

Trust-based Resource Allocation in Web Services

Kaiqi Xiong

Department of Computer Science
North Carolina State University
Raleigh, NC 27965-7534, USA
xiong@csc.ncsu.edu

Harry Perros

Department of Computer Science
North Carolina State University
Raleigh, NC 27965-7534, USA
hp@csc.ncsu.edu

Abstract

With the number of e-Business applications dramatically increasing, service level agreement (SLA) will play an important part in Web services. A SLA is a combination of several quality of services (QoS), such as security, performance, and availability, agreed between a customer and a service provider. Most existing research addresses only one QoS metric, and in the case of the response time, the average time to process and complete a job is typically used.

In this paper, we study trustworthiness, percentile response time and availability. We consider all these qualities for a trust-based resource allocation problem which typically arises in Web services applications. We formulate the trust-based resource allocation problem as an optimization problem under SLA constraints, and we solve it using an efficient numerical procedure.

1 Introduction

Web services technology was introduced as a major component of .NET technology by Microsoft in June of 2000, and it is widely considered as an emerging paradigm for the next generation of Internet computing (Zhang et al. [25]). Web services technology has become the most popular computing paradigm for e-business applications and distributed environments. Many companies such as Google and Amazon are boosting their traffic using Web services APIs (Menasce 2004 [16]). By adopting service-oriented architectures (SOAs), services components from several universe service providers can be flexibly integrated into a composite service regardless of their location, platform, and execution speed (Barry[2], and Singh and Huhns [21]).

The main standard in Web services is Extensible Markup Language, XML. XML provides a foundation for many core standards including Web Services Description Language (WSDL), Universal Description, Discovery and Integration specification (UDDI), and Simple Object Access

Protocol (SOAP) (Curbera et al. [7]). WSDL allows developers to describe what services are offered, and it helps Web services of e-business to be accessed in public. UDDI defines XML-based registries in which businesses can upload information about themselves and the services they offer. SOAP gives us a way to move XML messages between locations, and it enables programs on separate computers to interact across any network. Thus Web services allows us to exchange data with other applications on different computers by using Internet protocols.

However, most existing Web services products do not support service level agreement (SLA) that guarantees a level of service delivered to a customer for a given price. A SLA is a formal contract between a customer and a service provider that defines all aspects of the service being provided. It generally consists of security, performance and availability. *Security* can be categorized as *identity security* and *behavior security*. Identity security includes the authentication and authorization between a customer and a service provider, data confidentiality and data integrity. Behavior security describes the trustworthiness among multiple resource sites, and the trustworthiness of these resource sites by customers, including the trustworthiness of computing results provided by these sites. Performance includes the two following aspects (Menasce [15]).

1. *Response time* is the time for a service request to be satisfied.
2. *Throughput* is the service rate that a service provider can offer. It is defined by the maximum throughput or by the undergoing change of throughput with service intensity.

Finally, *availability* is the percentage of time that a service provider can offer services.

In the paper we consider a resource management problem for Web services under SLA guarantees. Specifically, we define and solve a trust-based resource allocation problem that occurs in typical Web services applications, subject

to the constraints of trustworthiness, percentile response time and availability.

The rest of the paper is organized as follows. Related work is briefly reviewed in section 2. In section 3 we define the trust-based resource allocation problem, and we give its solution in section 4. A numerical example is given in section 5 that demonstrates the validity of this approach. We conclude our results in section 6.

2 Related Work

Web services is often contracted through SLAs which typically specify a certain QoS in return for a certain price. Although QoS was not defined in the initial UDDI standard for Web services, many studies have been carried out to extend the initial UDDI, such as WSLA (Keller and Ludwig [12]), WSOL (Tosic et al. [22]), and QML (Dobson [8]). The issue of a reputation-based SLA has been studied by Jurca and Faltings [11] in which the cost is determined by the QoS that was actually delivered.

SLAs is a combination of various service quality metrics, such as security, availability, and response time. (Menasse [15]). The issue of service quality has been extensively investigated. Zhang et al. [26], and Ziegler and Lausen [27] classified various trust metrics, including a rank-based metric. Vu et al. [23] proposed a new QoS-based semantic Web services selection and ranking solution using a trust and reputation management and assuming known QoS qualities. Wickramage and Weerawarana [24] introduced a performance model to analyze the round trip time of SOAP messages using measurements and a curve fitting technique. Brown and Patterson [5] defined a new availability metric to capture the variations of the system quality of service over time. It is defined by the number of request satisfied per second (or the latency of a request service) and the number of server failures that can be tolerated by a system.

Among the QoS metrics, the response time is the most important one. The computation of the response time has been extensively studied for a variety of computing systems. However, only the average response time is calculated rather than a percentile of the response time. Menasse and Almeida [17] designed self-managing systems to control QoS. Levy et al. [14] presented a performance management system for cluster-based Web services. In both studies the average response time is used as a metric. Chandra [6] employed an online measurement method, and considered a resource allocation problem based on measured response times.

