

---

## Reliable web service selection via QoS uncertainty computing

---

Shangguang Wang\*

State Key Laboratory of Networking and Switching Technology,  
Beijing University of Posts and Telecommunications,  
Haidian, Beijing, China 100876  
E-mail: sgwang@bupt.edu.cn

\*Corresponding author

Author: Please supply zipcode for each affiliation.

Zibin Zheng

Department of Computer Science and Engineering,  
The Chinese University of Hong Kong,  
Shatin, NT, Hong Kong  
E-mail: zbzheng@cse.cuhk.edu.hk

"Hua Zou" is right

Qibo Sun, Hua Zou and Fangchun Yang

State Key Laboratory of Networking and Switching Technology,  
Beijing University of Posts and Telecommunications,  
Haidian, Beijing, China  
E-mail: qbsun@bupt.edu.cn  
E-mail: huazou@bupt.edu.cn  
E-mail: fcyang@bupt.edu.cn

Author: Please check if the highlighted author name is 'Hua Zou' or 'Hua Zhou'.

**Abstract:** Performance of web services may fluctuate owing to the dynamic internet environment, which makes the Quality-of-Service (QoS) of web services inherently uncertain. With the increase in web services in the internet, selecting the optimal service from a set of functionally equivalent candidates becomes an important research problem. In this paper, we propose an efficient and effective approach for reliable web service selection. Our approach first employs cloud model to compute the QoS uncertainty for pruning redundant services while extracting reliable services. Then, based on QoS uncertainty computing, Mixed Integer Programming (MIP) is used to select optimal services. We evaluate our approach experimentally on real-world web services as well as randomly generated QoS values. The experimental results show that our approach can provide reliable and efficient service selection for users.

**Keywords:** QoS; quality-of-service; web service; service selection; cloud model; mixed integer programming.

Zou, H.

**Reference** to this paper should be made as follows: Wang, S., Zheng, Z., Sun, Q., Zhou, Z. and Yang, F. (xxxx) 'Reliable web service selection via QoS uncertainty computing', *Int. J. Web and Grid Services*, Vol. x, No. x, pp.xxx-xxx.

**Biographical notes:** Shangguang Wang is a Postdoctoral Fellow at Beijing Jiaotong University. He received his PhD in Computer Science and Technology from Beijing University of Posts and Telecommunications in 2011. His research interests include service computing and cloud computing and network security. He served as a reviewer for IET Software; *Journal of Network and Computer Applications*; *International Journal of Web and Grid Services*; *The Computer Journal* and *IEEE ICME 2011*. He also served as session chair for IEEE GLOBECOM 2010 Workshop on Web and Pervasive Security and ICICA 2010.

Zibin Zheng is a Postdoctoral Fellow at The Chinese University of Hong Kong. He received his PhD from The Chinese University of Hong Kong in 2011. He received ACM SIGSOFT Distinguished Paper Award at ICSE'2010, Best Student Paper Award at ICWS'2010, First Runner-up Award at IEEE Hong Kong Postgraduate Research Paper Competition and IBM PhD Fellowship Award 2010–2011. He served as Programme Committee Member of IEEE CLOUD'2009, CLOUDCOMPUTING'2010 and 2011 and SCC'2011. He also served as a reviewer for international journals and conferences, e.g., TSE, TPDS, TSC, IJCCBS, IJBPM, IJWGS, JSS, JSW, ISFI, WWW, DSN, KDD, WSDM, CloudCom, ICEBE, SCC, HASE, ISAS, P2P, QSIC, PRDC, WCNC, SEKE, etc. His research interests include service computing, cloud computing and software reliability engineering.

Qibo Sun received his PhD in Communication and Electronic System from the Beijing University of Posts and Telecommunication in 2002. He is currently an Associate Professor at the Beijing University of Posts and Telecommunication in China. He is a member of the China Computer Federation. His research interests include services computing, internet of things and network security.

Hua Zou received her PhD in Communication and Electronic System from the Beijing University of Posts and Telecommunication in 2010. She is currently a Professor at the Beijing University of Posts and Telecommunication, China. Her current research interests include network intelligence and services computing.

Fangchun Yang received his PhD in Communication and Electronic System from the Beijing University of Posts and Telecommunication in 1990. He is currently a Professor at the Beijing University of Posts and Telecommunication, China. He has published 6 books and more than 80 papers. His research interests include network intelligence, services computing, communications software, soft switching technology and network security. He is a fellow of the IET.

---

## 1 Introduction

Web services have become a promising technology to design and build complex business applications out of atomic web-based software components in Service-Oriented Applications (SOAs) and distributed applications (Eid et al., 2008). According to the SOA paradigm, composite applications are specified as abstract processes composed of a set of abstract services (called *service class*). At service run-time, a concrete web service (called *service candidate*) is selected and invoked for each service class. This procedure ensures loose coupling and flexibility of the design (Canfora et al., 2008). The QoS

parameters (Dong et al., 2009; Marchione et al., 2010) (e.g., response time, price, throughput, etc.) play an important role in determining the success of the composite system. Service Level Agreement (SLA) is often used as a contractual basis between service consumers and service providers on the expected QoS level (Cardellini et al., 2007). QoS-aware service selection aims at efficiently finding the best combination of web service candidates that satisfies a set of end-to-end QoS constraints to fulfil a given SLA.

In the dynamic internet environment, QoS values of web services can change dynamically owing to the update of hardware/software or workload change of servers. Moreover, some of the selected services may become unavailable suddenly at run-time while new service candidates may be launched (Alrifai and Risse, 2009). Therefore, quick responses and auto-adaptation to the dynamic environments are very important for the service-oriented. Performance of the service selection algorithm can have a great influence on the overall performance of the composed system. A number of service selection approaches have been proposed recently (Ardagna and Pernici, 2005; Hwang et al., 2008; Jang et al., 2006; Ma and Zhang, 2008; Yu et al., 2007; Zeng et al., 2004). However, these previous approaches have two major limitations.

