Characterizing Web Application Performance for Maximizing Service Provider's Profits in Clouds

Overview

A number of challenges in implementing cloud technique related to further improving Web application performance and decreasing the cost. In order to achieve high profits, cloud-based web application providers must carefully balance cloud resources and dynamic workloads. However, this task is usually difficulty because of the complex nature of most web application. The volume of demand in applications fluctuates on several times scales. Therefore, performance model must be effectively adjustable to workloads in order to support next step scheduling. We presented a predictive performance model to analyze such applications and to determine when and how much resource to allocate to each tier of an application. In addition, we proposed a new profit model to describe revenues specified by the Service Level Agreement (SLA) and costs generated by leased resources. Furthermore, we employed profit driven model to guide our resource management algorithms to maximize the profits earned to the service provider. We also designed and implemented a simulation experiment on CloudSim that adopts our proposed methodology. Experimental results indicated that our model faithfully captures the performance and resources are allocated properly in response to the changing workload, thus the goal of maximizing the profit has been achieved.

Problem Statement

◆ Workload Prediction

The capacity manager is required in the framework as for adjusting allocation of VMs for future demand. More specifically, we want to predict the demand curves in the time period $[t_i, t_i + \Delta t]$ for SaaS provider, where t_i denotes the current time and Δt denotes the prediction period. We deploy auto-regressive (AR) mode to predict the expected demand curve. In our experiments, we used linear AR model provided by MATLAB to obtain $\lambda_{i,k+1}$. Its value is estimated by using the historical values $\lambda_{i-1,k}$, …, $\lambda_{i,k-j}$ as:

$$\lambda_{i,k+1} = \sum_{j=1}^{k} \eta_j \cdot \lambda_{i,k-j} + \epsilon_t \qquad (1)$$

where η_1, η_2, …, η_j constitutes a set of parameters for historical values, and ϵ_t is white noise uncorrelated. All of the above parameters can be computed from historical data.

◆ Performance Prediction

The resource pool is modeled by a queuing network composed of a set of multi-class single-VMs queue as shown in Fig. 1.

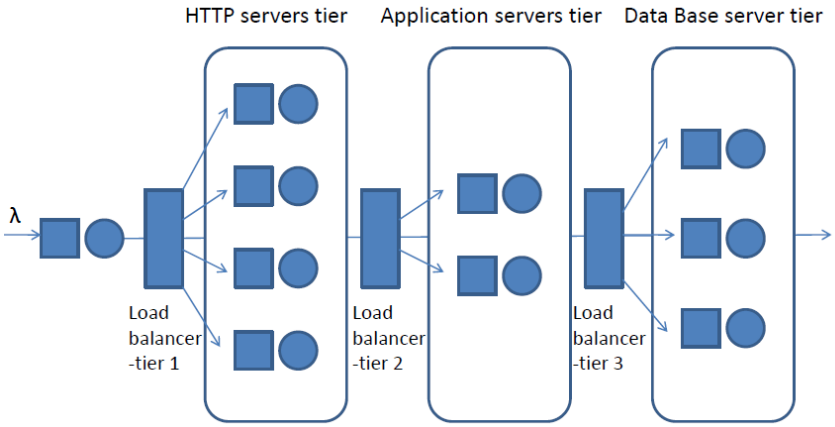


Figure 1. Performance model

The application environment under study is a resource pool consisting of m class of heterogeneous VM. We extend this residence time to Equation 4 by adding terms representing system characteristics as follow:

$$res_{S_{j,i}} = \frac{1}{U_j * C_i * \phi_{j,i}/\beta_i - \sum \lambda_{j,i}/n_i} \qquad (2)$$

A user request is supported by multi-tier web applications. We apply our control schemes and solve the profit optimization for independent tiers. When confronted with different kinds of service, each VM distribute its own computing resource in accordance with the ratio of service rate. Let $C_i = \left(c_i^{cpu}, c_i^{ram}, c_i^{bandwidth}\right)^T$ specifies the capacity of a single VM of single tier i to serve S_j. To be specific, the service rate for service S_j at single tier i, includes the CPU, RAM, and bandwidth. By this definition, the estimated CPU service rate of a service managed by tier i is given by:

$$c_i^{cpu} = \frac{capcity * cores(p)}{ls_i} \qquad (3)$$

Let $U_i = \left(u_i^{cpu}, u_i^{ram}, u_i^{network}\right)$ denotes the average utilization of specific resource at single tier i. β_i is the ratio of $\lambda_{j,i}$ of different services, and accordingly, we unite different service into one service. Let $\phi_{j,i}$ be the scheduling parameter for service S_j at tier i. To solve for $\phi_{j,i}$, one can choose a regression method from a variety of known methods in the literature. An approximated $res'_{S_{j,i}}$ for the next interval Δt can be calculated through Equ. (3)

Finding the ideal regression method for this problem is over the scope of this paper, and one can choose a regression method from a variety of known methods in the literature. In our system, the object of for regression method is to minimize the variance: $\sum_k (res'_{S_{j,i}} - res_{S_{j,i}})_k^2$ where *k* is the index of monitoring window over time. Then, this set of $\phi_{j,i}$ can be solved in order to estimate the parameter of different service in different tiers. After this step, the approximated response time of next monitoring window is computed from the predicted $\lambda_{i,k+1}$ and estimated $\phi_{j,i,k+1}$.

◆ Profit-driven scheduling policy

In Figure 4, Marginal Revenue (MR) and the Marginal Cost (MC) are two critical factors we need to account for in this problem: $MR = \frac{dRev}{dVM}$, $MC = \frac{dC}{dVM}$

The maximization of profit is when MR=MC at point A. The total profit is $P = R - C$.

