Reputation-based QoS Provisioning in Cloud Computing via Dirichlet Multinomial Model

Yanping Xiao, Chuang Lin, Yixin Jiang
Department of Computer Science and Technology
Tsinghua University
Beijing, China, 100084
{ypxiao, clin, yxjiang}@csnet1.cs.tsinghua.edu.cn

Xiaowen Chu
Department of Computer Science
Hong Kong Baptist University
Hong Kong, P. R. China
chxw@comp.hkbu.edu.h

Xuemin (Sherman) Shen
Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
xshen@bbcr.uwaterloo.ca

Abstract—**In Cloud computing, users with different service requirement often need to negotiate with service providers subject to Service Level Agreement (SLA). The unique pay-as-you-go billing manner and different virtualization levels of Cloud computing present challenges to resource provisioning for service providers. In this paper, based on the Dirichlet multinomial model, we present an efficient reputation-based QoS provisioning scheme for Cloud computing, which can minimize the cost of computing resources, while satisfying the desired QoS metrics. Unlike the previous counterparts, we consider the statistical probability of the response time as a practical metric rather than the typical mean response time. We also present an optimization algorithm to balance performance and computing cost. Numerical results show the efficiency and effectiveness of the proposed scheme.**

Keywords—***Reputation; QoS Provisioning; Cloud computing***

I. INTRODUCTION

Cloud computing [1] is undoubtedly the most promising new paradigm, since it has the potential to make everything more attractive as a service [2]. The service, also called "Cloud service," is delivered through next-generation data centers that are built on compute and storage virtualization technologies. Users can access application and data from a "Cloud" anywhere in the world on demand through a service provider, who takes charge of running resource provisioning algorithms to provide virtual computing resource for users according to Service Level Agreements (SLAs). Cloud computing services should be highly available, scalable, and autonomic to support ubiquitous access, dynamic discovery and composition. In such a complex environment, how to make QoS provisioning for different user requests is a big challenge.

However, little efforts focus on QoS provisioning in Cloud computing especially for workflow-based composite service. The existing works focus on one individual service, and QoS assurance only for individual services is not enough to meet the requirement of workflow-based composite services [3]. Moreover, more and more Cloud services with the same function will be provided by different data centers/Cloud providers. Different data centers offer different QoS, and one data center can also offer different QoS with different charge model. Therefore, How to select the service from the available service sets attached to different candidate data centers and to make resource provisioning should be considered in order to satisfy QoS requests.

To select the most promising service from the fittest service sets, we introduce the reputation mechanisms to achieve it. "Reputation" [7, 12], as a security mechanism, has been deployed in many successful commercial online applications. The mathematical model of reputation is based on the Dirichlet distribution, which allows multiple graded ratings to be expressed directly in the derived reputation scores. In this paper "Reputation" is used to describe the service competence of a data center that acts as it is expected. The reputation of a data center is an indicator of QoS and Quality of Protection (QoP) provided by the data center based on the task completion experience. It often predicates the future behavior of the data center. Therefore, we establish multi-parameter Bayesian model (Dirichlet multinomial model) to analyze data centers' behavior, which can define any set of discrete rating levels and provide great flexibility and usability.

Besides, minimizing the user service cost is also another concern. To minimize the total cost of the required computing resource while satisfying QoS requirements, in this paper we propose an efficient QoS provisioning scheme for Cloud computing. Our main contributions can be summarized as follows. First, a Reputation-based QoS provisioning model is proposed for Cloud computing paradigm, as well as a feasible provisioning algorithm, which can help Cloud service providers optimize resource allocation. Second, a Reputation management framework for Cloud services is presented, which can assist service providers to select the promising service sites from multiple service sites. Finally, an efficient solution to probability distribution of service response time is introduced to evaluate the M/M/C/∞ queue and tandem network model with multi-nodes for Cloud computing. To the best of our knowledge, it is the first study to consider the resource provisioning under cost, response time, and reputation constraints in Cloud Computing.

The rest of the paper is organized as follows: Section II proposes the architecture, model and algorithm of Reputation-based QoS provisioning in Cloud computing. Section III gives the simulation results, followed by conclusions in Section IV.