First, the previous selection approaches do not consider the uncertainty of QoS seriously. QoS values of the composite applications are usually computed by aggregated QoS values of the remote web services, which may be provided by different organisations, implemented by different programming languages, and run on different platforms (Shangguang et al., 2011). In the unpredictable internet environment, any changes in server location, network condition, time and many other factors may impact the QoS of these internet web services (Xi et al., 2010). In addition, the QoS values may not precisely reflect the actual performance of a web service. For example, service providers of an e-commerce application may not always deliver their 'promised' quality level because of '*intentional deceptions*' (Qi and Bouguettaya, 2010), aiming at attracting a large number of users in a short time and obtain lots of illegal profits. It is worth noting that a web service with consistently good QoS performance is typically more desirable than a web service with a large variance on its QoS performance. Therefore, consistency should be considered as an important criterion when making service selection. Unfortunately, many existing service selection approaches have not considered consistency seriously in their QoS models.

Second, because more and more web services are deployed in the internet, computation time of the web service selection approaches becomes larger. Real-time optimal web service selection becomes more and more difficult. The existing approaches focused too much on the optimisation of selection algorithms themselves to reduce computation time and neglected the basic impact factor, i.e., the exponential increase in web service numbers. As mentioned in the work of Al-Masri and Mahmoud (2008), there has been a more than 130% growth in the number of published web services in the period from October 2006–2007. The statistics published by Seekda!,<sup>1</sup> a real-world web services search engine, also indicated that the number of web services increased exponentially over the last three years. In addition, the pay-per-use business model promoted by cloud computing paradigm will enable service providers to offer their services in different service levels (Candan et al., 2009). Thus, service users will be soon faced with a huge number of variations of the same services offered at different QoS levels. The need for efficient web service selection approaches is becoming more and more urgent.

To address these challenges, we propose a web service selection approach via QoS uncertainty computing. The main contributions of this paper can be summarised as follows:

- We address the problem of web service selection and demonstrate the influence of uncertainty of QoS on the service selection process.
- We propose a novel concept, called QoS uncertainty computing, to model the inherent uncertainty of web service QoS. We adopt cloud model to compute the uncertainty of QoS. According to the three numerical characteristics of cloud model, the web services with a large variance on their QoS can be pruned. To the best of our knowledge, this is the first work that computes the uncertainty of QoS for web service selection.
- On the basis of QoS uncertainty computing, we propose a fast and reliable QoS-aware service selection approach. We evaluate our approach experimentally on 10,258 real-world web services with detailed QoS values as well as on randomly generated QoS values.

The remainder of this paper is organised as follows. Section 2 introduces the background of service composition. Section 3 describes the service selection approach. Section 4 shows the experiments and Section 5 concludes the paper.

## 2 Background

### 2.1 Related concepts

In this section, we introduce some basic concepts of service composition. More detailed background information can be found in Sun et al. (2011) and Wang et al. (2010).

For an abstract composite service, it can be defined as an abstract representation of a composition request  $\mathbb{S} = \{S_1, \dots, S_n\}$ , where  $\mathbb{S}$  refers to the required service classes and  $S_j$  is a service class. A concrete composite service can be defined as an instantiation of an abstract composite service. A concrete composite service can be obtained by binding each abstract service class in  $\mathbb{S}$  to a concrete service  $s_j$ , where  $s_j \in S_j$  and  $\mathbb{S}_j = \{s_{j1}, \dots, s_{jl}\}$  contains  $l(l > 1)$  functionally equivalent services with different QoS values.

For each web service, its QoS attributes can be divided into two categories: positive and negative QoS attributes. Positive attributes mean that the higher the attribute value is, the better the quality is (e.g., reliability and availability). Negative attributes means inversely with the positive attributes. Since positive attribute values can be easily converted into negative attribute values (i.e., positive attribute values are multiplied by  $-1$ ), we only consider negative attributes in this paper for the sake of simplicity.

For a composite service, the QoS attributes must be considered to differentiate large numbers of web services. Hence, to describe QoS-aware web services universally, general QoS aggregation functions are required for service composition. Then, QoS values of each composite service are aggregated by the selected service candidates. In the service composition, the service candidates have different values of the QoS attributes. Utility functions are usually employed to map the vector of QoS values into a single real value, to enable sorting and ranking of service candidates. The QoS utility

function in the paper is similar to Yu et al. (2007). For example, the minimum and maximum aggregated values of the  $k$ th QoS attribute of  $\mathbb{S}$  are computed as follows:

$$U(s) = \sum_{k=1}^r \frac{Q_{j,k}^{\max} - q_k(s)}{Q_{j,k}^{\max} - Q_{j,k}^{\min}} \cdot w_k \quad (1)$$

$$U(\mathbb{S}) = \sum_{k=1}^r \frac{Q_k^{\max} - q_k(\mathbb{S})}{Q_k^{\max} - Q_k^{\min}} \cdot w_k \quad (2)$$

with

$$Q_k^{\max} = \sum_{j=1}^n Q_{j,k}^{\max}, \quad Q_{j,k}^{\max} = \max_{\forall s_{ji} \in S_j} q_k(s_{ji})$$

$$Q_k^{\min} = \sum_{j=1}^n Q_{j,k}^{\min}, \quad Q_{j,k}^{\min} = \min_{\forall s_{ji} \in S_j} q_k(s_{ji})$$

where  $w_k \in R^+$  ( $\sum_{k=1}^r w_k = 1$ ) represents users' preferences,  $Q_{j,k}^{\min}$  is the minimum value of the  $k$ th attribute in all service candidates of the service class  $S_j$  and similarly  $Q_{j,k}^{\max}$  is the maximum value,  $Q_k^{\min}$  is the minimum value of the  $k$ th attribute of  $\mathbb{S}$  and similarly  $Q_k^{\max}$  is the maximum value.

