SLA Based Profit Optimization in Web Systems

Li Zhang
IBM
T.J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598
zhangli@us.ibm.com

Danilo Ardagna
Politecnico di Milano
Dipartimento di Elettronica e Informazione
Via Ponzio 34/5, 20133 Milano, Italy
ardagna@elet.polimi.it

ABSTRACT

With the rapid growth of eBusiness, the Web services are becoming a commodity. To reduce the management cost for the IT infrastructure, companies often outsource their IT services to third party service providers. Large service centers have been set up to provide services to many customers by sharing the IT resources. This leads to the efficient use of resources and a reduction of the operating cost. The service provider and their customers often negotiate utility based Service Level Agreements (SLAs) to determine the cost and penalty based on the achieved performance level. The system is based on a centralized controller which can control the request volumes at various servers and the scheduling policy at each server. The controller can also decide to turn ON or OFF servers depending on the system load. This paper designs a resource allocation scheduler for such web environments so as to maximize the profits associated with multiple class SLAs.

Keywords

Resource Allocation, Quality of Service, Utility Function, SLA Optimization, Load Balancing

1. INTRODUCTION

The service provider and their customers often negotiate utility based Service Level Agreements (SLAs) to determine the cost and penalty based on the achieved performance level. The service provider need to manage its resource to maximize its profits.

This paper designs a resource allocation scheduler for Web service environments. The scheduling policy is designed to maximize the revenue while balancing the cost (or energy) of using the resources. The overall profit (utility) includes the revenues and penalties incurred when Quality of Service guarantees are satisfied or violated. The revenue depends on the QoS levels in a discrete fashion. We show that the overall problem is NP-hard. We further develop meta-heuristic solutions based on the tabu-search algorithm. The neighborhood exploration is based on a fixed-point iteration, which requires solving a new network allocation flow problem. Experimental results are presented to show the benefits of our approach.

2. THE SYSTEM

We consider the service system to be a distributed computer system consisting of M heterogeneous clusters of servers hosting N different e-commerce web sites. Each cluster is built from a number of homogeneous machines. There are totally K classes of request streams. Each class of request can be served by a collection of servers. For simplicity assume that each class of request is associated with a single web site. Let $A_{i,k}$ be the indicator function that assigns requests (and sites) to clusters: $A_{i,k}$ equals 1 if class k request can be executed by server i, 0 otherwise.

The architecture comprises of a request dispatcher in front of the clusters to assign the incoming requests to individual servers in the cluster. The controller can also establish the scheduling policy at each server. Each server has a Generalized Processor Sharing (GPS) scheduler. The allocation weights for each class can be set by the controller. The controller can also turn OFF and ON individual server inside a clusters in order to reduce the overall cost. For each class k requests, a step-wise utility function is defined to specify the per request revenue (or penalty) incurred when the corresponding average response time assumes a given value. Figure 1 shows, as an example, the plot of an utility function. We observe the discontinuity in the function in Figure 1. As we will discuss in the next section, this discontinuity in the cost function and the discrete nature of the problem make the optimization problem NP-hard. In the literature the load balancing problem with SLA profits was faced considering always continuous convex and differentiable cost functions [4]. Considering step-wise functions of the mean response time is more intuitive from the customer point of view and are currently adopted [2].

Each data center is modeled by a queueing network composed of a set of multi-class single-server queues and a multi-class infinite-server queues. The former represents the collection of servers within heterogeneous clusters. The infinite-server queues represent the client-based delays, or think times, between the server completion of one request and the arrival of the subsequent request within a Web session. User sessions begin with a class k request arriving to the data center from an exogenous source with rate λ_k.

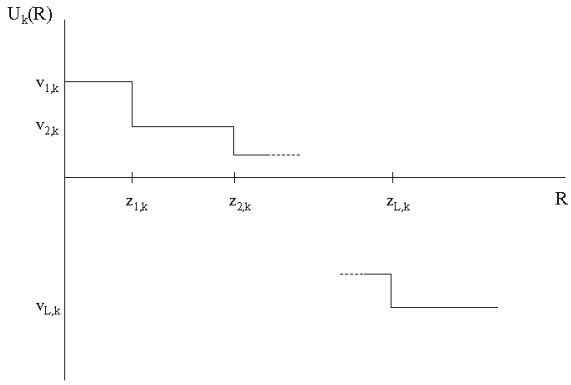

Figure 1: Utility Function

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

Table 1: Problem Variables

Variable	Description
M_i	Number of homogeneous server within cluster i
y_i	Number of cluster i servers ON
C_i	Capacity of a single server in cluster i
μ_k	Service rate for class k jobs at a server of capacity 1
$\lambda_{i,k}$	Load at cluster i for class k jobs
$\lambda_{i,m,k}$	Load at server m in cluster i for class k jobs ($\sum_{m=1}^{y_i} \lambda_{i,m,k} = \lambda_{i,k}$)
$\phi_{i,m,k}$	Scheduling GPS parameter for class k jobs at server m within cluster i
$R_{i,m,k}$	Response time for class k jobs at server m in cluster i
$U_k(R)$	Utility step-wise function for class k jobs
L	Number of thresholds for utility functions
c_i	Cost associated with turning on a server in cluster i

Upon completion the request either returns to the system as a class k' request with probability $p_{k,k'}$ or it completes with probability $1 - \sum_{l=1}^{K} p_{k,l}$. Let Λ_k denote the aggregate rate of arrivals for class k requests $\Lambda_k = \sum_{k'=1}^{K} \Lambda_{k'} p_{k',k} + \lambda_k$. In next sections the notation reported in Table 2 will be adopted. In the following, we will assume that the first $K-1$ job classes are associated with SLA, and class K is the best effort class.