II. THE SCHEME OF REPUTATION-BASED QOS PROVISIONING

A. The Achitecture

In Cloud computing [8], the users and service providers interact through SLA [4-6]. A typical reputation-based QoS provisioning architecture is shown in Fig. 1.

The left ellipse frame denotes Cloud users who generate a stream of service requests. The middle ellipse frame denotes a Cloud computing service provider consisting of six components: service selector, service broker, task dispatcher, service monitor, reputation management and billing. The right ellipse frame denotes infrastructure providers consisting of physical/virtual devices which make use of "virtualized" technology such as VMware hypervisor to dynamically provide computing service and storage service on demand.

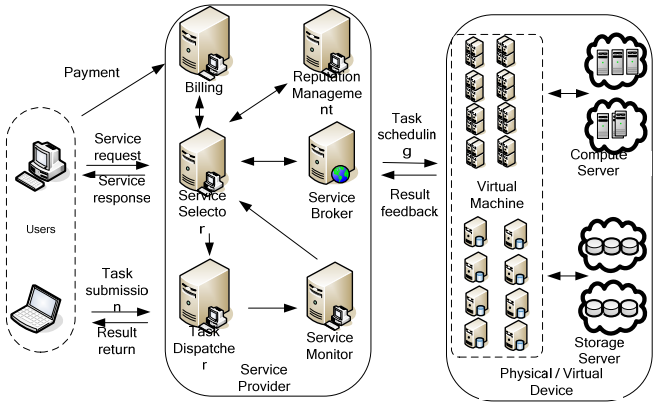

Figure.1 A Reputation-based QoS Provisioning Architecture

When users need to execute a task in the Cloud computing data centers, it first submits service request (e.g., workflow with QoS and QoP requirement) to a service provider. The service broker is in charge of discovering available Cloud computing service resources. Then service selector takes the responsibility of selecting appropriate service from all the available service sets according to some metrics such as the reputation information and makes resource provisioning. If users' requirements can be satisfied by the available services, service provider will inform the task dispatcher to be ready to dispatch the user's task, and at the same time service monitor will monitor the status of service execution. When the service providers accomplish the designated tasks, they send the results back to the users. The billing service is in charge of the fee according to used resources per unit time. The reputation management service is in charge of updating the reputation value of services according to the users' feedback and other reports constantly and providing decision service for service selectors.

B. *The QoS Metrics*

Reputation: The reputation of Cloud services is constructed by Dirichlet multinomial model based on Dirichlet distribution, which can assist service providers to select the most promising service sites to participate the computing.

The Dirichlet distribution is defined as follows: let $\Theta = \{\theta_1, \cdots, \theta_k\}$ be a state space consisting of mutually disjoint events. Let $\vec{p} = (p(\theta_1), \cdots, p(\theta_k))$ be a continuous random vector in the *k-dimension* simplex with the joint PDF

$$f(\vec{p} \mid \vec{\alpha}) = \frac{\Gamma(\sum_{i=1}^{k} \alpha(\theta_i))}{\prod_{i=1}^{k} \Gamma(\alpha(\theta_i))} \prod_{i=1}^{k} p(\theta_i)^{\alpha(\theta_i)-1},$$

where $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$ is the Gamma function. Then $\vec{p}$ is said to have a *k-dimension* Dirichlet distribution with parameter vector $\vec{\alpha} = (\alpha(\theta_1), \cdots, \alpha(\theta_k))$, $(\alpha(\theta_i) \geq 0, \ for\ i = 1, \cdots, k)$. The Dirichlet distribution is the multivariate generalization of the Beta distribution and the vector of expectations is the function of parameters $\alpha(\theta_i)$, that is

$$E(p(\theta_i) \mid \vec{\alpha}) = \frac{\alpha(\theta_i)}{\sum_{i=1}^{k} \alpha(\theta_i)}.$$