The service selection with global QoS constraints is an optimisation process. The optimal selection for a given service composition  $\mathbb{S}$  must meet the following two conditions:

- For a given vector of global QoS constraints  $\mathbb{C} = \{C_1, \dots, C_m\}$  ( $0 \leq m \leq r$ ),  $q(\mathbb{S}) \leq C$  ( $\forall C_k \in \mathbb{C}$ ), where  $q(\mathbb{S})$  is the aggregated QoS value of the composition service
- The maximum overall utility value  $U(\mathbb{S})$  in the composition service.

However, finding the optimal composition requires enumerating all possible combinations of service candidates, which can be very expensive in terms of computation time. Furthermore, it is difficult to assure the reliability of the selected services.

## 2.2 Related work

To obtain reliable composite services with low cost, some notable approaches or schemes have been proposed.

Zeng et al. (2004) described and compared two web service selection approaches, i.e., local optimisation and global optimisation. They pointed out that the global optimisation was better than local optimisation, which can obtain the optimal composite service with global constraints guarantee. At present, most existing solutions of QoS-aware web service selection in SOAs are based on the global optimisation. In Yu et al. (2007), heuristic algorithms were used to find a near-to-optimal solution. The algorithms are more efficient than exact solutions and are suitable for making run-time decisions. However, these algorithms do not consider the uncertainty of QoS. The selected services may deviate from the actual execution results in the dynamic service environment.

Alrifai and Risse (2009) proposed an effective and efficient web service selection approach that combines global optimisation with local selection techniques to find the

most suitable services according to users' global QoS constraints. The basic idea of the proposed solution is to adopt MIP to find the optimal decomposition of global QoS constraints into local constraints, and then to find the best web services that satisfy these local constraints by using distributed local selection.

San-Yih et al. (2007) defined an Aggregated Reliability (AR) to measure the probability that a given state in the composite web service will lead to successful execution in the context where each component web service may fail with some probability. Moreover, to orchestrate a composite web service, they proposed two strategies to select component web services that are more likely to successfully complete the execution of a given sequence of operations.

Ardagna and Pernici (2007) proposed a novel service selection optimisation approach, which contains three main ideas:

- loops peeling is adopted in the optimisation
- if a feasible solution for the service composition problem does not exist, negotiating QoS parameters is performed
- a new class of global constraints is introduced.

The approach performs well in QoS-aware web service selection, which made service users obtain composite services with their QoS requirements quickly.

Hwang et al. (2008) formulated the dynamic web service selection problem in a dynamic and failure-prone environment. They proposed using a finite state machine to model the invocation order of operations in each web service and to construct a web service composition that enumerates all possible delegations. Then, a measure, called AR, was defined to determine the probability that the execution of operations starting from a configuration will successfully terminate. Furthermore, two web service selection strategies, AR-based and Composability and AR-based, were proposed. The approach improves the success rate and reliability of service composition effectively.

Zheng and Lyu (2009) proposed a QoS-aware fault-tolerant middleware to make fault tolerance for the distributed SOA systems efficient, effective and optimised. The core of this study mainly contains a user-collaborated QoS model and a context-aware algorithm. The user-collaborated QoS model is used to comply with the key concept of web 2.0. The context-aware algorithm is designed for determining optimal fault tolerance strategy dynamically and automatically for both stateless and stateful web services. Soon afterwards, to evaluate distributed QoS of real-world web services, Zheng et al. (2010) collected 21,358 web service addresses by crawling web service information from the internet and conducted two large-scale distributed evaluations. For the first evaluation, 1,542,884 web service invocations were executed by 100 distributed service users on 150 web services. For the second evaluation, 1,974,675 real-world web service invocations were executed by 339 distributed service users on 5825 web services. The scales of the distributed web service evaluations are the largest in the field of service computing. Their released data sets (WS-DREAM) can be employed by a lot of QoS-aware research topics on web services. Moreover, the other studies of Zheng and Lyu (2010), Zibin et al. (2011) and Zibin and Lyu (2008) also made good achievements on reliable web service or service composition, and have been shown to perform well in QoS-aware web service selection.

San-Yih et al. (2010) proposed a choreography model where each composite web service participating is associated with an abstract process and is capable of dynamically

Author: Please supply expansion for the acronym 'WS'.

performing WS selection for each step defined based on the information about itself and provided by its collaborators. Then, a method was developed for each composite web service to dynamically select and invoke other web services based on the limited information about other web services so as to maximise the likelihood of completing the entire choreography in a failure-prone environment.

Jing et al. (2010) proposed a distributed heuristic web service selection approach for composite system. The basic idea of the approach is to decompose the global optimisation problem into local ones, and then perform service selection on each QoS registry independently, finally collect the local results and perform centralised optimisation. In this efficient approach, variable elimination is used to reduce the size of service selection problem; constraint decomposition allows performing service selection independently on each QoS registry; supplementary service selection and concentrated optimisation improve the approximation ratio.

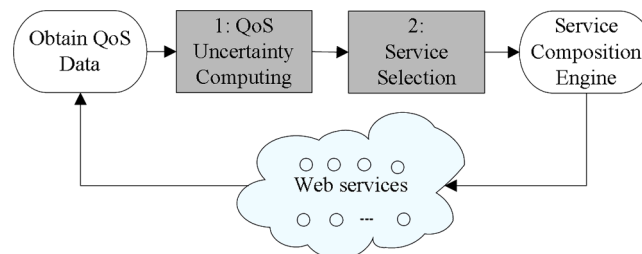
Liu et al. (2011) proposed a global selection-optimal web service selection approach by considering both transactional constraints and end-to-end QoS constraints. The basic idea of the approach is to identify building rules and build layer-based Directed Acyclic Graph model, which can model transactional relationships among candidate services, and then present Graph-building algorithms and an optimal selection algorithm to explain the specific execution procedures. By using the efficient approach, the problem of solving global optimal QoS utility with transactional constraints can be regarded as a problem of solving single-source shortest path in Directed Acyclic Graph.

Other related studies include QoS estimation (Wang et al., 2010), policy-driven QoS monitoring (Fei et al., 2008), quality of online service estimation (Le-Hung and Aberer, 2009), etc.

