

Performance Models Oriented to the Dynamic Resource Provisioning in Shared Data Centres

Wang Xiu-wen^{*}, Qu Hai-ping^{*}, Xu Lu^{**}, Han Xiao-ming^{**}, Zhang Jian-gang^{**}

^{*} Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190

^{**} Graduate University of the Chinese Academy of Sciences, Beijing, 100039

{wangxw, quhp, xulu, hanxm, zhangjg}@ict.ac.cn

Abstract— It is a challenge to quickly supply performance numbers online driving dynamic resource provisioning in shared centres in face of the complication of applications both in scale and architecture. In this paper, we provide a practical solution to the above problem by laying out a theoretical framework. In order to improve the representative characterization of workload, we classify the workload into classes and adopt the regression-based methodology to extract these parameters online. We constructed both effective open and closed queuing models to evaluate the correctness and the generality of our idea. In our experiments, we analyse the effectiveness of the regression method with different number of classes. The results for the performance evaluation in the open queuing network show that almost 100% tests show relative error less than 2%, so these performance indexes can be effectively used as the basis for autonomic resource provisioning.

Keywords— analytical models, closed queuing network, open queuing network, regression methodology, workload classification

I. MOTIVATION

In face of the problem of energy consumption, cooling and space caused by vast idle servers in data centre, shared data centre emerges as a solution. Using the virtualization technology, it implements the resource sharing, increases the utilization of resource, and meets the application's need of dynamic requirement for resource by means of dynamic resource provisioning. Self-adaptive resource provisioning requires exactly predicting the performance metrics of the applications in response to swift workload changes. Nowadays, Computer systems become more and more complicated both in scale and in architecture, and moreover it is an uneasy task to predict the behavior of the users of the system. All these factors provide huge challenges for precisely predicting the performance of the system.

The need for swift changes and timely performance predictions argues against the use of traditional simulation models and black box models. Our point is that effective analytic models can quickly supply performance numbers, which then can provide powerful and simple solutions for dynamic resource provisioning. Analytical model describes

the relationship between the performance, computer system and workload through mathematical tools. The advantage of analytic models relates to their ability of providing a contained abstraction of the system by considering flows of customers (requests) in the queuing network (tiers). The effectiveness of the modelling ability of the queuing network relates to whether this abstraction is done properly. If salient characteristics of the system workload are captured well within the abstraction, then simple queuing network models can be effective in predicting the performance of complex systems.

Facing the problem of online performance evaluation using analytical models, we need to resolve the following two problems. The first problem needed to solve is the online extraction of parameters used in the analytical models, which are also called the characteristic parameters of the workload. Another problem to solve is the construction of the proper model.

According to the above requirements, we provide a practical solution to the above problems by laying out a theoretical framework. As for the method used for parameters extraction, we first classify the workload according to some attributes, and then use a regression-based methodology to approximate CPU service demands and DISK service demands of requests on given hardware. With respect to the method used for performance evaluation, we constructed both open and closed queuing network. During phase of experimental validation of model, we conducted a detailed set of experimentation using the TPC-W ecommerce suite [1]. Regardless of the system status (e.g., light load and heavy load), our experiments show that the values produced by the performance models which in turn get the parameter values based on classification method and regression-based analysis are in excellent agreement with experimental data. Furthermore, we also provide the analysis for the sensitivity of regression to the number of classes.

The rest of this paper is organized as follows. Section II introduces the regression-based methodology used in online extraction of characteristic parameters. Section III describes the analytical models constructed. Section IV presents the prototype implementation and the experimental results. Section V introduces some related work. Future work and conclusions are discussed in section VI.

II. ONLINE EXTRACTION OF CHARACTERISTIC PARAMETERS

Characteristic parameters of the workload are the basis and input of the performance models. In this paper, we will focus on the online extraction of static attributes, especially the average service demand of requests. As for the other parameters of workload, we can use the already existing methods.