Since the Dirichlet distribution is a conjugate prior of the multinomial distribution, the posteriori distribution is also Dirichlet and can be calculated as follows [16]:

$$f(\vec{p} \mid \vec{r}, \vec{\alpha}) = \frac{\Gamma(\sum_{i=1}^{k} (r(x_i) + C\alpha(x_i)))}{\prod_{i=1}^{k} \Gamma(r(x_i) + C\alpha(x_i))} \prod_{i=1}^{k} p(x_i)^{(r(x_i)+C\alpha(x_i)-1)} \quad (1)$$

where

$$\begin{cases} p(x_1), \cdots, p(x_k) \geq 0, \sum_{i=1}^{k} p(x_i) = 1; \\ \alpha(x_1), \cdots, \alpha(x_k) > 0, \ \sum_{i=1}^{k} \alpha(x_i) = 1, \end{cases}$$

where $\alpha(\theta_i)$ is a *base vector* over the state space Θ, C is *a priori* constant which is equal to the cardinality of the state space over which a uniform distribution is assumed (C is usually set to 2), and the vector $r(\theta_i)$ is a *posterior* evidence over the state space Θ. Given a Dirichlet distribution of Eq.(1), the probability expectation of any of the k variables can be:

$$E(p(\theta_i) \mid \vec{r}, \vec{\alpha}) = \frac{r(\theta_i) + C\alpha(\theta_i)}{C + \sum_{i=1}^{k} r(\theta_i)}$$

Fig. 2 shows a framework for reputation management. The reputation of the data center/service site is evaluated by a trusted agent who updates the reputation value according to the collected user feedback information. In this paper we focus on the user feedback only, and the user feedback can be any level in a set of predefined rating levels. The more categories of users' feedback, the more complicated the implementation. Thus, we could consider three categories of feedback: unsatisfactory, basic satisfactory and satisfactory.

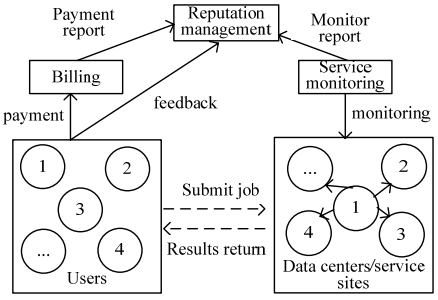

Figure.2 A Framework for Reputation Management

We take each interaction result as an indicator to distinguish different categories, and introduce the following parameter ζ as a metric to judge the disjoint emerging events, which is defined as follows:

$$\zeta = F(t)\big|_{t=T^B} / F(t)\big|_{t=T^{B'}} = \int_0^{T^B} f(t)dt \Big/ \int_0^{T^{B'}} f(t)dt, \quad (2)$$

where $T^{B'}$ is the actual response time of data center j, and T^B is the desired response time.

In this paper we adopt the value 1.2 as a determination metric; of course, we can select different value as needed. If $\zeta \geq 1.2$, the rating is satisfactory. If $1 \leq \zeta < 1.2$, the rating is basic satisfactory. If $\zeta < 1$, it means that the interaction fails and gets an unsatisfactory rating.

$$r = \begin{cases} unsatisfactory & if\ \zeta < 1 \\ basic\ satisfactory & if\ 1 \leq \zeta < 1.2 \\ satisfactory & if \zeta \geq 1.2 \end{cases}.$$

To establish multiple-parameter Bayesian model to analyze the reputation of service sites, let us consider the discrete time t_k $(k = 1,2,\cdots)$ in an increasing order, and let the vector $\vec{r}_{j,t_k}$ be the total accumulated ratings of data center j calculated by all the customers in period t_k. Specifically, it is the sum of all ratings $\vec{r}_j^{\,x}$ of data center j by all the customer c within that period, expressed by:

$$\vec{r}_{j,t_k} = \sum_{x \in M_{j,t_k}} \vec{r}_j^{\,x} \tag{3}$$

where M_{j,t_k} is the set of all the customers who rated data center j during period t_k.

Let the total accumulated ratings of data center j after the time period t_k be denoted by $\vec{R}_{j,t_k}$, then the new accumulated rating after time period t_{k+1} can be expressed as:

$$\vec{R}_{j,t_{k+1}} = e^{(-t_{k+1}+t_k)} \cdot \vec{R}_{j,t_k} + \vec{r}_{j,t_{k+1}} \tag{4}$$

Eq. (4) represents a recursive updating algorithm that can be executed in every period for all the data centers. It reflects the latest interaction and the past interaction. The latest feedback dominates the reputation indicator. The longer the elapsed time is, the less it has effects on the reputation.