### 3 Service Selection based on QoS Uncertainty Computing (SSQC)

As shown in Figure 1, the proposed Service Selection based on QoS Uncertainty Computing (SSQC) approach contains two phases. Phase 1 is QoS uncertainty computing (will be introduced in Section 3.1), in which we adopt cloud model to transform the quantitative QoS to qualitative QoS for the QoS uncertainty computation. Phase 2 is service selection (will be introduced in Section 3.2), in which we adopt an MIP to find the most suitable service from each service class. Eventually, service composition invokes the selected services and returns the invocation results to service users.

**Figure 1** Procedures of SSQC approach (see online version for colours)



### 3.1 QoS uncertainty computing

To reduce the fluency of QoS uncertainty on the reliability of service selection, SSQUC adopts cloud model to compute the uncertainty by transforming quantitative QoS values (transaction logs) to qualitative QoS concept (uncertainty level). According to the uncertainty level, a web service with consistently good QoS can be distinguished from those services with a large QoS variance.

#### 3.1.1 Cloud model

Cloud model (Li et al., 1998) is a model of uncertainty transition between a linguistic term of a qualitative concept and its numerical representation. It can be employed for the uncertainty transition between qualitative concept and quantitative description. A cloud model can be defined as:

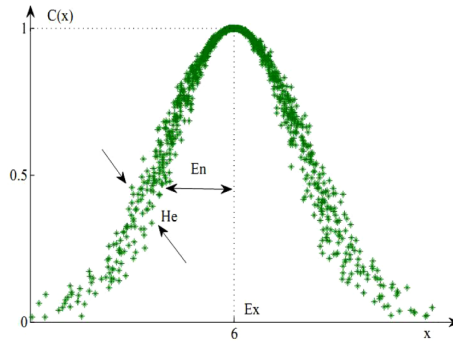
**Definition 1:** Let  $U$  be the set as the universe of discourse and  $C$  be a qualitative concept associated with  $U$ . The membership degree of quantitative numerical representation  $x$  in  $U$  to the concept  $C$ ,  $\mu(x) \in [0, 1]$ , is a random number with a stable tendency, which is as in equation (3):

$$\mu: U \rightarrow [0,1], \quad \forall x \in U, \quad x \rightarrow \mu(x). \quad (3)$$

The distribution of  $x$  in the universe of discourse  $U$  is called cloud  $C(X)$ , and  $x$  is called a cloud drop.

The literature (Li et al., 1998) gave many kinds of cloud models, such as normal cloud,  $\gamma$  cloud. Because a lot of uncertainty concepts behave normal clouds in social and natural phenomena, in our study, we mainly apply normal cloud model. The overall characteristics of cloud model may be reflected by its three numerical characteristics: Expected value ( $Ex$ ), Entropy ( $En$ ) and Hyper-entropy ( $He$ ). Figure 2 shows the three numerical characteristics of cloud model, where the number of cloud drops is 1000. In the discourse universe,  $Ex$  is the position corresponding to the centre of the cloud gravity, whose elements are fully compatible with the linguistic concept;  $En$  is a measure of the concept coverage, i.e., a measure of the fuzziness, which indicates how many elements could be accepted to the qualitative linguistic concept, and  $He$  is a measure of the dispersion on these cloud drops, which can also be considered as the entropy of  $En$ . Then, the vector  $NC = \{Ex, En, He\}$  is called the eigenvector of cloud model.

**Figure 2** Three numerical characteristics of cloud model (see online version for colours)





By using the three numerical characteristics *Ex*, *En* and *He*, the cloud generator can produce many cloud drops. The transforms between a qualitative concept expressed by the three numerical characteristics of a cloud and its quantitative representation expressed by a set of numerical cloud drops are performed by cloud generators: forward cloud generator and backward cloud generator. In this study, we apply these three numerical characteristics of backward cloud generator (Algorithm 1) to denote the uncertainty of QoS by transforming QoS quantitative values to qualitative concept. More information about the cloud model is in Li et al. (1998).

### 3.1.2 Application of the cloud model

We first take two services WSA and WSB that offer the similar hotel web service as an example to illustrate the different implications for reliable web services between certain and uncertain QoS. In this example, the performance of WSA and WSB is recorded by a series of transaction logs, which helps capture the actual QoS delivered by each provider in practical application. Because in dynamic environment, these service providers operate, which causes the uncertainty of their performance, this can be reflected by the fluctuation among different transactions. Although the actual number of transactions should be much larger, for the ease of illustration, we only consider five transactions with services WSA and WSB, respectively. These transactions are represented as ( $wsa_1, \dots, wsa_5$ ) and ( $wsb_1, \dots, wsb_5$ ) as shown in Table 1. Table 1 gives response-time values of these transactions. The aggregated QoS values ( $\underline{wsa}$  and  $\underline{wsb}$ ), which are obtained by averaging all transactions, are given in the last row of Table 1.

**Table 1** A set of service transactions

<i>Web service: WSA</i>		<i>Web service: WSB</i>	
<i>ID</i>	<i>Response time (ms)</i>	<i>ID</i>	<i>Response time (ms)</i>
$wsa_1$	29	$wsb_1$	16
$wsa_2$	26	$wsb_2$	40
$wsa_3$	30	$wsb_3$	31
$wsa_4$	26	$wsb_4$	34
$wsa_5$	26	$wsb_5$	12
$\underline{wsa}$	27.4	$\underline{wsb}$	26.6

From Table 1, the aggregate QoS values of WSA are larger than that of WSB, i.e.,  $\underline{wsa} > \underline{wsb}$ . In traditional service selection approach, web services WSB is usually selected as a service component in service composition because of  $26.6 < 27.4$ . However, after analysing each transaction of these two services in great depth, we find the following two important facts that may be ignored by traditional approaches:

- Although the average response time of service WSB is slightly less than that of WSA, the three transactions ( $wsa_2, wsa_3$  and  $wsa_4$ ) of service WSA is less than ( $wsb_2, wsb_3$  and  $wsb_4$ ) of service, i.e.,  $wsa_2 < wsb_2, wsa_3 < wsb_3$  and  $wsa_4 < wsb_4$ . This means that the response time of service WSA is less than that of service WSB in most transactions.

