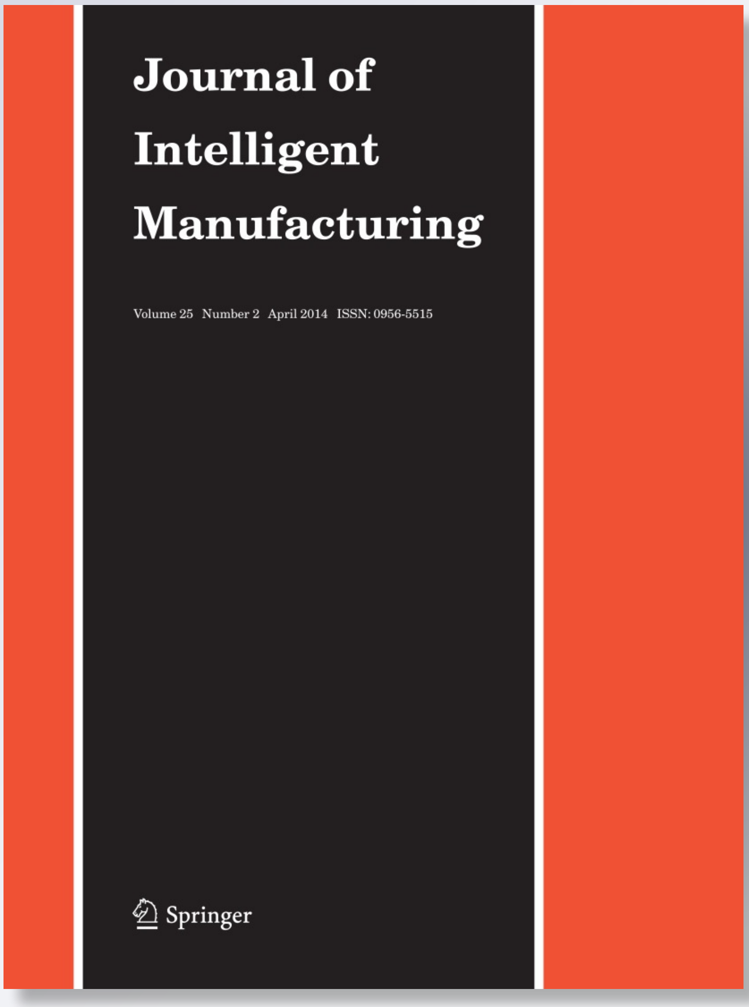
Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing

Shangguang Wang, Zhipiao Liu, Qibo Sun, Hua Zou & Fangchun Yang

Journal of Intelligent Manufacturing

ISSN 0956-5515
Volume 25
Number 2

J Intell Manuf (2014) 25:283-291
DOI 10.1007/s10845-012-0661-6


Journal of Intelligent Manufacturing
Volume 25 Number 2 April 2014 ISSN: 0956-5515
Springer

J Intell Manuf (2014) 25:283–291
DOI 10.1007/s10845-012-0661-6

Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing

Shangguang Wang · Zhipiao Liu · Qibo Sun · Hua Zou · Fangchun Yang

Received: 1 October 2011 / Accepted: 15 May 2012 / Published online: 25 May 2012

Abstract Cloud computing promises to provide high quality, on-demand services with service-oriented architecture. However, cloud service typically come with various levels of services and performance characteristics, which makes Quality of Cloud Service (QoCS) high variance. Hence, it is difficult for the users to evaluate these cloud services and select them to fit their QoCS requirements. In this paper, we propose an accurate evaluation approach of QoCS in service-oriented cloud computing. We first employ fuzzy synthetic decision to evaluate cloud service providers according to cloud users' preferences and then adopt cloud model to computing the uncertainty of cloud services based on monitored QoCS data. Finally, we obtain the evaluation results of QoCS using fuzzy logic control. The simulation results demonstrate that our proposed approach can perform an accurate evaluation of QoCS in service-oriented cloud computing.

Keywords Service-oriented cloud computing · Cloud service · QoCS · Fuzzy synthetic decision · Cloud model · Fuzzy logic control

S. Wang (✉) · Z. Liu · Q. Sun · H. Zou · F. Yang
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
e-mail: sgwang@bupt.edu.cn

Z. Liu
e-mail: zpliu@bupt.edu.cn

Q. Sun
e-mail: qbsun@bupt.edu.cn

H. Zou
e-mail: hauzou@bupt.edu.cn

F. Yang
e-mail: fcyang@bupt.edu.cn

Introduction

Recently, many large organizations such as Google, Amazon, etc. has invested in large data centers. These investments were done in order to satisfy growing customer requirements. However, these data centers are not very flexible and their operation and maintenance cost is important. Hence, in order to reduce data centers overall cost, these organizations have started moving to cloud computing, as service providers.