The analysis of multi-class queueing system is notoriously difficult. We use the GPS bounding technique in [5] to approximate the queueing system. In the approximation each multi-class single-server queue associated with server m in cluster i is decomposed into multiple single-class single-server queues with capacity greater than or equal to $C_i \phi_{i,m,k}$. The response times evaluated in the isolated per-class queues are then upper bounds on the corresponding measures in the original system.

3. OPTIMIZATION PROBLEM

Given the system model in Section 2 we formulate the cost optimization problem below.

$$\max \quad \sum_{i=1}^{M} \left(\sum_{m=1}^{y_i} \sum_{k=1}^{K-1} U_k(R_{i,m,k}) \lambda_{i,m,k} - c_i y_i \right) \tag{1}$$

$$\sum_{i=1}^{M} \lambda_{i,k} \quad = \quad \Lambda_k \tag{2}$$

$$\lambda_{i,k} = 0 \qquad \text{if } A_{i,k} = 0 \tag{3}$$

$$\lambda_{i,k} \geq 0 \qquad \text{if } A_{i,k} = 1 \wedge y_i > 0 \tag{4}$$

$$\sum_{k=1}^{K-1} \phi_{i,m,k} \quad \leq \quad 1 \tag{5}$$

$$R_{i,m,k} = \frac{1}{C_i \mu_k \phi_{i,m,k} - \lambda_{i,m,k}}; \qquad y_i > 0; \quad \lambda_{i,m,k} < C_i \mu_k \phi_{i,m,k} \tag{6}$$

$$y_i \in [0, M_i]; \qquad y_i \text{ integral}$$

Note that in autonomic computing systems the exogenous arrival usually is a prediction of the arrival rate for the current inter-scheduler period. Equations (3-4) assign sites requests to clusters according to $A_{i,k}$ constrains and the status of the cluster (servers ON or OFF). Here y_i, $\lambda_{i,m,k}$ and $\phi_{i,m,k}$ are decision variables and overall we have a Mixed Integer Programming problem. It can be shown that, if the number of server ON and the load at each server are fixed, then in order to maximize the objective function one can maximize revenues at single servers obtaining $\sum_{i=1}^{M} \overline{y_i}$ multiple choice knapsack (MKP) sub-problems [1]. Vice versa, if the number of server ON and the scheduling policy at each server are fixed, then in order to maximize the objective function one can establish the load at each server and solve $K-1$ NP-hard network flow allocation sub-problems. The overall problem is NP-hard and is solved by implementing a tabu-search algorithm. The evaluation of the neighborhood is based on a fixed point iteration of MKP and network flow resource allocation problems, whose solution is obtained by applying the HEU heuristic [1] and an ad-hoc local search algorithm. In order to evaluate the effectiveness of our approach, several tests were performed. Data centers with 200 servers and 100 job classes have been considered. Service times were random generated and the utilization of data center resources varied between 0.2 and 0.8. An estimate of the quality of our solution is obtained by comparing our results with results of an exhaustive search algorithm. Results are encouraging since the average error was about 30%. In order to compare our results with other approaches in the literature which adopt utilization thresholds in resource allocation control [3] the number of servers that has to be turned ON is evaluated as the number of servers that keeps the utilization of the data center equals to 0.6 and the proportional assignment schema is applied for routing and scheduling problems [4]. Considering this scenario our approach improves SLA revenues of one order of magnitude since for the same load our controller is able to reduce the number of servers ON. Furthermore solutions show that in general the load is not balanced among all of the servers of a cluster, as in the proportional assignment schema.

4. CONCLUSIONS

We proposed an allocation controller for web data center environments which maximizes the profits associated with multi-class Service Levels Agreements. Experimental results show that revenues that can be obtained with a proportional assignment schema can be significantly improved. Future work will consider the problem of maximization of SLA profits in multi-tiers systems and the model will be extended in order to include in the cost function the tail distribution of response times.

5. REFERENCES

[1] Akbar, M. M., Manning, E.,G., Shoja, G., C., Khan, S. 2001. *Heuristic solution for the Multiple-Choice Multiple-Dimension Knapsack problem*. Conference on Computational Science, San Francisco, USA.

[2] Boutilier, C., Das, R., Kephart, G. Tesauro, G. Walsh W. 2003. *Cooperative Negotiation in Autonomic Systems using Incremental Utility Elicitation*. To appear, Uncertainty in Artificial Intelligence.

[3] Chase , J. S., Anderson, D. C. 2001 *Managing energy and server resources in hosting centers*. In Proc. of the eighteenth ACM symposium on Operating systems principles, 103-116.

[4] Liu, Z., Squillante, M. S., Wolf, J. 2002. *Optimal Resource Management in e-Business Environments with Strict Quality-of-Service Performance Guarantees*. IEEE Conference on Decision and Control.

[5] Zhang, Z. L., Towsley, D., Kurose, J. 1995. *Statistical analysis of the generalized processor sharing scheduling discipline*. IEEE Journal on Selected Areas in Communications, 13,6, 1071-1080.