Traditionally we use the utilization law ($U_i = X_i * S_i$) and service demand law ($D_i = U_i / X_0$) to calculate their values. In the formula, the subscript i designates i -th resource and the X_0 is the average throughput of the system. This method can calculate the average service demands of the requests. But in real system, not all requests that flow through the resources of a QN are similar in terms of the resources used and the time spent at each resource. In this case, real workloads can be viewed as a collection of heterogeneous components. Because of this heterogeneity, many times the representation of a workload by a single class lacks accuracy. The motivation for partitioning the workload is twofold: improve representativeness of the characterization and increase the predictive power of model.

From the point above, we can then classify the requests by some attributes. Let's suppose there are N classes, and we use the regression analysis introduced below to calculate the service demands for every class. Using the utilization law, we can get the Eq. (1) and Eq. (2).

$$\sum_i N_i * D_{i,cpu} = U_{cpu} * T, 1 \leq i \leq N \quad (1)$$

$$\sum_i N_i * D_{i,disk} = U_{disk} * T, 1 \leq i \leq N \quad (2)$$

The meaning of symbols used in the formulas is described in Table 1.

Table 1. Notation used in utilization law

Symbol	Meaning
T	the length of the monitoring window
N_i	the number of requests of the i -th class
U_{cpu}	the average CPU utilization
U_{disk}	the average DISK utilization
$D_{i,cpu}$	the average CPU service demand of the i -th class
$D_{i,disk}$	the average DISK service demand of the i -th class

Let $C_{i,cpu}$ denote the approximated CPU cost of $D_{i,cpu}$ for i -th class. In the same way we use the $C_{i,disk}$ to denote the approximated DISK cost of $D_{i,disk}$ for i -th class. Then, the approximated utilizations of CPU and DISK can be calculated by the Eq. (3) and Eq. (4).

$$U'_{cpu} = \frac{\sum_i N_i * C_{i,cpu}}{T} \quad (3)$$

$$U'_{disk} = \frac{\sum_i N_i * C_{i,disk}}{T} \quad (4)$$

To get the values of $C_{i,cpu}$ and $C_{i,disk}$, one can choose a regression method from a variety of known methods in the

literature. In this paper, we adopt the method of ordinary least squares (OLS).

The online regression-based calculation of service demands is feasible. First, the method of regression is mature and we can choose one from a variety of known methods in the literature. Moreover, the time complexity of algorithm is efficient for online calculation. In general, the common least squares algorithms have polynomial time complexity as $O(u^3v)$ when solving v equations with u variables, and hence, can be efficiently used as a part of on-line resource evaluation method[2].

III. ANALYTICAL MODEL

As we know [7], the analytical models using queuing network are classified as three classes, that's closed QN, open QN and mixed QN. Intuitively, open QN and closed QN are used to describe different environment of the application. Nevertheless we stress that closed system is equivalent to the open system with admission control. In this paper, we construct both open and closed QN and conduct experiments to validate them.

A. Closed Queuing Network

Closed QN is characterized by having a fixed number of requests. One closed QN with M tiers is shown in the Figure 1 below. Q_1 designates the queue for the requests of fixed number, and in this case requests also mean sessions. When a client receives the response from the server, it issues another request after certain think time which is Z , i.e., after spending some time at Q_1 .

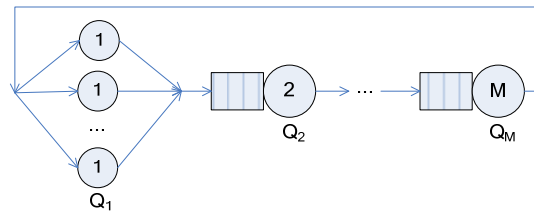


Figure 1. The closed queuing model with M tiers

This model can be efficiently solved using Mean-Value Analysis (MVA), a classic algorithm for solving closed queuing network. As for the computing cost and applicability, the MVA algorithm has several desirable features, such as simplicity, generality, etc [3]. As for the monitoring cost, the parameters required by the MVA algorithm are easy to get.

B. Open Queuing Network

In this paper, we use the open QN model with admission control. The model consists of a combination of a Markov Chain and a closed queuing network (QN) model. Let $X(k)$, $k = 1, \dots, N$ be the rate at which the computer system composed of the CPU and DISK completes its execution when there are k requests in execution, and these rates are the result of the closed QN Model solved by the MVA algorithm mentioned above (see Section III-A).