Cloud computing is based on Visualization and Service-Oriented Architecture (SoA). As shown in Fig. 1, in SOA, a service can be a piece of software (Software as a Service: SaaS), a platform (Platform as a Service: PaaS), or a part of the infrastructure itself (Infrastructure as a Service: IaaS) (Chazalet 2010a). We use the term cloud service to represent these services. The goal of cloud computing is to optimize the usage of physical and software resources, improve flexibility and automate management, which is thus seen as a way to reduce costs and to increase revenue for cloud service providers (Hoi and Trieu 2010; Chazalet 2010b). The success of cloud computing will depend on how effectively it will be able to instantiate and dynamically maintain computing platforms, construct out of cloud services (or sources), that meet arbitrarily varying service requirements of cloud costumers (Ferretti et al. 2010). Typically, these services will be characterized by Quality of Cloud Service (QoCS) requirements, such as timeliness, scalability, high availability, trust, security, and so on.

A successful cloud service in service-oriented cloud computing has two main objectives: to provide the needed functionality and to provide the needed QoCS (Stantchev 2009). QoCS parameters are part of the run-time related nonfunctional properties of a cloud service. Contrary to design time related functionality properties (e.g., language of service or compliance), run-time related QoCS are performance

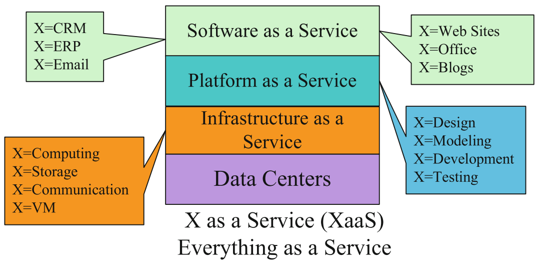


Fig. 1 Cloud service

oriented such as response time, transaction rate, availability. They change during runtime. Hence, it is difficult to assure the accurate evaluation of QoCS.

Although both the industrial and the academic research communities are showing growing interest on issues of QoCS assurance within service-oriented cloud computing, to the best of our knowledge, current cloud technology is not fully tailored to honor possible QoCS guarantees. The main challenges and complexities is that cloud services hosted (e.g., Web services, Web applications), are often characterized by high QoCS variance as follows.

Firstly, with the benefits of cloud computing also come new challenges such as reliability, security etc. that must be properly addressed. Indeed, not all cloud service providers can create equal QoCS, some are superior in computing power, some are good at offering seamless and unlimited storage, and some excel in security management while some offer the lowest cost (Hoi and Trieu 2010). In such a setting, decisions have to be made somewhere as to which applications from the cloud users are to be executed on which cloud service providers.

Secondly, it is also important to realize that there are different types of cloud users with different types of applications with different set of personalized preferences or QoCS requirements. Some applications require substantial computing and storage power while others have compelling need for maximum confidentiality. From the cloud users' perspective, their goal is to run their applications seamlessly and meet their performance, security and cost target. Therefore, matching and determining the best cloud service for a personalized application is important and often determines the success of the underlying business of the cloud users.

Finally, in the unpredictable Internet environment, any changes in network condition, time and many other factors may impact the quality of these cloud services. It is worth noting that a cloud service with consistently good QoCS performance is typically more desirable than a cloud service with a large variance on its QoCS performance. Hence, consistency should be considered as an important criterion for the evaluation of cloud service performance.

To overcome the challenges above, in this paper, based on our previous work (Wang et al. 2010; Li et al. 2008; Shangguang et al. 2010), we present an accurate evaluation approach of QoCS in service-oriented cloud computing. The core of our proposed approach is to combine the performance evaluation of cloud service providers with monitored QoCS data to obtain an accurate evaluation of QoCS for cloud users in service-oriented cloud computing. We first adopt fuzzy synthetic decision to evaluate cloud service providers according to cloud users' preferences, and then based on monitored QoCS data, we employ cloud model to calculate the uncertainty of cloud services. Finally, fuzzy logic control is used to obtain QoCS evaluation. In order to evaluate our proposed approach, we conduct a simulation. The simulation results show our proposed approach can effectively achieve an accurate evaluation of QoCS for service-oriented cloud computing.

The remainder of this paper is organized as follows. Section "Our proposed approach" describes our proposed approach including personalized evaluation, consistency evaluation, and synthetic evaluation. Simulations in section "Simulation evaluation" demonstrate the benefits of our approach. Section "Related work" reviews related work about QoCS evaluation. Finally, section "Conclusion" concludes the paper.