Resource allocation problems in distributed systems have been widely studied based on one of the metrics: trustworthiness and response time (e.g., see [18]). In the paper we consider a resource allocation problem in Web services

subject to all three QoS metrics, expressed by trustworthiness, percentile response time and service availability.

3 The Trust-based Resource Allocation Problem and SLA Metrics

Services components from several universe service resource sites can be flexibly integrated into a composite service with cross-language and cross-platform regardless of their location, platform, execution speed, and process. Delivering quality services to meet customer's requirement under SLA is very important and challenging due to the dynamical and unpredictable nature of these computing systems.

In the paper, a Web services computing system is proposed as shown in Figure 1. The computing system consists of a trust manager that monitors and determines the trustworthiness of the resource sites S_1, S_2, \dots, S_M owned by different service managers A_1, A_2, \dots, A_M , which may not be controlled by the same service provider. As an end user, a customer submits a service request with a certain price for a given quality of services. After the trust manager negotiates a SLA with a service broker who represents resource sites, it checks the trustworthy information of the resource sites and selects those sites which meet pre-define trustworthy requirements for serving the service request. All service managers at the selected sites need to work together to achieve the SLA's requirement, and they overlook all resources within their sites. Upon the completion of a service request, the completed result is sent back to the trust manager. The trust manager forwards the completed job to the customer after checking and updating trustworthiness information in its database. Meanwhile, customers periodically send feedbacks to the trust manager who uses it to update its trustworthiness information as well. The trust-

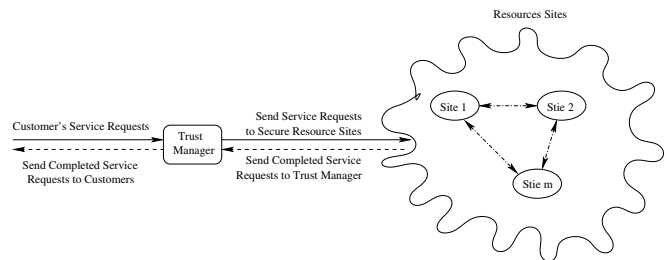


Figure 1. A SLA-based Web Services Model

based resource allocation problem is to minimize the overall cost of the trusted computing resources required while satisfying SLA requirements. For presentation purpose, we assume that each resource site has only one type of server, each with cost c_j . Otherwise, if they have multiple types of servers, we can decompose each resource site into several

sites so that each one only contains one type of server with the same cost. Let N_j be the number of servers at site j ($j = 1, 2, \dots, M$). Thus, the trust-based resource allocation is quantified by solving for n_j ($n_j = 1, 2, \dots, N_j$) in the following optimization problem:

$$\min_{n_1, \dots, n_M} (n_1 c_1 + \dots + n_M c_M) \quad (1)$$

subject to constraints by a SLA. We discuss these constraints in the following subsections.

3.1 The Trustworthiness of Resource Sites

In section 1 we classified security into *identity security* and *behavior security*, similar to identity trust and behavior trust defined in a grid computing system (Azzedin et al. [1]). Identity security includes the authentication and authorization between a customer and a service provider, data confidentiality and data integrity. It has been widely studied in literature (e.g., see Bishop [3], Kim et al. [13], Perrig et al. [19] and J. Rosenberg and D. Remy [20]). The identity security is beyond the scope of our study in this paper.

In this paper, “trust” is used to deal with the notion of the trustworthiness in behavior security. Its definition is varied in literature. In this paper, trust is a firm belief in the competence of a resource site that acts as expected. The trustworthiness of resource sites is an indicator of the quality of services provided by these sites based on previous and current job completion experience. It often predicates the future behavior of the quality of services at these sites. Moreover, we are only interested in the trustworthiness of resource sites from a customer’s perspective. Hence, in this paper we simply assume that these resource sites trust each other. Furthermore, assume that the trust manager is a trusted agent who represents customers. The trust manager uses the collected trustworthy information of the resource sites to evaluate their security behaviors. We consider security behaviors by modeling the behavior trusts of all sites, and quantify the trustworthiness of these sites using a rank-based approach. This approach is based on previous job completion experience assessed by the trust manager and customers.