The diagram illustrates a discrete-time Markov chain with states $0, 1, \dots, J, J+1, \dots, N$. The transitions and their associated probabilities are as follows:

- From state 0 to state 1 : N/Z
- From state 1 to state 0 : $(N-1)/Z$
- From state 1 to state 2 : $X(2)$
- From state 2 to state 1 : $X(1)$
- From state J to state $J+1$: $(N-J+1)/Z$
- From state $J+1$ to state J : $(N-J)/Z$
- From state $J+1$ to state $J+2$: $X(J)$
- From state $J+2$ to state $J+1$: $X(J)$
- From state $N-1$ to state N : $1/Z$

We stress that open queuing network (QN) model with admission control is more realistic because computer systems always has limited capacity.

In order to evaluate the approach described in the previous sections, we use a testbed of an e-commerce site that simulates the operation of an online bookstore, according to the classic TPC-W benchmark [1]. The specifics of the software/hardware used in the experiment are given in Table 2 below.

	Client	Server
Processor	Intel(R) Pentium(R) D CPU 2.80GHz	Intel(R) Pentium(R) D CPU 2.80GHz
RAM	2GB	2GB
OS	FC10	FC10
Software	Emulated- Browsers	Tomcat6.0.18 Mysql5.0.77

According to TCP-W specifications, the number of customers or emulated browsers (EBs) is kept constant throughout the experiment. For each EB, TCP-W statistically defines the user session length, the user think time, and the queries that are generated by the session.

TPC-W defines 14 different Web interactions which are coarsely classified as either browsing or ordering. According to the weight of each type of activity in a given traffic mix, TPC-W defines 3 types of traffic mixes, namely, the browsing mix, the shopping mix and the ordering mix. We use the default three workload mixes.

As mentioned before, more detailed workload models that capture multi-class behavior can be more effective than single class workloads. So we classify the workload into 14 classes according to the transaction type of the web interactions defined by TPC-W. In order to compare the correctness influenced by the number of classes, we also test with 2 classes and 6 classes. As for the resource, we consider both the CPU and the DISK, and the regression method is the OLS. The utilization of CPU and DISK are obtained by using sysstat package in Linux, and the rest data are got by analysing the log provided by the application.

1) Let p_{ij} be the probability of the transaction type i following the transaction type j in the same client session, where $1 \leq i, j \leq 14$.

2) Let P be the probability matrix of the transition probabilities of all the transaction types, where P is the 14-order matrix and the element p_{ij} is just as mentioned above.

3) Let $\pi = [\pi_1, \dots, \pi_{14}]$ be the vector of stationary probabilities of the transactions, i.e., $\pi P = \pi$ and $\pi e = 1$, where e is a column vector of 1s with the appropriate dimension.

After calculating the transaction mix distribution vector π , the overall service demands are given by $S_{cpu} = \sum_{i=1}^{14} \pi_i * C_{i,cpu}$

and $S_{disk} = \sum_{i=1}^{14} \pi_i * C_{i,disk}$ where $C_{i,cpu}$ is the CPU cost and $C_{i,disk}$ is the DISK cost for class i.

Then we use the value of S_{cpu} and S_{disk} in the constructed open and closed analytical models to calculate the performance numbers. The closed queuing network has two queues. One queue is for CPU resource and another is for DISK resource.

1) The Analysis for Sensitivity of Regression to the Number of Classes

In order to analyse the effectiveness of the regression method with different number of classes, we separately discuss the conditions of light-load, moderate-load and heavy-load. We use the relative error to validate the regression accuracy. For every i -th test, the relative error of approximated utilization is

defined as: $Error_{CPU_R} = \frac{|U'_{CPU,j} - U_{CPU,j}|}{U_{CPU,j}}$ and

$$Error_{DISK_R} = \frac{|U'_{DISK,j} - U_{DISK,j}|}{U_{DISK,i}} \text{ (see Section II).}$$

For each workload mix, we all ran experiments for 50 times. The numbers of EBs for browsing, shopping and ordering mix is set as 70, 100 and 100 respectively under the moderate-load condition. These values are respectively 10, 15 and 20 under

the light-load condition. These values are respectively 80, 150 and 300 under the heavy-load condition. Each experiment ran for 20 minutes. The first 1 minute and the last 1 minute are considered as warm-up and cool-down periods, thus omitted in our analysis.