After obtaining the aggregated rating, we can define the reputation indicator as a function of the probability expectation values of each element in the state space. The multinomial probability reputation indicator vector $\vec{s}$ is defined as follows:

$$\vec{S}_{j,t_{k+1}} = \frac{\vec{R}_{j,t_{k+1}} + C\vec{\alpha}}{C + \sum_{i=1}^{k} \vec{R}_{j,t_{k+1}}} \tag{5}$$

Where C is set 2, and $\vec{a}$ is the base rate vector over the state space. The reputation indicator $\vec{S}_{j,t_{k+1}}$ can be interpreted like a multinomial probability measure as an indication of how a data center is expected to behave in future transactions. It can easily be verified that $\sum_{i \in State\ space} \vec{S}_{j,t_{k+1}}(i) = 1$.

Service Cost: Cloud computing is available in a pay-as-you-go manner, which involves metering usage and charging, independently of the time period over which the usage occurs. Generally, there are M sites attached to different data centers in Cloud computing, which provide services such as computing, storage, networking, or some other services. For clarity, we assume that each site only provides one type of service associated with cost c_i in virtualization mode.

Let N_i denote the available number of virtual servers at site $i\,(i = 1, 2, \cdots, m)$, $m \in M$, then the cost optimization can be quantified by solving the following optimization problem:

$$C = \min_{n_1, n_2, \cdots, n_m} (n_1 c_1 + n_2 c_2 + \cdots + n_m c_m) \tag{6}$$

where $n_i \in [1, N_i]$ denotes the number of virtualized servers allocated at site i. Eq. (6) is subject to cost constraints in SLA.

Response Time: Cloud computing could assign more virtual servers for users to improve performance. The response time, as a key performance metric, is typically considered through its mean. However, it may not be sufficient to reflect users' QoS requirements in some cases, where they may be more interested in a statistical bound of the response time [5]. Therefore, we take the probability of the response time being less than a predefined value as a performance metric.

Let T be a random variable denoting the response time, and let $f(t)$ and $F(t)$ be its probability and cumulative distributions respectively. Also let T^B be the desired response time which is negotiated by a customer and the service provider based on a fee paid by the customer. Then, the SLA response time metrics can be denoted as follows.

$$F(t)\Big|_{t=T^B} = \int_0^{T^B} f(t)dt \geq \sigma \tag{7}$$

Eq. (7) represents that the probability of the response time being less than T^B should be no less than σ.

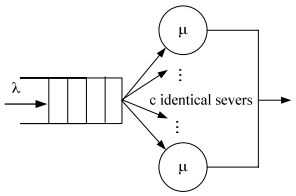

Figure 3. An $M/M/c/\infty$ queue system

Let us consider an $M/M/c/\infty$ queue with an arrival rate λ, and c parallel identical virtual servers, each with service rate μ. Customers are served in order of arrival. Suppose that the occupation rate per server, $\rho_c = \lambda/(c\mu)$, satisfies $\rho_c < 1$.

Let ρ be λ/μ, the steady-state probability of the system can be calculated as

$$\begin{cases} p_0 = \left(\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{c\rho^c}{c!} \cdot \frac{1}{c-\rho} \right)^{-1} \\ p_n = \frac{\rho^n}{n!} p_0,\ n = 1, \cdots, c-1 \\ p_{c+n} = \rho^{n+c} \frac{1}{c^n c!} p_0,\ n = 0, 1, 2 \cdots \end{cases}$$

The response time T is exponentially distributed with parameter $r = 1 \Big/ \left(\frac{\rho_c}{\lambda(1-\rho_c)^2} \cdot p_c + \frac{1}{\mu} \right)$, i.e., its probability distribution is given by $f_T(t) = re^{-rt}$.

Using the definition given in Eq. (7), we have

$$F_T(t)\Big|_{t=T^B} = 1 - e^{-rT^B} \geq \sigma \quad (8)$$

or $$r \geq -\ln(1-\sigma)/T^B .$$

This means that to guarantee higher SLA service levels, for a given arrival rate λ, c parallel virtual servers with service rate μ, and the desired response time T^B, we can adjust the parameter c to guarantee the probability of the response time of the service being less than T^B is at least σ.

C. The Proposed Model

Each data center accomplishes some particular functions. Workflow-base services are usually provided by different data centers. The tasks submitted by the user often need to be allocated in such sites, so we consider a common tandem workflow-based queue model to examine our system.