- The response time of service WSB is more volatile than that of service WSA, i.e., service WSB is with a large variance on its QoS, while service WSA is with consistently good QoS.

According to the two facts stated earlier, if service WSB is selected as a service component, the actual execution result of service WSB may deviate from  $w_{sb}$ , leading to poor composition service quality or service selection failure. Thus, it may be obvious that service WSA is more stable than service WSB, and WSA as a service component may be more suitable than service WSB. So how to compute the uncertainty of web service, i.e., how to distinguish one web service with a consistently good QoS from another web service with a large variance on its QoS, is an important issue. In this paper, we adopt the backward cloud generator algorithm of cloud model to distinguish these services.

---

**Algorithm:** Backward cloud generator

**Input:**  $n$  transactions of a Web service, i.e.,  $n$  cloud drops  $\{x_1, x_2, \dots, x_n\}$ ;

**Output:** the three numerical characteristics  $Ex$ ,  $En$ , and  $He$  of the  $n$  cloud drops.

---

1. According to  $x_i$ , computing the sample mean  $\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i$ , and the sample variance  $S^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{X})^2$ ;
  2. The Expected value of the Web service on its QoS can be calculated by  $Ex = \bar{X}$ ;
  3. The Entropy of the Web service on its QoS also can be calculated by  $En = \frac{\sqrt{\pi/2}}{N} \sum_{i=1}^N |x_i - Ex|$ ;
  4. Finally, the Hyper-Entropy can be obtained by  $He = \sqrt{S^2 - En^2}$ .
- 

We apply the above-mentioned algorithm to Table 2. These response-time values can be seen as quantitative QoS values expressed by five cloud drops, i.e.,  $(wsa_1, \dots, wsa_5)$  or  $(wsb_1, \dots, wsb_5)$ . The qualitative QoS concept (uncertainty level) of each service can be expressed by its eigenvector. Then, these eigenvectors of services WSA and WSB can be calculated as follows:  $NC_S = \{27.40, 2.11, 3.10\}$ , and  $NC_T = \{26.60, 12.63, 144.25\}$ . Since  $2.11 < 12.63$  and  $3.10 < 144.25$ , the uncertainty level of service WSA is smaller than that of service WSB. This means that the QoS of service WSA is consistently good but service WSB is with a large variance on its QoS. Thus, the service WSA should be selected as a service component rather than service WSB, which is different from the traditional approaches.

To apply the cloud mode to web services, we set the parameters  $\lambda$  and  $h$  as the thresholds of  $En$  and  $He$  according to different web service environments. The web services with a large QoS variance and the web services with a consistently good QoS performance can be distinguished using the conditions  $En \leq \lambda$  and  $He \leq \lambda$ , respectively. The latter condition will be seen as service candidates prior to the former for reliable service selection. This will guarantee that selected services can be reliably executed. Furthermore, redundant service candidates will be pruned for each service class because of  $En > \lambda$  or  $He > \lambda$ . By this way, cloud model can help reduce the search space of service selection and shorten the computation time in service composition. Because the QoS uncertainty computing is independent of any individual service request, it does not need to be conducted online at request time. Hence, we make use of the cloud model for obtaining reliable service candidates offline to speed up the service selection process.

### 3.2 Service selection

After the QoS uncertainty computing, service candidates with consistently good QoS performance can be discovered in each service class. Then, a service selection algorithm needs to be designed to find the most suitable service of each class under global QoS constraints. By only focusing on the services that have consistently good performance, we speed up the selection process and are able to select reliable services.

MIP is used to solve the optimisation problem of service selection based on the obtained services. The MIP has recently been used to solve the service composition problem by several researchers (Alrifai and Risse, 2009; Yu et al., 2007). In our study, binary decision variables are used in the problem to represent the service candidates. A service candidate  $s_{ji}$  is selected in the optimal composition if its corresponding variable  $x_{ji}$  is set to 1 in the solution of the model and discarded otherwise. By re-writing equation (2) to include the decision variables, the problem of solving the model can be formulated as a maximisation problem of the overall utility value given by

$$\sum_{k=1}^r \frac{Q_k^{\max} - \sum_{j=1}^n \sum_{i=1}^l x_{ji} \cdot q_k(s_{ji})}{Q_k^{\max} - Q_k^{\min}} \cdot W_k \quad (4)$$

subject to the global QoS constraints and satisfying the allocation constraints on the decision as

$$\begin{cases} \sum_{j=1}^n \sum_{i=1}^l q_k(s_{ji}) \cdot x_{ji} \geq C_k, & 1 \leq k \leq m \\ \sum_{j=1}^n x_{ji} = 1, & 1 \leq j \leq n \end{cases} \quad (5)$$

By solving equations (4) and (5) using any MIP solver methods, a list of service candidates are obtained and returned to service composition engine or service broker providing new value-added services for service users.

## 4 Experiments

In this section, we compare with the approach proposed in Ardagna and Pernici (2007) by conducting two experiments in Sections 4.2 and 4.3, respectively. The first experiment indicates that our approach has much higher success ratio than other approaches, and the second experiment shows that the computation time consumed by our approach is much shorter than other approaches.

### 4.1 Experiment set-up

In this study, we conduct our experiments using two types of data sets. The first is a real-world web service QoS data set named WS-DREAM<sup>2</sup> from Zheng et al. (2010). WS-DREAM data set contains about 1.5 millions web service invocation records of 150 service users in 24 countries. Values of three QoS attributes (i.e., *Response Time*, *Response Data Size* and *Failure Probability*) are collected by these 150 service users on

10,258 web services. The second data set is a randomly generated data set (named RG) that also contains values of three QoS attributes of 10,000 web services.