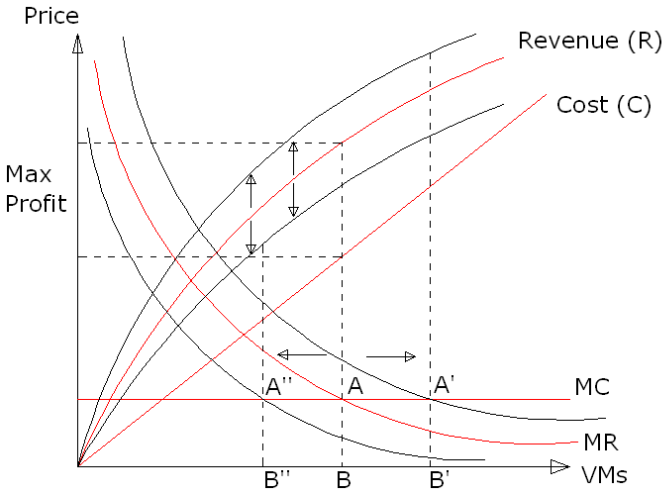


Figure 2. Profit-driven scheduling policy

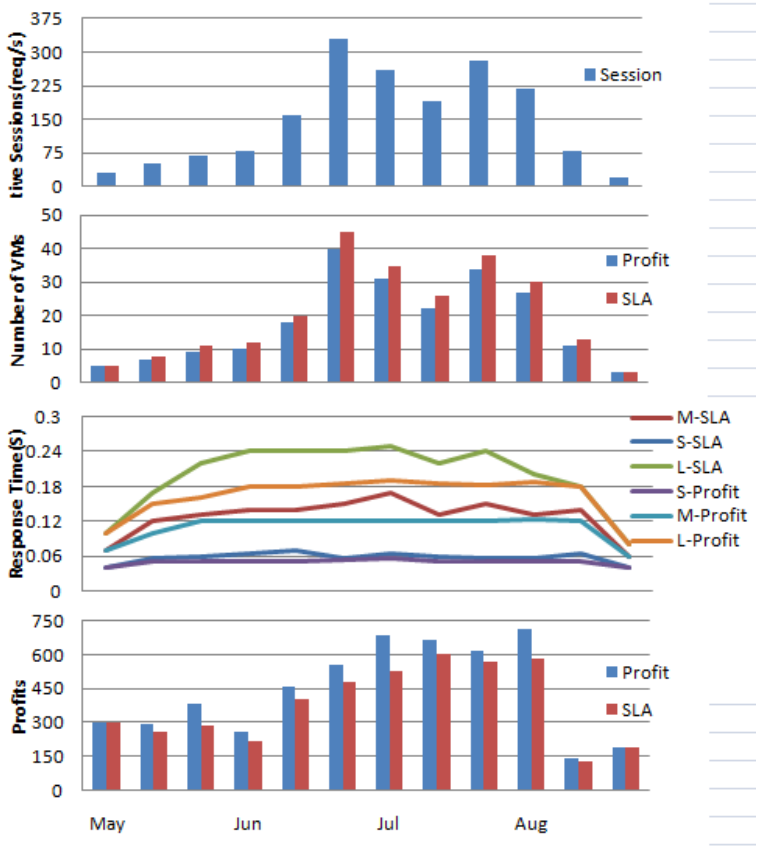
Simulation and Experiment

To evaluate the effectiveness of the proposed approach, we have designed and implemented a prototype of our framework using CloudSim [8], a toolkit for modeling and simulation of Cloud computing infrastructures and services. The simulation data consisted of 20 VMs on homogenous physical nodes on the first place. In the simulation, each node is modeled to have one CPU core with performance of 3000 MIPS, 8 Gb of RAM and I TB storage, and each VMs has 512 Mb of RAM, the same as the physical machine in our lab for the purpose of processing physical environments in the future. Table 1 shows all parameters of our simulation.

TABLE I. CLOUDSIM SIMULATION PARAMETER SETTINGDS

Simulation Parameters	Value
Initial number of VMs	5
VM cost	0.3$
VM per Memory	0.05$
VM per Storage	0.1$
VM per Bandwidth	0.1$
Small service revenue function	$R_{S_1} = 2/\sqrt{t}$
Middle service revenue function	$R_{S_2} = 3/\sqrt{t}$
Large service revenue function	$R_{S_1} = 5/\sqrt{t}$

We repeated these two experiments by using our profit-driven provisioning technique and QoS-driven. Our results are shown in Figure 3.



Firgue 3. Two scheduler evaluation: QoS-driven scheduling policy and Profit-driven scheduling policy. From the top to bottom: input workload (a), number of VMs needed in two policies (b), response time of different services (c), profit obtained in two policies (d).

The entire results obtained from our simulation are summarized in Table 2. Our results demonstrated that Service provider and Cloud resource provide can both be profitable through profit-driven provisioning technique by less number of VMs and 15.37% profit increased. Also, because that our provisioning technique is able to take profits imposed by revenue and cost as priority, it can also maintain a relatively acceptable response time targets by adding more VMs when profits do not reach the max at this workload.

TABLE II. OVERALL COMPARATIVE RESULTS

Algorithm	Avg. response time (%)	Total Revenue	Total cost	Total profit
Profit	84.1%	15650.0	10396.6	5253.4(15.37%)
SLA	100%	17200.7	12663.3	4537.4

Acknowledgement

This paper is also supported by the Project of Daystar of Shanghai Jiao Tong University.

Publication

Xi Chen, Haopeng Chen, Qing Zheng, Wenting Wang, Guodong Liu, Characterizing Web Application Performance for Maximizing Service Provider's Profits in Clouds, to be appeared on 2011 IEEE International Conference on Cloud and Service Computing, Hong Kong,China, 2011.12.12-2011.12.14