As shown in Fig. 4, the proposed model consists of a service provider and m sites from different centers numbered sequentially from 1 to m. The period of executing a task is the time it takes for a task to traverse the whole tandem queue network. The first node represents the service provider who is in charge of selection of sites, and distributing the jobs to the sites and forwarding the results back to the users. We do not model the propagation delay, because it depends on network traffic and data size. We also assume each site never delay its work on purpose. And the reply from each data center back to the service provider and to the user is not modeled explicitly.

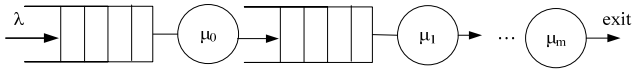

Figure 4. A tandem queue model

If the service provider can meet the SLA requirement, it will distribute the jobs to the sites attached to different data centers. Each site I is modeled as a single FIFO queue served by n_i identical server instances, each instance providing a service at the rate u_i. Let λ be the external arrival rate to the service provider, and λ_i be the effective arrival rates to the site $i\,(i = 1, 2, \cdots, m)$. We assume that all service time are exponentially distributed and the external arrival to the server provider occurs in a Poisson fashion.

In the following discussion each site is modeled as a single $M/M/c$ queue with arrival rate λ_i, each instance's service rate μ_i and total service rate $c\mu_i$. We also assume that the arrival rate λ_i is smaller than $c\mu_i$.

Since queue network is overtake-free [9], the waiting time of a customer at a site is independent of its waiting times at other sites [10]. Let T be the total delay from the user to the service provider and also from site 1 to m. Let Γ be the service time at the service provider and Γ_i be the time elapsed from the moment a customer arrives at site i to the moment it departs from the site. Then, the total response time is:

$$T = \Gamma + \Gamma_1 + \Gamma_2 + \cdots + \Gamma_m \quad (9)$$

The LST (Laplace-Stieltjes Transform) of response time T is

$$L_T(s) = L_\Gamma(s) L_{\Gamma_1}(s) L_{\Gamma_2}(s) \cdots L_{\Gamma_m}(s) \quad (10)$$

where $L_\Gamma(s)$ is the LST of the service time Γ. given by

$$L_\Gamma(s) = \frac{\mu_0(1-\rho_0)}{\mu_0(1-\rho_0)+s} \quad (11)$$

And $L_{\Gamma_1}(s)$ is the LST of the response time Γ_i at the *i-th* site, which can be calculated as

$$L_{\Gamma_i}(s) = \frac{r_i}{r_i + s} \quad (12)$$

where $r_i = \dfrac{1}{\dfrac{(\rho_c)_i}{\lambda_i[1-(\rho_c)_i]^2}\cdot(p_c)_i + \dfrac{1}{\mu_i}}\,(i = 1,2,\cdots,m)$. From Eq. (10), Eq. (11), and Eq. (12), we have

$$L_T(s) = \frac{\mu_0(1-\rho_0)}{\mu_0(1-\rho_0)+s} \prod_{i=1}^{m} \frac{r_i}{r_i + s} \quad (13)$$

We observe that $f(t)$ and $F(t)$ are usually nonlinear functions as the variable of t and n_i. Hence, the resource optimization is an m-dimensional linear optimization problem subject to nonlinear constraints. From Eq. (13), we have

$$f(t) = L^{-1}\left\{\frac{\mu_0(1-\rho_0)}{\mu_0(1-\rho_0)+s} \cdot \prod_{i=1}^{m} \frac{r_i}{r_i + s}\right\} \quad (14)$$

And the cumulative distribution function is

$$F(t) = L^{-1}\left\{\frac{\mu_0(1-\rho_0)}{(\mu_0(1-\rho_0)+s)s} \cdot \prod_{i=1}^{m} \frac{r_i}{r_i + s}\right\} \quad (15)$$

It can be verified that the output of one $M/M/1$ queue follows Poisson distribution at the rate of its arrival rate. Accordingly, the following equation holds.

$$\lambda = \lambda_i \quad (16)$$

From Eq. (15), it can be concluded that $F(t)$ is a function of variable n_i. Thus we can find n_i in the m-dimensional optimization problem:

$$n_i \leftarrow \arg\min F(t)\Big|_{t=T^B} \quad (17)$$

subject to the constraint $F(t)\Big|_{t=T^B} \geq \sigma$, where $F(t)$ is given by Eq. (15).

