowledgement This research was supported in part by an award from the Committee on Research and Creative Works (CRCW) of the University of Colorado at Colorado Springs.

References

1. G. Banga, P. Druschel, and J. Mogul. Resource containers: A new facility for resource management in server systems. In *Proc. USENIX OSDI*, 1999.
2. X. Chen and P. Mohapatra. Performance evaluation of service differentiating Internet servers. *IEEE Trans. on Computers*, 51(11):1,368–1,375, 2002.
3. C.-Z. Xu. Scalable and Secure Internet Services and Architecture. Chapman & Hall/CRC Press, ISBN 1-58488-377-4, 2005.
4. C. Dovrolis, D. Stiliadis, and P. Ramanathan. Proportional differentiated services: Delay differentiation and packet scheduling. In *Proc. ACM SIGCOMM*, pages 109–120, 1999.
5. M. Harchol-Balter. Task assignment with unknown duration. *Journal of ACM*, 29(2):260–288, 2002.
6. L. Kleinrock. *Queueing Systems, Volume II.* John Wiley and Sons, 1976.
7. S. C. M. Lee, J. C. S. Lui, and D. K. Y. Yau. A proportional-delay diffserv-enabled Web server: admission control and dynamic adaptation. *IEEE Trans. on Parallel and Distributed Systems*, 15(5):385–400, 2004.
8. K. Li and S. Jamin. A measurement-based admission-controlled Web server. In *Proc. IEEE INFOCOM*, pages 544–551, 2000.
9. C. Lu, X. Wang, and X. Koutsoukos. Feedback utilization control in distributed real-time systems with end-to-end tasks. *IEEE Trans. on Parallel and Distributed Systems*, 16(6):550–561, 2004.
10. M. M. Rashid, A. S. Alfa, E. Hossain, and M. Maheswaran. An analytical approach to providing controllable differentiated quality of service in web servers. *IEEE Trans. on Parallel and Distributed Systems*, 16(11):1022–1033, 2005.
11. X. Zhou, J. Wei, and C.-Z. Xu. Processing rate allocation for proportional slowdown differentiation on Internet servers. In *Proc. IEEE IPDPS*, pages 88–97, 2004.
12. X. Zhou and C.-Z. Xu. Harmonic proportional bandwidth allocation and scheduling for service differentiation on streaming servers. *IEEE Trans. on Parallel and Distributed Systems*, 15(9):835–848, 2004.
13. H. Zhu, H. Tang, and T. Yang. Demand-driven service differentiation for cluster-based network servers. In *Proc. IEEE INFOCOM*, pages 679–688, 2001.