The following Figure 3 shows the CDF of the relative error for the CPU utilization under different workload condition. I omit to show the figure for the DISK utilization since it has the similar form as the CPU utilization. But in my case, the precision of DISK is a little bit worse than that of CPU.

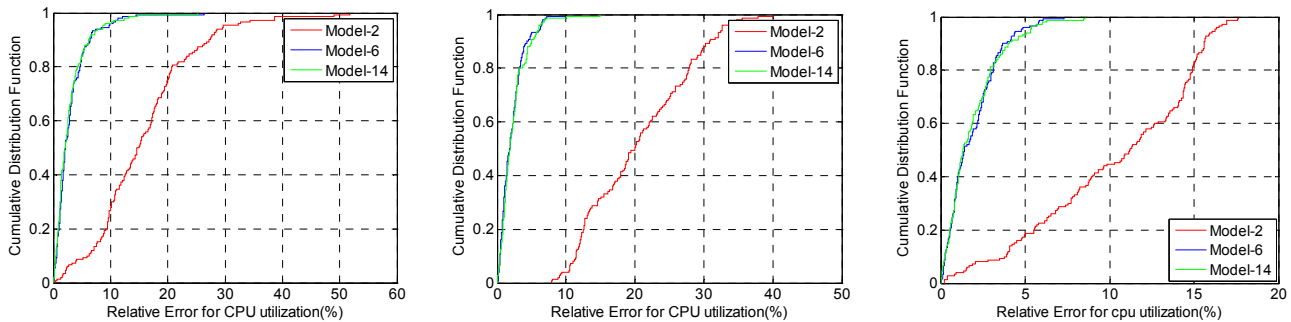


Figure 3. CDF of relative error of CPU utilization, left for light-load, middle for moderate-load, right for heavy-load

There results can be summarized as follows:

- Generally, the result is the best under the moderate-load condition and is the worst under the heavy-load condition.
- The precision of 2 classes is apparently worse than the case of 6 classes and 14 classes, we can also conclude

that when the number of classes is increased to a certain extent, the continual increase of the number of classes has no apparent improvement on the precision. In the case of moderate-load, the relative error greater than 90% for the CPU utilization is about 5%.