Let us consider the discrete times $t_1, t_2, \dots, t_k, \dots$ in an increasing order ($k = 1, 2, \dots$). Let $I_j^{t_k}$ be the trust index of site j at time t_k . Then, a *trust function* is defined by

$$I_j^{t_{k+1}} = \xi \frac{R_j^s(k+1)}{R_j^a(k+1)} + (1 - \xi) I_j^{t_k} \quad (2)$$

for each site j ($j = 1, 2, \dots, M$), where $R_j^s(k+1)$ is the number of service jobs completed at site j that satisfied both the trust manager and customers, and $R_j^a(k+1)$ is the total number of service jobs submitted to site j during the time period $[t_k, t_{k+1}]$. The trust manager’s satisfaction is

assessed by the validation of a customer’s SLA requirement after a service is completed, and a customer’s satisfaction is based on the customer’s feedback that may be determined by itself, a third party, or both. Denote $r_j(k+1)$ by

$$r_j(k+1) = \frac{R_j^s(k+1)}{R_j^a(k+1)}$$

Then, $r_j(k+1)$ is the satisfactory rate at site j from time t_k to t_{k+1} . Clearly, when a set of the chosen sites is unchanged during this period of time, the number of completion jobs is the same for all chosen sites. Thereby, $r_j(k+1)$ is the same for these chosen sites as well. ξ is a parameter determined by the trust manager. It is chosen depending on the type of resource sites. If a critical service job is processed at site j and site j ’s security is sensitive with the change of time, then ξ should be close to 1 (e.g., bigger than 0.7). Otherwise, it should be close to 0 (e.g., smaller than 0.3). $I_j^{t_k}$ ranges from 0 to 1, and it is called a *trust index*. It gives the percent of a time that a resource site completes jobs to the satisfaction of the trust manager and its customers.

The trust function describes the trustworthiness of resource sites by the trust manager. Therefore, the trust function reflects a probabilistic security behavior of the resource sites from a customer’s perspective.

3.2 The Percentile Response Time

The SLA performance metric defined in section 1 includes *throughput* and *response time*. As an end user, a customer is in general concerned about response time rather than throughput. So, in the study we only consider *the response time*. In the trust-based resource allocation problem for Web services, a response time is the time it takes for a job to be executed and completed in a distributed computing environment consisting of a trust manager and multiple resource sites.

In the literature, typically the average response time (or an average execution time) is used (e.g., see Menasce [15] and [16]). The average response time is heavily influenced by “outliers,” which occur in almost all measurements. Therefore, although the average response time is relatively easy to calculate, it may not address the concerns of a customer. Typically, a customer is more inclined to request a statistical bound on its response time than an average response time. For instance, a customer can request that 95% of the time its response time should be less than a desired value. Hence, in this paper we are concerned with finding an optimal resource allocation from trusted resource sites that meets a desired percentile response time.

Let $f_T(t)$ be the probability distribution of a response time T , and T^D be a desired response time that a customer requests and agrees with its service provider based on a fee

paid by the customer. Then, the $\gamma\%$ of the response time is guaranteed if

$$\int_0^{T^D} f_T(t) dt \geq \gamma\% \quad (3)$$

As an example let us consider an $M/M/1$ queue with an arrival rate λ and a service rate μ . The service discipline is FIFO. Thus, the steady-state probability of the system is

$$p_0 = 1 - \rho, \text{ and } p_k = (1 - \rho)\rho^k \text{ for } k > 0,$$

where $\rho = \frac{\lambda}{\mu}$.

According to Bolch et al. [4], the response time T is exponentially distributed with the parameter $\mu(1 - \rho)$, i.e., its probability distribution is given by

$$f_T(t) = \mu(1 - \rho)e^{-\mu(1-\rho)t}$$

From (3) we obtain that $\mu \geq \frac{-\ln(1-\gamma\%)}{T^D} + \lambda$. We observe that μ increases when T^D decreases.

3.3 The Service Availability

Availability is a critical metric in today's computer design (Hennessy [10]). Brown and Patterson [5] used an availability metric to describe the variations in system quality of service over time. It is defined by the latency of a request service and the number of failures that can be tolerated by a system. The former was discussed in section 3.2. So, we only need to study the latter in the section. We use consider the percentage of time that a resource is "up" or "down" as a metric, which is the traditional way to define service availability.

Let $MMTF_j$ (Mean Time to Failure) be the average time of a server failure, and MTR_j (Mean Time to Recover) be the average time for recovering a server at the resource site j . Thus each server fails at a rate of $\frac{1}{MMTF_j}$, denoted by a_j , and recovers (i.e., is put back into operation) at a rate of $\frac{1}{MTR_j}$, denoted by b_j ($j = 1, \dots, m$). Thus, a two-state Markov chain with the states "up" and "down" can be used to study the service availability at site j . The failure rate a_j is the state of transition from "up" to "down," and the recovery rate b_j represents the rate of transition from "down" to "up." Then, the probability p_j^i that i servers are down is calculated by

$$p_j^i = \frac{N_j!}{i!(N_j - i)!} \eta_j^i p_j^0, \text{ for } i = 1, \dots, N_j \quad (4)$$

where $\eta_j = \frac{a_j}{a_j + b_j}$ is the server unavailability rate, and p_j^0 is given by