In the experiments, we randomly partitioned each data set into 10 service classes. The threshold values of Entropy and Hyper-Entropy are set to ( $\lambda = 3.8$ ,  $h = 5.9$ ). The number of QoS attributes is set to 3, and the number of QoS constraints is set to 2. The number of service candidates per service class varies from 10 to 100.

All the experiments are conducted on the same computer with Pentium 2.0GHz processor, 2.0GB of RAM, Windows XP SP3, Ip-solve 5.5 and Matlab 7.6. All approaches are run for 20 times and all results are reported on average.

#### 4.2 Success ratio

An important goal of service selection approach is to select reliable services for service users. However, owing to the uncertainty of QoS, the selected service often deviates from the user expectations, which may lead to service composition failure in practical application. Thus, the aim of this section is to compare the success ratio of our proposed SSQUC approach with other well-known approach in service selection process, i.e., the MAIS approach proposed in Ardagna and Pernici (2007).

**Definition 2:** Success Ratio (SR) is how often the ratio of users' QoS constraints ( $C_i$ ) to monitored aggregated QoS values ( $\bar{U}_i(\mathbb{S})$ ) is greater than or equal to a threshold value ( $th$ ) for  $n$  composition services, i.e.,

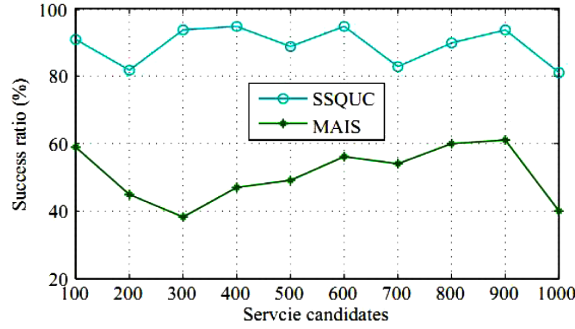
$$SR = \frac{srn}{n} \times 100\% \quad \text{and} \quad srn = \sum_{i=1}^{100} 1, \quad \bigcap_{i=1}^k \left\{ \frac{C_i}{\bar{U}_i(\mathbb{S})} \geq th \right\}$$

$$SR = \frac{srn}{n} \times 100\% \quad \text{and} \quad srn = \sum_{i=1}^{100} \begin{cases} 1, & \bigcap_{i=1}^k \frac{C_i}{\bar{U}_i(\mathbb{S})} \geq th \\ 0, & \text{otherwise} \end{cases}$$

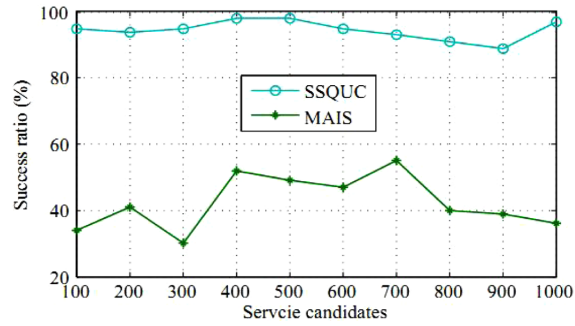
According to Definition 2, for a service selection approach, the higher its success ratio is, the better its performance. Figures 3 and 4 show the comparison with MAIS on success ratio employing two different QoS data sets. In the experiment, the parameters are set as  $th = 0.86$  and  $n = 200$ .

From Figures 3 and 4, with different number of service candidates, the success ratio of SSQUC is much higher than that of MAIS on both data sets. The overall success ratio of SSQUC is 91.95% on average, while that of MAIS is only 47.05%. These experimental results indicate that our approach effectively reduces the influence of QoS uncertainty on the quality of composition service and greatly improves the reliability of service selection, because cloud model is adopted to compute the uncertainty of web service QoS in our SSQUC approach. By using En and He to monitor QoS transactions, SSQUC effectively defences from web services with a large variances on its QoS, and reduces the difference between selected services and actual execution results. Hence, the reliability of service selection can be greatly improved.

**Figure 3** Success ratio comparison employing WS-DREAM data set (see online version for colours)



**Figure 4** Success ratio comparison employing RG data set (see online version for colours)



### 4.3 Computation time

The existing service selection approaches usually ignore reducing the redundant candidates, which will lead to large computation time. In this section, we conduct experiments to compare SSQUC approach with MIAS approach on the computation time. Table 2 gives the experimental results of the two approaches.

**Table 2** Experimental results on computation time

Number of service candidates	RG data set		WS-DREAM data set	
	MIAS (s)	SSQUC (s)	MIAS (s)	SSQUC (s)
100	0.443	0.300	0.344	0.316
200	0.500	0.441	0.828	0.341
300	0.765	0.672	1.110	0.643
400	1.140	1.020	1.578	1.101
500	1.641	1.187	2.140	1.578
600	2.328	1.453	2.828	1.812
700	3.844	1.906	3.640	2.312
800	3.562	2.203	4.422	2.625
900	4.469	2.735	5.953	3.109
1000	5.359	3.125	7.469	3.531

From Table 2, we can see that computation time (second unit) of SSQUC is very short. All results are shorter than 4 sec. Compared with the MIAS approach, the computation time of SSQUC is shorter than that of MIAS on both data sets. The experimental results indicate that SSQUC significantly reduces the time cost of service selection, since the searching space is reduced in our approach.

Thus, according to the experimental results on success ratio and computation time, SSQUC can efficiently and reliably perform service selection for service-oriented systems and composed cloud applications.

#### 4.4 Parameters study

In this section, we conduct two experiments to analyse the impact of parameters setting on success ratio of service selection by using the WS-DREAM data set.