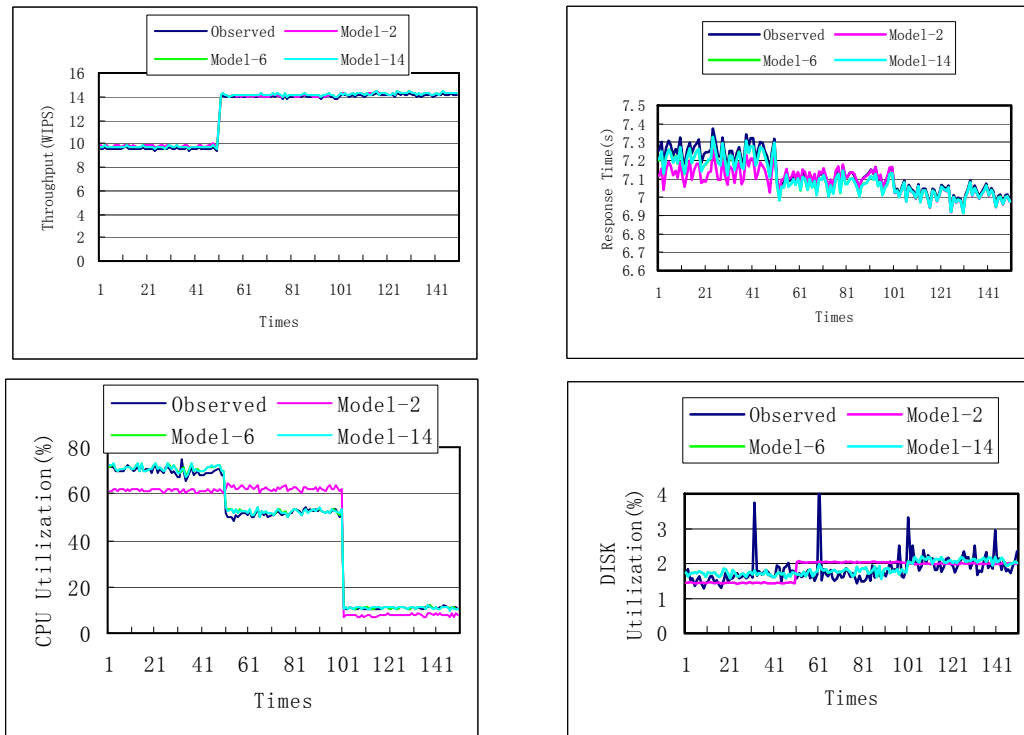


Figure 4. Closed queuing model validation in the case study

2) Performance Evaluation

For the performance evaluation, we only discuss the moderate-load case since the result is the best. For each workload mix, we all ran experiments for 50 times. The numbers of EBs for browsing, shopping and ordering mix is set as 70, 100 and 100 respectively. Each experiment ran for 40 minutes. The first 5 minutes and the last 5 minutes are

considered as warm-up and cool-down periods, thus omitted in our analysis.

Let's consider the closed queuing network solved by MVA algorithm first. The comparison between observed and modeled performance values is presented in Figure 4 above. The dark blue line is for the observed performance values and the pink line is for the modeled ones with 2 classes. The green line is the

result of 6 classes and the blue green is for the 14 classes. Overall, the results obtained from the analytic performance model imply that the models track the measurement reasonably well enough, and the detailed workload classification gives better matching. In the case of 14 classes, the overall average relative error for throughput, response time, CPU utilization and DISK utilization is 1.27%, 0.38%, 2.36% and 10.23% respectively. The correspondent values for 2 classes are 1.46%, 0.63%, 20.47% and 15.14% respectively, and the values for 6 classes are 1.27%, 0.38%, 2.19%, and 10.21%. From the figure, we can see that the 6 classes and 14 classes get the very close result just in accordance with the discussion above.

From the figure, we also notice that the extreme value observed in the figure of DISK utilization which corresponds to the concentrated access to the disk. This is the normal case for the TPC-W application, the DISK has more complex use pattern compared with the use of CPU.