$$p_j^0 = [N_j! \sum_{i=0}^{N_j} \frac{\eta_j^i}{i!(N_j - i)!}]^{-1} \quad (5)$$

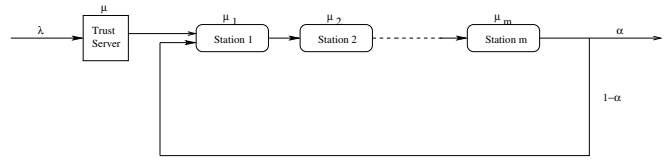


Figure 2. A Web Services Computing System

Moreover, the probability that no more than n_j servers at site j are down is

$$P_j(n_j, N_j) = p_j^0 \sum_{i=0}^{N_j - n_j} \frac{N_j!}{i!(N_j - i)!} \eta_j^i \quad (6)$$

This means that the probability that at least n_j servers in site j are available is P_j given in (6).

4 Trust-based Resource Allocation Algorithms

As we see in section 3, a Web services model consists of a trust manager, and M resource sites. Let m ($0 < m \leq M$) be the number of resource sites necessary for processing a customer's service job. Assume that after the trust manager selects m resource sites from the M ones. Then, these m resource sites can be modelled as a queueing network as shown in Figure 2. Without loss of generality, we assume that the first m sites are chosen. In Figure 2, the tandem queueing network consists of a trust server and m stations numbered sequentially from 1 to m . The trust server represents the trust manager, and each station represents a resource site. Each station carries out a particular function, such as, it could be a database server, a computing server, a file server, and a web server. The feedback loop describes the collaboration of these servers at all sites. The collaborative relationship may be complicated in the real world, but the main idea of our proposed modelling approach can be extended to describe any collaborative relationship among resource sites as long as the relationship can be quantified.

In Figure 2, after a customer exits from the trust server, it continues to be served at the selected sites that are modelled by the m stations. Upon completion of its service at the m -th station, a customer exits the system with probability α , or returns to the beginning of the first station with probability $1 - \alpha$ for another round of service.

Each station j is modelled as a single FIFO queue served by n_j identical servers each providing a service at the rate μ_j . Let λ be the external arrival rate to the trust server that processes service requests sent by customers, and let λ_j be the effective arrival rates to a server at the j -th station for $j = 1, 2, \dots, m$. We assume that all service times are exponentially distributed and the external arrival to the trust

server occurs in a Poisson fashion. The trust server provides a service at the rate μ .

In this paper we are interested in minimizing the overall cost of the above Web services computing system so that the desired SLA is guaranteed. Before presenting an algorithm for solving the minimization problem, we first need to calculate the response time of a customer's service request. Recall that the response time refers to the time elapsed from the moment a customer's service request joins the computing system to the time it departs. It is the most important QoS metric. The response time also reflects security behavior and the availability of services in some degree.

Let T be a random variable that represents the response time of a service request. Also, assume that $f_T(t)$ is the probability distribution of T . Denote the overall service cost (1) by

$$g(n_1, n_2, \dots, n_m) = \sum_{j=1}^m n_j c_j \quad (7)$$

where n_1, n_2, \dots, n_m are the number of servers allocated to a specific stream of jobs with arrival rate λ , not one job.

Then, the trust-based resource allocation problem can be formulated as the following three sub-problems:

1. Select m resource sites within a predefined trust index at time $t = t_k$.
2. Solve for n_j in the m -dimensional integer optimization problem:

$$\min_{n_1, \dots, n_m} g(n_1, n_2, \dots, n_m) \quad (8)$$

Under the constraint of a percentile response time of (3), and the constraint of service availability:

$$P_j(n_j, N_j) \geq \delta_j \% \quad (9)$$

where T^D is a desired response time defined by a customer, and δ_j is a desired percentage of service availability at site j . $P_j(n_j, N_j)$ is given by (6).

3. Update the trust indices of all M sites based on the activity during the time interval $[t_k, t_k + T^D]$. Then, the trust manager decides if a completed job is accepted. If each trust index at those selected sites who completes the service job meet a predefined index value, then the completed job is accepted. Otherwise, then the completed job is discarded and the trust manager needs to resubmit the job.

Note that in the model the verifying time of a trust index is ignored since we are interested in the processing time of a job at resource sites. While sub-problems 1 and 3 are solved at the customer side to ensure a customer service received from reliable resource sites, sub-problem 2 is solved by the

service broker who represents these resource sites. Hence, the trustworthiness of resource sites in sub-problems 1 and 3 cannot be considered as a constraint for the optimization problem in sub-problem 2. The optimization problem is to find a minimal cost presented in (8) such that $\gamma\%$ of the time a customer's response time is less than a predefined value T^D , and $\delta_j\%$ of time the selected resource sites have n_j servers available for the job.