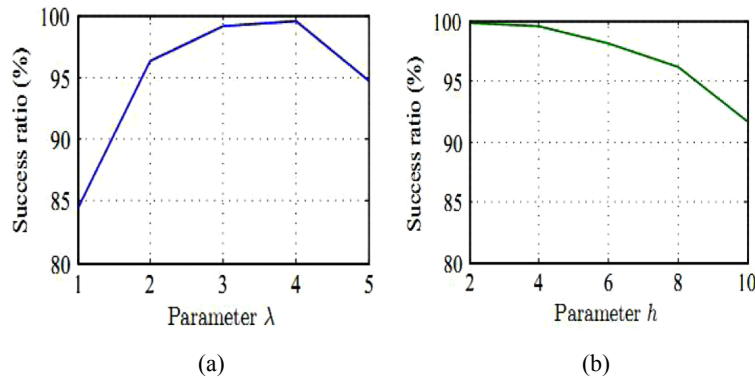
Parameter  $\lambda$  indicates the threshold value of the entropy of QoS for monitored web service. In this experiment, we vary  $\lambda$  from 1 to 5 with a step value of 1, the value of  $h$  is set to 3, the number of service candidates is set to 50, and the number of service classes is set to 10. Figure 5(a) shows the experimental results.

Figure 5(a) shows that the success ratio will not increase with the increase in  $\lambda$  value. This observation indicates that by using QoS uncertainty computing, we can simply set the value of  $\lambda$  to an appropriate value for obtaining higher success ratio.

Parameter  $h$  indicates the threshold value of the hyper-entropy of QoS for monitored WS. In this experiment, we vary  $h$  from 2 to 10 with a step value of 2. In addition, we set  $\lambda = 3$ , the number of service candidates = 500, the number of service classes = 10 in the experiments. Figure 5(b) shows the experimental results.

Figure 5(b) shows that the success ratio is stable when  $h$  is smaller than a certain value. In addition, the success ratio will decrease when  $h$  is larger than a certain value. This observation indicates that we should set a small  $h$  value for obtaining higher success ratio.

**Figure 5** Impact of parameters on success ratio: (a) success ratio with respect to  $\lambda$  and (b) success ratio with respect to  $h$  (see online version for colours)



#### 4.5 Discussions

Web service composition is a failure-prone environment partly owing to the autonomous requirement of each participating service provider. A web service may become malfunctioned or unavailable at run-time, causing failure to the execution of a composite web service (Hwang et al., 2008). Hence, it is worth noting that a web service with consistently good QoS is typically more desirable than a web service with a large variance on its QoS. In the SSQUC approach, we use cloud model to prune the services with a large variance on their QoS. Then since the QoS of service candidates in each service class is consistently good, the services selected are also reliable. The experimental results from Figures 3 and 4 validate the reliability of service selection.

In addition, a few hundreds of services can provide the same functionality, which made the computation time of service selection very long. In our SSQUC approach, some web services with a large variance on their QoS are pruned, which made the size of service selection become small. Hence, the time cost of SSQUC obviously decreases. The experimental results from Table 2 validate the efficiency of our QoS uncertainty computing.

### 5 Conclusions

To guarantee the reliability and real-time requirement of web service selection, this paper proposes an efficient approach for reliable QoS-aware SSQUC. Our approach applies cloud model to compute the uncertainty of QoS and uses MIP to identify the most suitable web services. We evaluate our approach experimentally using both real-world and randomly generated web service QoS data sets. The experimental results show that our approach can perform reliable service selection for users.

In our future work, we will continue to investigate more efficient web service selection approaches. For instance, a more comprehensive parameter study with respect to the number of service classes and the number of services candidates will be considered. Besides, based on the cloud model, a new web service selection approach combining QoS prediction will be explored.

### Acknowledgements

The work presented in this study is supported by the National Basic Research and Development Program (973 program) of China (No. 2009CB320406); the National High Technology Research and Development Program (863 program) of China (No. 2008AA01A317) and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 60821001) and ‘HeGaoJi’-Important National Science & Technology Specific Projects (2009ZX01039-001-002-01).

### References

- Al-Masri, E. and Mahmoud, Q.H. (2008) ‘Investigating web services on the world wide web’, *Proceedings of 17th International Conference on World Wide Web 2008, WWW’08*, 21–25 April 2008, Beijing, China, pp.795–804.

- Alrifai, M. and Risse, T. (2009) 'Combining global optimization with local selection for efficient QoS-aware service composition', *Proceedings of Proceedings of the 18th International Conference on World Wide Web(WWW2009)*, 18 August 2009, New York, United States, pp.881–890.
- Ardagna, D. and Pernici, B. (2005) 'Global and local QoS constraints guarantee in web service selection', *Proceedings of*, 2005 Orlando, FL, USA, pp.805–806. **AUTHOR PLEASE SUPPLY FULL DETAILS.**
- Ardagna, D. and Pernici, B. (2007) 'Adaptive service composition in flexible processes', *IEEE Transactions on Software Engineering*, Vol. 33, No. 6, pp.369–384.
- Candan, K.S., Li, W-S., Phan, T. and Zhou, M. (2009) 'Frontiers in information and software as services', *Proceedings of 25th IEEE International Conference on Data Engineering*, 2009, 29 March–2 April 2009, Shanghai, China, pp.1761–1768.
- Canfora, G., Di Penta, M., Esposito, R. and Villani, M.L. (2008) 'A framework for QoS-binding and re-binding of composite web services', *Journal of Systems and Software*, Vol. No. 10, pp.1754–1769.
- Cardellini, V., Casalicchio, E., Grassi, V. and Lo Presti, F. (2007) 'Flow-based service selection for web service composition supporting multiple QoS classes', *Proceedings of Web Services, 2007. ICWS 2007. IEEE International Conference on*, 9–13 July 2007, pp.743–750. **AUTHOR PLEASE SUPPLY LOCATION.**
- Dong, H., Hussain, F.K. and Chang, E. (2009) 'A QoS-based service retrieval methodology for digital ecosystems', *International Journal of Web and Grid Services*, Vol. 5, No. 3, pp.261–283.
- Eid, M., Alamri, A. and El Saddik, A. (2008) 'A reference model for dynamic web service composition systems', *International Journal of Web and Grid Services*, Vol. 4, No. 2, pp.149–168.
- Fei, L., Fangchun, Y., Kai, S. and Sen, S. (2008) 'A policy-driven distributed framework for monitoring quality of web services', *Proceedings of IEEE International Conference on Web Services, ICWS 2008*, 23–26 September, 2008, Beijing, China, pp.708–715.
- Hwang, S-Y., Lim, E-P., Lee, C-H. and Chen, C-H. (2008) 'Dynamic web service selection for reliable web service composition', *IEEE Transactions on Services Computing*, Vol. 1, No. 2, pp.104–116.
- Jang, J.H., Shin, D.H. and Lee, K.H. (2006) 'Fast quality driven selection of composite web services', *Proceedings of Proceedings of ECOWS 2006: 4th European Conference on Web Services*, 2006 Zurich, Switzerland, pp.87–96.
- Jing, L., Yongwang, Z., Min, L., Hailong, S. and Dianfu, M. (2010) 'An adaptive heuristic approach for distributed QoS-based service composition', *Proceedings of Computers and Communications (ISCC), 2010 IEEE Symposium on*, 22–25 June, 2010, Riccione, Italy, pp.687–694.
- Le-Hung, V. and Aberer, K. (2009) 'Towards probabilistic estimation of quality of online services', *Proceedings of Web Services, 2009. ICWS 2009. IEEE International Conference on*, 2009 Los Angeles, USA, pp.99–106.
- Li, D., Cheung, D., Shi, X. and Ng, V. (1998) 'Uncertainty reasoning based on cloud models in controllers', *Computers and Mathematics with Applications*, Vol. 35, No. 3, pp.99–123.
- Liu, H., Zheng, Z., Zhang, W. and Ren, K. (2011) 'A global graph-based approach for transaction and QoS-aware service composition', *KSII Transactions on Internet and Information Systems*, Vol. 5, No. 7, pp.1252–1273.
- Ma, Y. and Zhang, C. (2008) 'Quick convergence of genetic algorithm for QoS-driven web service selection', *Computer Networks*, Vol. 52, No. 5, pp.1093–1104.
- Marchione, F.G., Fantinato, M., de Toledo, M.B.F. and Gimenes, I.M.D. (2010) 'E-contracting with price configuration for web services and QoS', *International Journal of Web and Grid Services*, Vol. 6, No. 4, pp.357–384.