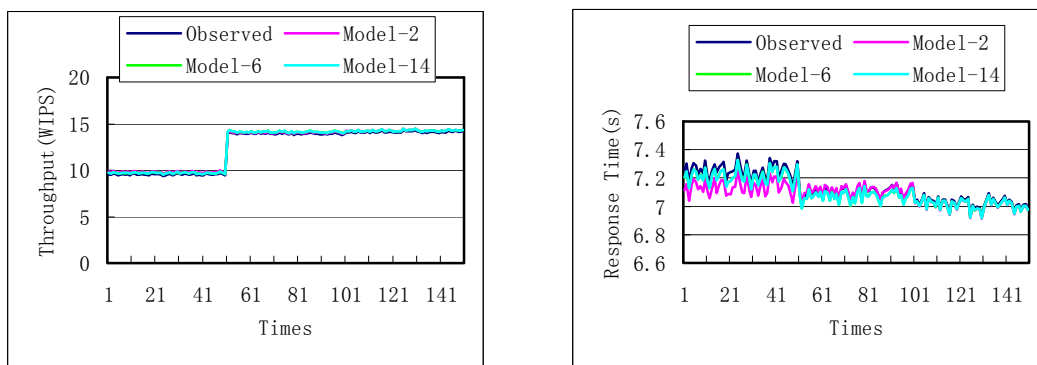


Figure 5. Open queuing model validation in the case study

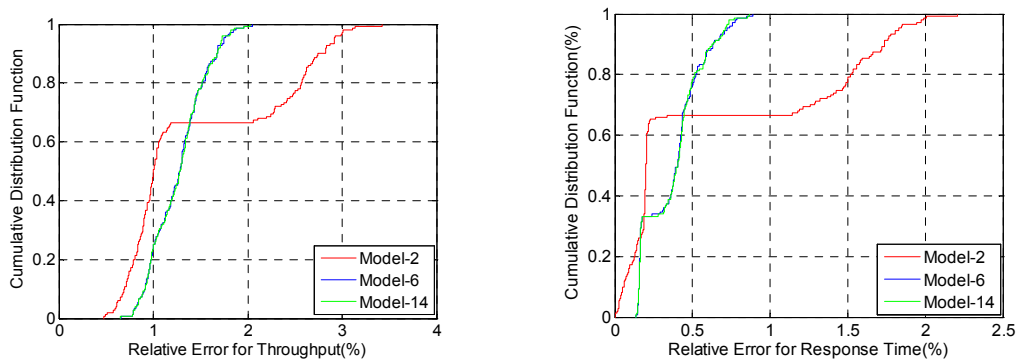


Figure 6. CDF of relative error for the throughput and response time

From the analysis above, the use of classification method and regression methodology for the extraction of characteristic parameters and the simple constructed analytical models are acceptable since this method is simple, general, low cost and high correctness. So, we deduce that this analysis result is acceptable and can be the basis which drives self-adaptive resource provisioning in shared data centres.

V. RELATED WORKS

For this complex service process, the use of Poisson model is not sufficient and we need more complex model such as MAP (Markov Arrival Process). Nevertheless, for the online performance evaluation, the approximate 10% of average relative error is acceptable (under 6 or 14 classes).

We show the result of the open queuing network in the Figure 5 below. The using of Markov method gets the similar result as in the closed QN. The overall average relative error for the throughput and response time is 1.27% and 0.38% respectively in the detailed case of 14 classes. We also provide the cumulative distribution function in Figure 6 below, in it the red one with detailed workload classification of 14 classes and the dark blue one with less detailed workload classification of 6 give more accurate results. We can also see that for the detailed case, the relative error is really small and for the case of throughput, almost 100% tests show relative error less than 2%. For the case of response time, the precision of analysis is even better.

A. Online Extraction of Characteristic Parameters

This includes extracting the static and dynamic characteristic parameters of workload, and we only focus on the static parameters. [3]When extracting the service demands of requests, it uses the response time to approximately substitute them. So this method has limitation and is only suitable for light load condition. It also gives a simple improved method used in the condition of heavy load. Furthermore, the parameters needed are much more and complex compared to the method in this paper. [2]This paper adopts the similar

concept as ours, it also use regression analysis to get the service demands of requests. But it only considers the resource of CPU, so this method is only suitable for computing-intensive applications. [4]The parameters needed to extraction are complex, for example, σ_a^2 and σ_b^2 which are the variance of inter-arrival time and variance of service time. Moreover, it need to analysis offline to breaking down the end-to-end response time into tier-specific response time, so it is not really suitable for online analysis.

Contrastively, the method used in this paper synthetically considers the resource of CPU and DISK. The OLS regression method is simple and it "absorb" some level of uncertainty or noise present in real-world data by effectively "compacting" information on workload demands within a few model parameters only. Moreover, the methodology is not intrusive and is based on monitoring data that are typically available in enterprise production environments.

B. Analytical Models

Analytic models are widely used for its quickly supplying performance numbers. [4]Bhuvan Urgaonkar uses the G/G/1 queuing model, but it assumes that servers are homogeneous and that servers in each tier are load-balanced. [3]It adopts single-class/multi-class closed queuing network solved by MVA algorithm, and it is only suitable for light load condition. [2]Qi Zhang uses the single-class closed queuing network and applied to the computing-intensive applications. [5][6]XiaoYing WANG combines the time-domain queuing model with M/M/c queuing model. [9][10]Terence Kelly uses the black box model solved by regression methodology. [8]It also uses multiclass open and closed queuing network models to get the performance values, but it need to know the utilization of resources due to every class which is actually hard to get and calculate online.