It should be pointed out that the difficulty of the aforementioned optimization problem is to find, $f_T(t)$, the probability distribution of T . We obtain the probability distribution assuming that each station is separate from the other stations, and its waiting time is independent of the waiting times in returning visits to the same station. We first have the following traffic equations:

$$\lambda_1 = \lambda + (1 - \alpha) \lambda_m, \quad \text{and} \quad \lambda_j = \lambda_{j-1}$$

which implies $\lambda_j = \frac{\lambda}{\alpha}$ ($j = 2, \dots, m$).

In the following discussion each server station is modelled as a single $M/M/1$ queue with arrival rate λ_j and service rate $\psi_j \mu_j$, where $\psi_j \in [1, n_j]$ is a function of n_j to be determined by the configuration of servers at each station ($j = 1, 2, \dots, m$). For simplicity, we assume that $\psi_j = n_j$. The following calculation and result can be easily adjusted to the case of $\psi_j \neq n_j$.

Let ρ and ρ_j be the utilizations of the trust server and each station respectively. Then, $\rho = \frac{\lambda}{\mu}$ and

$$\rho_j = \frac{\lambda_j}{\mu_j n_j} = \frac{\lambda}{\alpha \mu_j n_j} \quad (j = 1, 2, \dots, m).$$

Furthermore, the response time of the i -th pass is considered as the sum of $m + 1$ distributed random variables:

$$T(i) = X + X_1 + X_2 + \dots + X_m$$

where X is the service time at the trust server and X_j is the time elapsed from the moment a customer arriving at station j to the moment it departs from it. Thus the total response time is

$$T = \sum_{i=1}^{\infty} p(i) T(i)$$

where $p(i)$ is the steady state probability that a job will circulate i times through the computing system. We have $p(i) = \alpha (1 - \alpha)^{i-1}$ ($i = 1, 2, \dots$).

Under our assumptions, these random variables are independently distributed, and hence the LST (Laplace-Stieltjes transform) of the response time T is:

$$L_T(s) = (L_X(s)) \sum_{i=1}^{\infty} p(i) (L_{X_1}(s))^i \dots (L_{X_m}(s))^i$$

which can be simplified as

$$L_T(s) = \frac{\alpha L_X(s) L_{X_1}(s) \dots L_{X_m}(s)}{1 - (1 - \alpha) L_{X_1}(s) \dots L_{X_m}(s)} \quad (10)$$

where the LST of the service time X is

$$L_X(s) = \frac{\lambda(1-\rho)}{s + \lambda(1-\rho)} \quad (11)$$

and the LST of the response time X_j at the j -th station is

$$L_{X_j}(s) = \frac{n_j \mu_j (1 - \rho_j)}{s + n_j \mu_j (1 - \rho_j)}, \quad (j = 1, 2, \dots, m) \quad (12)$$

Let $\hat{a}_j = n_j \mu_j (1 - \rho_j)$. Then, $L_{X_j}(s)$ in (12) can be rewritten as

$$L_{X_j}(s) = \frac{\hat{a}_j}{s + \hat{a}_j}, \quad (j = 1, 2, \dots, m) \quad (13)$$

Replacing L_X and L_{X_j} by (11) and (13) in (10), we have that

$$L_T(s) = \frac{[\alpha\lambda(1-\rho)]\prod_{j=1}^m \hat{a}_j}{[s + \lambda(1-\rho)][\prod_{j=1}^m (s + \hat{a}_j) - (1-\alpha)\prod_{j=1}^m \hat{a}_j]}$$

which implies the cumulative distribution of the response time given by

$$F_T(t) = \alpha\lambda(1-\rho)\prod_{j=1}^m \hat{a}_j \times L^{-1}\left\{\frac{1}{s[s + \lambda(1-\rho)][\prod_{j=1}^m (s + \hat{a}_j) - (1-\alpha)\prod_{j=1}^m \hat{a}_j]}\right\} \quad (14)$$

To find the cumulative distribution of the response time $F_T(t)$, we need to invert the above LST using partial fraction decomposition of a rational function. But, the partial fraction decomposition of the rational function requires searching for roots of a high-order polynomial. It is usually not an easy task when the order of the polynomial is more than 5. Instead, in this paper the LST will be inverted numerically.

In general, it is not easy to solve the n -dimensional optimization problem presented in Sub-problem 2. But, after some careful analysis of the computing system in Figure 2, its complexity can be significantly reduced as follows.