- Qi, Y. and Bouguettaya, A. (2010) 'Computing service skyline from uncertain QoS', *Services Computing, IEEE Transactions on*, Vol. 3, No. 1, pp.16–29.
- San-Yih, H., Ee-Peng, L., Chien-Hsiang, L. and Cheng-Hung, C. (2007) 'On composing a reliable composite web service: a study of dynamic web service selection', *Proceedings of Web Services, 2007. ICWS 2007. IEEE International Conference on*, 9–13 July 2007, Salt Lake City, USA, pp.184–191.
- San-Yih, H., Wen-Po, L. and Chien-Hsiang, L. (2010) 'Web services selection in support of reliable web service choreography', *Proceedings of Web Services (ICWS), 2010 IEEE International Conference on*, 5–10 July 2010, pp.115–122.
- Shangguang, W., Zheng, Z., Qibo, S., Hua, Z. and Fangchun, Y. (2011) 'Cloud model for service selection', *Proceedings of Computer Communications Workshops on Cloud Computing (INFOCOM WKSHPs), 30th IEEE Conference on*, 10–15 April 2011, Shanghai, China, pp.666–671.
- Sun, Q.B., Wang, S.G., Zou, H. and Yang, F.C. (2011) 'QSSA: a QoS-aware service selection approach', *International Journal of Web and Grid Services*, Vol. 7, No. 2, pp.147–169.
- Wang, S.G., Sun, Q.B. and Yang, F.C. (2010) 'Towards web service selection based on QoS estimation', *International Journal of Web and Grid Services*, Vol. 6, No. 4, pp.424–443.
- Xi, C., Xudong, L., Zicheng, H. and Hailong, S. (2010) 'RegionKNN: a scalable hybrid collaborative filtering algorithm for personalized web service recommendation', *Proceedings of Web Services (ICWS), 2010 IEEE International Conference on*, 5–10 July 2010, pp.9–16.
- Yu, T., Zhang, Y. and Lin, K-J. (2007) 'Efficient algorithms for web services selection with end-to-end QoS constraints', *ACM Transactions on the Web*, Vol. 1, No. 1, pp.1–26.
- Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J. and Chang, H. (2004) 'QoS-aware middleware for web services composition', *IEEE Transactions on Software Engineering*, Vol. 30, No. 5, pp.311–327.
- Zheng, Z. and Lyu, M.R. (2009) 'A QoS-aware fault tolerant middleware for dependable service composition', *Proceedings of 39th IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2009*, 29 June–2 July, 2009, Lisbon, Portugal, pp.239–248.
- Zheng, Z. and Lyu, M.R. (2010) 'Collaborative reliability prediction of service-oriented systems', *Proceedings of 32nd ACM/IEEE International Conference on Software Engineering, ICSE 2010*, 1–8 May, 2010, Cape Town, South Africa, pp.35–44.
- Zheng, Z., Zhang, Y. and Lyu, M.R. (2010) 'Distributed QoS evaluation for real-world web services', *Proceedings of the 2010 IEEE International Conference on Web Services, 2010*, pp.83–90. **AUTHOR PLEASE SUPPLY LOCATION**
- Zibin, Z. and Lyu, M.R. (2008) 'A distributed replication strategy evaluation and selection framework for fault tolerant web services', *Proceedings of Web Services, 2008. ICWS '08. IEEE International Conference on*, 23–26 September, 2008, pp.145–152.
- Zibin, Z., Hao, M., Lyu, M. R. and King, I. (2011) 'QoS-aware web service recommendation by collaborative filtering', *Services Computing, IEEE Transactions on*, Vol. 4, No. 2, pp.140–152.

*Proceedings of the IEEE International Conference on Web Services (ICWS 2010), 5-10 July, Miami, China, pp. 83-90.*

## Notes

<sup>1</sup><http://webservices.seekda.com/>

<sup>2</sup><http://www.wsdream.net>