D. The Algorithm

For clarity, assume a user task consists of m subtasks and each subtask also has $N_i, i \in [1,m]$ candidate service sites, thus there are totally $\prod_{i=1}^{m} N_i$ possible service combinations for the required services, and the solution to compare all the service combinations is too time and memory consuming to be feasible. Accordingly, we present our reputation-based QoS provisioning scheme in Algorithm 1. The algorithm attempts to minimize the overall cost of the service while satisfying QoS requirements.

The first step of the algorithm is to select the most potential service sites after the service decomposition. The selected service site should satisfy certain pre-define reputation. As shown in Algorithm 1, the expectation of "satisfactory" and "basic satisfactory" events should be no less than S^B.

The following steps are virtual server provisioning problem under the condition of predefined service cost and response time. The problem can be mapped into a multi-objective programming problem, including service cost, response time, and some other required metrics. The core of the algorithm is a non-linear programming problem, which can be solved by widely-used mixed integer programming solvers. In this paper we transform the multi-objective programming into a single-objective programming problem through defining the performance cost ratio η metric, which is defined as follows:

$$\eta = response\ \ time/Cost \tag{18}$$

Algorithm 1:Rep-QoS-Pro(Prov., job=($\vec{S}^B$,T^B,σ,C^B))

Input: users submit a *job* to a service provider at time τ ; the service provider requests resource from all candidate sites from different data centers to execute this job.

Output: workload distribution if it meets user's QoS and reputation requirements. Otherwise, print "fail".

(1) Finding m sites from all the data centers satisfying at time t_k , $S_{i,\tau_k}(\theta) \geq S^B$, where θ denotes the "satisfactory" and "basic satisfactory" events in the state space.

(2) Finding $n_i (i = 1,2,\cdots m)$ that satisfies the equation from m sites.

$$\arg\min F(t)\big|_{t=T^B} \geq \sigma$$

(3) Solving for $n_i (i = 1,2,\cdots m)$ in the m-dimensional mixed integer linear programming:

$$\arg\min(n_1c_1 + n_2c_2 + \cdots + n_mc_m) \leq C^B$$

(4) Finding the most optimal solution from the intersection between Step. (2) and Step. (3).

With computing the performance cost ratio, we can acquire some relative optimized solutions in all the solution set, which will be demonstrated through some cases in the next section. The complexity of the algorithm is related to the node numbers of the tandem network. Assuming that the node number is m, and each node has n_i server instances, then the time/space complexity is $O(\sum_{i=1}^{m} n_i)$. However, if we can fix parameters n_i in a certain range, the time/space complexity will be further reduced greatly. In the following section, we will introduce an example to demonstrate the existence.

III. NUMERICAL SIMULATION

In this section, we show the validity of algorithms through numerical simulations. We assign some specific parameters for a service model as shown in Fig. 4. Let m=5, λ=100, μ_0 =125, N_j=50, and C^B=460 at random. Some other alternative values can also be assigned. Each serve instance's rates and cost are listed in Table I and Table II, respectively.

TABLE I. THE SERVICE INSTANCE' RATES OF FIVE SITES

Service rates	μ_1	μ_2	μ_3	μ_4	μ_5
values	52	18	35	80	41

TABLE II. THE SERVICE INSTANCE COST OF FIVE SITES

Service Cost	c_1	c_2	c_3	c_4	c_5
values	18	7	15	32	21

To illustrate the procedure of reputation computation, let $\{n_1,\cdots,n_5\}$ be $\{3,7,4,4,5\}$, then we simulate it using Arena [11] and get a group value from the trace of response time at random matching to the expectation value of the response time, which are listed in Table III. Also let $\vec{\alpha} = [0.2, 0.5, 0.3]$ be the initial default reputation value for unsatisfactory, basic satisfactory and satisfactory. And then we utilize the Eq. (2), (4) and (5) to get a group value for 8 periods. The result is listed in the Table III, where S_1, S_2 and S_3 denote the reputation of three events during 1-8 periods respectively. During the selection of sites from data centers with the same function, the sites with higher reputation will be chosen with high probabilities.