To achieve the best utilization of these stations, we need to find n_1, \dots, n_m such that $n_1 \mu_1 = \dots = n_m \mu_m$. That is, the maximum service capacity is the same at each station. We have

$$\hat{a}_i = n_i \mu_i (1 - \rho_i) = n_j \mu_j (1 - \rho_j) = \hat{a}_j \triangleq \hat{a}$$

for $i, j = 1, 2, \dots, m$. Thus, equation (14) reduces to

$$F_T(t) = \alpha\lambda(1-\rho)\hat{a}^m \times L^{-1}\left\{\frac{1}{s[s + \lambda(1-\rho)][(s + \hat{a})^m - (1-\alpha)\hat{a}^m]}\right\} \quad (15)$$

and the constraint of service availability at each resource site in (9) is rewritten as

$$G_j(\hat{a}) \stackrel{def}{=} P_j\left(\left\lceil \frac{\hat{a}}{\mu_j(1-\rho_j)} \right\rceil, N_j\right) \quad (16)$$

At this time, $g(n_1, \dots, n_m)$ in (7) reduces to a function of one variable. Thus the trust-based resource allocation problem is solved using the following iterative algorithm.

Algorithm 1

1. At time t_k , select m resource sites within a predefined trust index \hat{I}_j , or the highest m trust indices. If such a selection is impossible, then the trust manager prints ‘‘We cannot process the service request at this moment.’’ If the request is waited for more than a given threshold time, then the trust manager print ‘‘We need to re-negotiate a SLA with the service broker.’’ Otherwise, continue to do Step 2.

2. Find \hat{a} in the following two one-dimensional optimization problems.

- (a) The minimization problem of a percentile response time:

$$\hat{a}^{(1)} \leftarrow \arg \min_{\hat{a}} F_T(t)|_{t=T^D}$$

subject to the constraint $F_T(t)|_{t=T^D} \geq \gamma\%$ at $\hat{a} = \hat{a}^{(1)}$, where $F_T(t)$ is given by (15).

- (b) The minimization problem of service availability:

$$\hat{a}_j^{(2)} \leftarrow \arg \min_{\hat{a}} G_j(\hat{a})$$

subject to the constraint $G_j(\hat{a}^{(2)}) \geq \delta_j\%$, where $G(\hat{a})$ is given by (16).

3. Compute integers n_j by using $n_j = \lceil \frac{\hat{a}_j^M}{\mu_j(1-\rho_j)} \rceil$, where $\hat{a}_j^M = \max(\hat{a}^{(1)}, \hat{a}_j^{(2)})$ for $1 \leq n_j \leq N_j$ and $j = 1, 2, \dots, m$.

4. Update $I_j^{t_k}$ to $I_j^{t_k+T^D}$ based on the trust function (2). If $|I_j^{t_k+T^D}| \geq \hat{I}_j$, then the completed service job is accepted. Otherwise, repeat Steps 1-3. If the repeated number is more than a given threshold, then the trust manager exits and prints ‘‘We need to re-negotiate a SLA with the service broker.’’

Algorithm 1 shows that the m -dimensional optimization problem in Sub-problem 2 significantly reduces to an one-dimensional optimization problem. Surprisingly, the optimal number of servers obtained by Algorithm 1 is independent of server costs c_j , due to $\hat{a}_i = \hat{a}_j$ ($i, j = 1, 2, \dots, m$). In general, each of the above two one-dimensional optimization problems can be solved numerically.

Note that the service broker cannot provide the quality of services at the moment in which a SLA negotiation is requested. In this case, the service broker may get a service penalty. It will be not further discussed here due to the page

limit. Additionally, we have only considered a single-class customer. The above results can be extended to the case of multiple-class customers. The extension will be reported in another paper.

The increased demand for Web services applications has necessitated the creation of large scale computing system with ever-mounting degrees of complexity. In addition, there has been an exponential growth in the number and variety of users and resource sites. Networks are used to interconnect these distributed, heterogeneous large scale systems and our information society continues to create highly complicated workloads for Web services. For this environment, we can introduce a self-managing trust-based resource allocation system for Web services applications that has the core attributes of an autonomic computing environment. We can begin with an adaptive algorithm by using the above proposed approach. This is akin to providing a foundation for the self-managing trust-based resource allocation system. Then, a hierarchical multilevel self-manage trust-based resource allocation system can be used to address the complexity of Web services applications. We omit the details due to lack of space.

5 A Numerical Example

In this section we demonstrate how to apply our algorithm to solve the trust-based resource allocation problem.