In this paper, we construct both open and closed queuing network, and in the open queuing network. We use the results of the regression method to parameterize the analytical models of queues. Furthermore, we partition the workload into classes to improve representativeness of the characterization and increase the predictive power of the workload. But when using the characteristic parameters in the analytic models, we use them trickily with the idea of transition probability matrix (see Section IV-B).

VI. CONCLUSIONS AND FUTURE WORK

Self-adaptive resource provisioning in shared data centres requires swift responses to workload changes. The complication of applications both in scale and architecture makes this a huge challenge. Our point is that effective analytic models can quickly supply performance numbers, which then can provide powerful and simple solutions for dynamic resource provisioning. We provide a simple, low-cost and precise method based on regression methodology to extract the characteristic parameters used in analytical models. In order to improve the representative characterization of workload, we classify the workload into classes and use them trickily with the idea of transition

probability matrix. With respect to the method used for performance evaluation, we construct both open and closed queuing network to evaluate the correctness and the generality of our method. Regardless of the system status, our experiments show that the values produced by the performance model are in excellent agreement with experimental data.

There are many works in which we can go further. We are not considering the architecture of applications since multi-tier or distributed architecture has become the industry standard for implementing scalable client-server applications. As for the resource, the NIC resource is not included in our analysis, and moreover the open and closed analytical models described in this paper do not account for queues with multiple resources even nowadays servers with multiple processors and disks are common. We can also evaluate our method with genuine multi-class open/closed queuing network since we use our classified workloads with a tricky. In the end, we focus our work on the performance evaluation which is the basis for the dynamic resource provisioning, and in the future we can integrate them in the shared data center to drive self-adaptive resource provisioning.

ACKNOWLEDGMENT

This work was supported by Major State Basic Research Development Program of China (No. 2004CB318205) and National High Technology Research and Development Program of China (No. 2007AA01Z402). We would also like to thank the anonymous reviewers for their comments.

REFERENCES

- [1] TPC-W Benchmark. URL <http://www.tpc.org>
- [2] Qi Zhang, Ludmila Cherkasova, Evgenia Smirni, "A Regression-Based Analytic Model for Dynamic Resource Provisioning of Multi-Tier Applications", Fourth International Conference on Autonomic Computing(ICAC'07), 27-27, June 2007
- [3] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi, "An Analytical Model for Multi-tier Internet Services and its Applications", Proceedings of the ACM SIGMETRICS'2005, Banff, Canada, June 2005
- [4] B. Urgaonkar, Prashant Shenoy, Abhishek Chandra, Pawan Goyal, "Dynamic Provisioning of Multi-Tier Internet Applications", Proceedings of Second International Conference on Automatic Computing 2005(ICAC 2005), 217-228, June 2005
- [5] XiaoYing Wang, DongJun Lan, Gang Wang, et al., "Appliance-based Autonomic Provisioning Framework for Virtualized Outsourcing Data Center", Proceedings of the 4th IEEE International Conference on Autonomic Computing(ICAC'2007), Florida, USA, 2007
- [6] XiaoYing Wang, ZhiHui Du, YiNong Chen, et al., "An Autonomic Provisioning Framework for Outsourcing Data Center Based on Virtual Appliances", Cluster Computing, Volume 11, Issue 3, 229-245, September 2008
- [7] D.A. Menasce, V.A.F. Almida, "Capacity Planning for Web Services: Metrics, Models, and Methods", Prentice Hall, 2002
- [8] M. N. Bennani and D. A. Menasce, "Resource Allocation for Autonomic Data Centers Using Analytic Performance Models", Proceedings of the 2nd IEEE International Conference on Autonomic Computing(ICAC'2005), Seattle, WA, 2005
- [9] HP Labs Technical Reports, "Predicting Performance in Distributed Enterprise Applications", <http://www.hpl.hp.com/techreports/2006/HPL-2006-76.html>
- [10] Terrence Kelly, "Detecting Performance Anomalies in Global Applications", Second USENIX Workshop on Real, Large Distributed Systems (WORLDS 2005)