TABLE III. THE SIMULATION VALUES AND REPUTATIONS FOR 1-8 PERIOD

Δt	**Simul.**	**Expect.**	ζ	**S_1**	**S_2**	**S_3**
1	0.4186	0.4210	1.003145	0.1333	0.6667	0.200
2	0.8760	0.8462	0.998082	0.4154	0.4065	0.178
3	0.5510	0.5252	0.985524	0.5045	0.3242	0.1712
4	0.2310	0.2026	0.798891	0.5356	0.2955	0.1689
5	0.6721	0.6910	1.004038	0.2668	0.5652	0.1680
6	0.1980	0.2460	1.437101	0.1686	0.3842	0.4472
7	0.3612	0.38	1.037308	0.1326	0.5971	0.2703
8	0.1828	0.168	0.831747	0.3986	0.3961	0.2053

Generally, it is not trivial to solve the optimization problems in Algorithm 1. But if we can get a bound of $\{n_1,\cdots,n_5\}$ in advance, the problem will be easy. In the following the validity of the bound will be exemplified. Although the method is incomplete, it can work and decrease the complexity of computing and storage greatly.

Fig. 5 shows the probability distribution of response time with service rate 52 and server numbers 2,3,4,5 respectively. From Fig. 5, it can be seen that as server number increasing, the mean response time is nearly invariable. But the corresponding cost will increase linearly. If the approximate order of the response time is 10^{-6} , the server number 10 can be as the upper bound. For the lower bound, it must satisfy the equation $\rho_c = \lambda/(c\mu) < 1$.

For a multi-level series networks with M/M/C/∞ queue, it has the similar conclusion. Fig. 6 shows three probability distribution of response time with $\{n_1,\cdots,n_5\}$ assigned by $\{10,16,12,9,12\}$, $\{2,6,3,2,3\}$, and $\{4,6,7,9,4\}$, respectively.

To extensively examine the relation between the response time and cost, we evaluate the response time, cost, performance cost ratio defined in Eq. (18), and utilization rate with different configuration parameters. Fig. 7 compares the response time, cost, performance cost ratio and utilization ratio of a single node with different server instance numbers in the condition that λ=100, μ=52 and σ =97.5%. Fig. 8 shows the response time, cost, performance cost ratio and utilization ratio of a tandem network with the same configuration in Fig. 6 under the condition σ =97.5%.

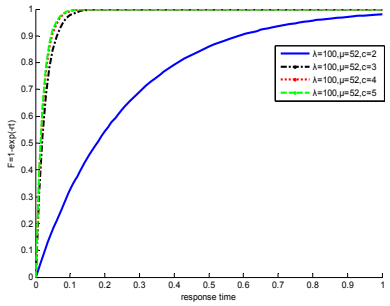

Figure.5 Probability Distribution of Response Time

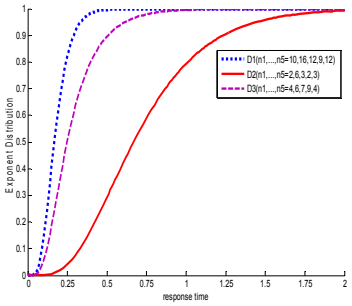

Figure.6 Probality Distribution of a Tandem Network

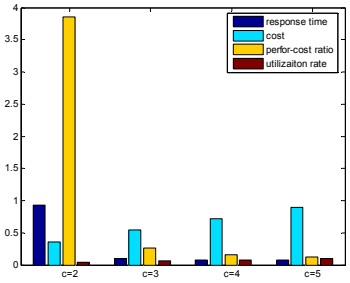

Figure.7 Res. time, Cost,Perform./Cost Ratio and Utilization with different number servers

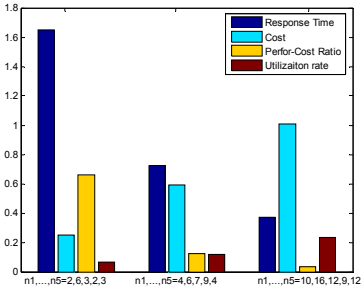

Figure.8 Respo. time, Cost, Perform./Cost ratio and Utilization for An Tandem Network