Let us consider a five resource site example modelled by the tandem queueing network presented in section 4. We choose $m = 2$, $\xi = 0.6$, and the trust index at time t_1 is given by

$$I^{t_1} = (I_1^{t_1}, I_2^{t_1}, I_3^{t_1}, I_4^{t_1}, I_5^{t_1}) = (0.8, 0.5, 0.2, 0.9, 0.6)$$

Assume that $r_j(k)$ is uniformly distributed in $[0.75, 1]$. r_i and r_j are independent for any $i \neq j$ ($i, j = 1, \dots, 5$). Furthermore, we choose $\lambda = 100$, $T^D = 0.06$, $\gamma = 95$, $\mu = 300$, $\mu_1 = 90$, $\mu_2 = 50$, $\mu_3 = 80$, $\mu_4 = 125$, $\mu_5 = 40$, $\alpha = 0.75$, $\eta_j = 0.008$ and $N_j = 150$ ($j=1, 2, 3$), $\eta_j = 0.009$ and $N_j = 80$ ($j=4, 5$), and $\delta_j = 99.999$ and $\hat{I}_j = 0.9$ ($j = 1, \dots, 5$). A customer submits a service job at time t_5 with $t_{k+1} - t_k = 0.01$ ($k = 1, 2, \dots$) and it requires two of these 5 resource sites satisfying the predefined \hat{I}_j for processing the job.

First, we generated I^{t_k} ($k = 2, \dots, 5, \dots, 11$) in Matlab that

$$I^{t_2} = (0.8661, 0.6814, 0.5870, 0.9275, 0.7921)$$

$$I^{t_3} = (0.8656, 0.8077, 0.8039, 0.8299, 0.8573)$$

$$I^{t_4} = (0.8038, 0.8354, 0.8173, 0.9131, 0.7952)$$

$$I^{t_5} = (0.8867, 0.9298, 0.9254, 0.9336, 0.8339)$$

...

Table 1. The Cumulative Distribution Functions of The Response Time

Response Time	0.02	0.04	0.06	0.08	0.10
Simulation	0.6306	0.8975	0.9704	0.9900	0.9964
Approximation	0.5506	0.8763	0.9673	0.9914	0.9977
Relative Error %	-12.6863	-2.3621	-0.3195	0.1414	0.1305
Response Time	0.12	0.14	0.16	0.18	0.20
Simulation	0.9986	0.9992	0.9998	0.9999	1.0000
Approximation	0.9994	0.9998	0.9999	1.0000	1.0000
Relative Error %	0.0801	0.0600	0.0100	0.0100	0.0000

$$I^{t_{10}} = (0.8884, 0.8697, 0.9174, 0.9242, 0.8642)$$

$$I^{t_{11}} = (0.8798, 0.9329, 0.9402, 0.9164, 0.9184)$$

As we see, sites 2, 3, and 4 meet the trust requirement at $t = t_5$. Thus sites 2 and 4 are selected because they have the highest two trust indices.

Then, we simulated the model in Arena 7.01 and implemented (14) by using the inverse Laplace transform method in Graf [9]. The obtained cumulative distributions are shown in Table 1. It is seen that our approximation result has a good accuracy, where the relative error in Table 1 was calculated by (Approximation Result – Simulation Result)/Simulation Result $\times 100$. Moreover, we obtained that $\hat{a}^{(1)} = 366.67$, i.e., the numbers of servers are 10 and 4 necessary to ensure a 95% response time guarantee for $T^D = 0.06$ at sites 2 and 4 respectively. We validated that they are consistent with the result obtained by a brute force search using the simulation model in Arena. Furthermore, we got $\hat{a}_2^{(2)} = 293.33$ and $\hat{a}_4^{(2)} = 550$, i.e., $n_2 = 8$ and $n_4 = 6$ required for 99.999% service availability using Step 2 (b) of Algorithm 1. By doing the calculation in Step 3 of Algorithm 3, we know that the numbers of servers are 10 and 6 required for the response time and service availability guarantees at sites 2 and 4 respectively.

Finally, the trust manager need to determine whether to accept the job completed by sites 2 and 4 when it receives it at $t = t_5 + T^D = t_5 + 0.06 = t_{11}$. As seen, sites 2 and 4 meet the trust requirement at $t = t_{11}$, and consequently the job is accepted.

6 Conclusions

In this paper, we have studied a trust-based resource allocation problem that arises in typical Web services applications. The problem has been constructed by minimizing the total cost of service providers while satisfying SLA guarantees. We have formulated it as an optimization problem under the constraints of trustworthiness, percentile response time and service availability.

In our approach we considered the percentile response time that may better address a customer's concern as op-

posed to the average response time that is commonly used in the literature. We obtained an efficient and accurate numerical solution for calculating the percentile response time. Then, we proposed an efficient approach for solving the trust-based resource allocation problem. We used a numerical example to illustrate the use of our proposed approach. Preliminary validation tests showed that this approach has a good accuracy. Validation results are not provided due to the page limit.

In the proposed approach we used the rank-based approach to define a trust index, and assumed that a service discipline was FIFO and each resource site could intermediately repair a server. In our future work we will extend our discussion by evaluating different trust approaches and using other service disciplines with a general repairing mechanism for secure resource management in Web service applications and other distributed computing environments.

References

- [1] F. Azzedin and M. Maheswaran. Evolving and managing trust in Grid computing systems. In *Proc. of the IEEE Canadian Conf. on Electrical Computing Engineering (CCECE '02)*. IEEE, May 2002.
- [2] D. K. Barry. *Web Services and Service-Oriented Architecture: Your Road Map to Emerging IT*. Morgan Kaufmann, 2003.
- [3] M. Bishop. *Computer Security*. Addison Wesley, Boston, MA, 2002.
- [4] G. Bolch, S. Greiner, H. Meer, and K. Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Application*. John Wiley & Sons, New York, 1998.
- [5] A. Brown and D. Patterson. Towards availability benchmarks: A case study of software RAID systems. In *Proceedings of 2000 USENIX Annual Technical Conference*. USENIX, June 2000.
- [6] A. Chandra, W. Gong, and P. Shenoy. Dynamic resource allocation for shared data centers using online measurements. In *Proceedings of Eleventh International Conference on Quality of Service (IWQoS 2003)*, June 2003.
- [7] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana. Unraveling the Web services Web: An introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing*, 6(2):86–93, March-April 2002.
- [8] G. Dobson. Quality of service in service-oriented architectures. <http://digs.sourceforge.net/papers/qos.html>, 2002.
- [9] U. Graf. *Applied Laplace Transforms and z-Transforms for Scientists and Engineers*. Birkhauser Verlag, Basel-Boston-Berlin, 2004.
- [10] J. Hennessy. The future of systems research. *IEEE Computer*, 32(8):27–33, August 1999.
- [11] R. Jurca and B. Faltings. Reputation-based service level agreements for Web services. In *Third International Conference on Service Oriented Computing (ICSOC 2005)*. Amsterdam, The Netherlands, December 2005.
- [12] A. Keller and H. Ludwig. The WSLA framework: Specifying and monitoring service level agreements for Web services. *Journal of Network and Systems Management, Special Issue on "E-Business Management"*, 11(1):27–33, March 2003.
- [13] Y. Kim, A. Perrig, and G. Tsudik. Simple and fault-tolerant key agreement for dynamic collaborative groups. In *Proceedings of the 7th ACM Conference on Computer and Communications Security (ACM CCS 2000)*, pages 235 – 244. ACM, 2000.
- [14] R. Levy, J. Nagarajao, G. Pacifici, M. Spreitzer, A. Tantawi, and A. Youssef. Performance management for cluster based Web services. In *The 8th IFIP/IEEE International Symposium on Integrated Network Management (IM2003)*. IEEE, 2003.
- [15] D. Menasce. QoS issues in Web services. *IEEE Internet Computing*, 6(4):72–75, November-December 2002.
- [16] D. Menasce. Response-time analysis of composite Web services. *IEEE Internet Computing*, 8(1):90–92, January-February 2004.
- [17] D. Menasce and M. Bennani. On the use of performance models to design self-managing computer systems. In *Proceedings of 2003 Computer measurement group conference*. IEEE, 2003.
- [18] J. Nabrzyski, J. Schopf, and J. Weglarz. *Grid Resource Management*. Kluwer Academic Publication, Boston, MA, 2004.
- [19] A. Perrig, R. Canetti, D. Song, and D. Tygar. Efficient and secure source authentication for multicast. In *Proceedings of Network and Distributed System Security Symposium*, February 2001.
- [20] J. Rosenberg and D. Remy. *Securing Web services with WS-Security: demystifying WS-Security, WS-Policy, SAML, XML Signature, and XML Encryption*. SAMS, Indianapolis, IN, 2004.
- [21] M. P. Singh and M. N. Huhns. *Service-Oriented Computing*. John Wiley & Sons, Ltd, 2003.
- [22] V. Tomic, K. Patel, and B. Pagurek. WSOL - Web service offerings language. In *Lecture Notes In Computer Science, Vol. 2512, Revised Papers from the International Workshop on Web services, E-Business, and the Semantic Web*, pages 57–67, 2002.
- [23] L. Vu, M. Hauswirth, and K. Aberer. QoS-based service selection and ranking with trust and reputation management. In *Infoscience's Technical Report*.
- [24] N. Wickramage and S. Weerawarana. A benchmark for Web service frameworks. In *Proceedings of the 2005 IEEE international conference on service computing*, pages 233–242. IEEE, July 2005.
- [25] J. Zhang and L. J. Zhang. Editorial preface: A framework to ensure trustworthy Web services. *International Journal of Web Services Research*, 2(3):1–7, July-September 2005.
- [26] Q. Zhang, T. Yu, and K. Irwin. A classification scheme for trust functions in reputation-based trust management. In *International Workshop on Trust, Security, and Reputation on the Semantic Web*. Hiroshima, November 2004.
- [27] C. Ziegler and G. Lausen. Spreading activation models for trust propagation. In *IEEE International Conference on e-Technology, e-commerce, and e-Service (EEE '04)*, April 2002.