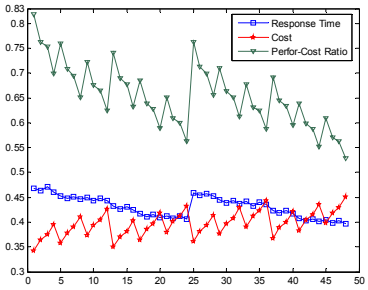

Figure.9 Respo. time, Cost, Perform./Cost ratio

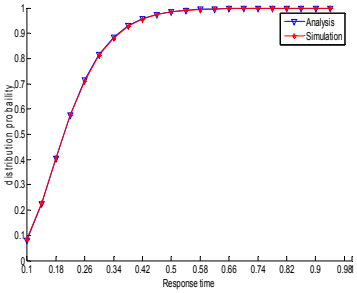

Figure.10 The cumulative distribution of resp time

Fig. 9 shows the curve of response time, cost and performance-cost ratio varies with the different server number when the response time is in the range of [0.4, 0.48] and σ =97.5%. From Fig. 9, there may be 48 group solutions satisfying the conditions, but from the performance cost ratio, the configuration $\{3,7,4,3,4\}$ for $\{n_1,\cdots,n_5\}$ can be deemed an optimization solution. If users present an explicit response time, for example 0.46 and its probability $>\sigma$ =97.5%, the configuration $\{3,7,4,4,5\}$ for $\{n_1,\cdots,n_5\}$ is the most approximate solution.

We use Arena to simulate the tandem queue network with the configuration $\{3,7,4,4,5\}$ for $\{n_1,\cdots,n_5\}$. As shown in Fig. 10, the simulation results are very approximate to the analysis results. Therefore, it can be verified that the algorithm is effective and operative.

IV. CONCLUSIONS

In this paper, we present an efficient reputation-based QoS provisioning scheme for Cloud computing, which can minimize the total cost of computing resources used by a customer, while satisfying the pre-defined response time. Unlike the previous counterparts, we consider the statistical bound of the response time as a more practical metric than the typically mean response time. We also present a reputation management framework for Cloud computing which can assistant service providers to select the promising service sites from multiple service sites. Finally, we present an optimization algorithm to make a tradeoff between performance and computing cost. In the future, we will introduce security and privacy metrics to the QoS and extensively investigate QoS provisioning algorithms.

REFERENCES

[1] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented Cloud Computing: Vision, hype, and reality for delivering it services as computing utilities," CoRR, vol. abs/0808.3558, 2008.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley view of Cloud Computing," Univ. of California, Berkeley, Tech. Rep., 2009.

[3] Mladen A. Vouk "Cloud Computing – Issues, Research and Implementations " *Journal of Computing and Information Technology*, -CIT 16, no.4, pp.235–246, 2008.

[4] S. Venugopal, X. Chu, and R. Buyya. "A Negotiation Mechanism for Advance Resource Reservation using the Alternate Offers Protocol." *Proc. of IEEE IWQoS*, June 2008

[5] K. Xiong and H. Perros. "SLA-based service composition in enterprise computing". *Proc. of IEEE IWQoS*, pp 30-39, 2008.

[6] D. Ardagna, M. Trubian, and L. Zhang. "SLA based resource allocation policies in autonomic environments". *Journal of Parallel and Distributed Computing*, vol.67, no.3, pp.259–270, 2007.

[7] A. Jøsang, and J. Haller, "Dirichlet Reputation Systems," *Proc. of the 2nd International Conference on Availability, Reliability and security (ARES 2007)*, pp. 112-119, 2007.

[8] Cloud architecture http://jineshvaria.s3.amazonaws.com/public/cloudarchitectures-varia.pdf

[9] J. Walrand and P. Varaiya. Sojourn times and the overtaking condition in Jacksonian networks. *Adv. Appl. Probab*. 12, 1980.

[10] H. Daduna, "Burke's theorem on passage times in Gordon-Newell networks," *Adv. Appl. Prob.*, 16, 1984.

[11] T. Altiok and B. Melamed. Simulation Modeling and Analysis with Arena. Cyber Research, Inc. and *Enterprise Technology Solutions, Inc., 2001.*

[12] R. Jurca and B. Faltings. "Reputation-based Service Level Agreements for Web Services". In Service Oriented Computing (ICSOC - 2005), vol. 3826 of LNCS, pp. 396- 409. 2005.