Our proposed approach

As shown in the Fig. 2, our proposed approach contains three modules. Module 1 (section "Personalized evaluation") is to evaluate the performance of cloud services according to cloud users' personalized preferences by using fuzzy synthetic decision, which makes QoCS accurate and unfair for different cloud users that select cloud service providers with their personalized preferences. Module 2 (section "Uncertainty evaluation") is to computing the uncertainty of QoCS by using cloud model, which determines the set of the services with consistent QoCS, and the set of the services with

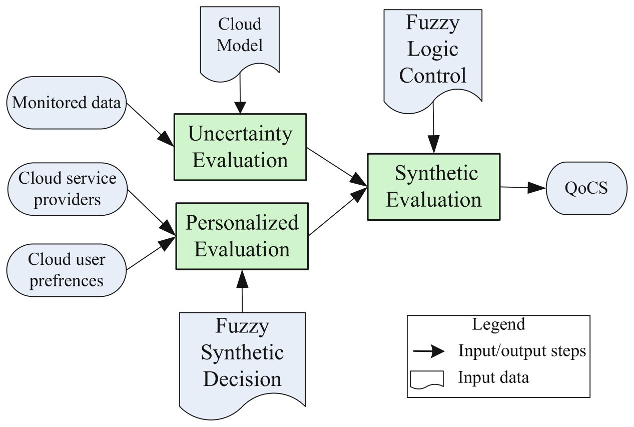


Fig. 2 Procedures of our proposed approach

inconsistent QoCS. Based on the output of the two modules above, Module 3 (section "Synthetic evaluation") involves evaluate synthetically QoCS by using fuzzy logic control.

Personalized evaluation

In traditional web service environment, it is hard to obtain the observed QoS values. But, in cloud computing environment, because the invoked services are also often deployed in the same cloud platform, it is much easier to get the observed QoCS information.

In the section, we apply fuzzy synthetic decision to perform the personalized evaluation of cloud service providers and cloud user preferences. Because Fuzzy Synthetic Decision (FSD) (Zadeh 1965) has various attributes concerning evaluation of objects and performs a comprehensive assessment and general appraisal on related factors to produce the overall assessment (Kuo and Chen 2006), it is suitable for personalized evaluation.

In the personalized evaluation of a cloud service provider, we suppose that domain $E_1 = \{e_{11}, e_{12}, e_{13}, e_{14}\}$ denotes a set of evaluated factors of a cloud service provider from a cloud user. Because the personalized preference of each cloud user may be different on the same cloud service provider. The elements of the set may be some of Computing power, Seamless, Storage, security, trust, and so on. $R = \{r_1, r_2, r_3, r_4, r_5\}$ denotes a set of evaluation ranks, and $r_j (j = 1 \ldots 5)$ denotes a probable evaluation which is described as very bad, bad, normal, good and excellent as shown in Fig. 3. For a cloud service provider, different cloud users may give different probable evaluation that is very personalized. Then the cloud service provider has a fuzzy relation matrix M_1 from E_1 to R by the following:

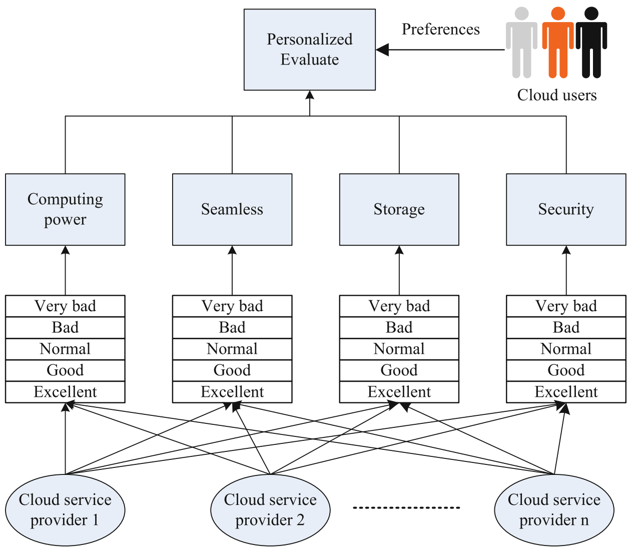


Fig. 3 Procedures of personalized evaluation

$$M_1 = (m_{ij})_{4\times 5} = \begin{bmatrix} m_{11} & \cdots & m_{15} \\ \vdots & \vdots & \vdots \\ m_{41} & \cdots & m_{45} \end{bmatrix}, \tag{1}$$

where $m_{ij} (i = 1 \ldots 4)$ denotes the membership degree, the appraisal object is measured as m_j considering attribute e_{1i}.

In the application of fuzzy synthetic decision, because the obtained appraisal matrix M_1 is not enough to appraise the cloud service provider yet, a fuzzy subset W_1 in E_1, called cloud users' preference set, $W_1 = \{w_{11}, w_{12}, w_{13}, w_{14}\}$ ($\sum_{i=1}^{4} w_{1i} = 1$ and $0 \le w_{1i} \le 1$) is needed. In this paper, W_1 can be obtained from the analytic hierarchy process (Saaty 1980). The cloud users' preference set W_1 denotes the relative importance of the various evaluated factors expressed by cloud users. Furthermore, a decision making set F_1 in R, is also needed, and denotes the overall fuzzy appraisal by the following:

$$F_1 = W_1 \circ M_1 = \{f_{11}, f_{12}, f_{13}, f_{14}, f_{15}\}, \tag{2}$$

with

$$f_{1j} = \sum_{i=1}^{4} (w_{1i} \times r_{ij}),$$

where "$\circ$" stands for a kind of fuzzy operation. There are three basic fuzzy operations: intersection, union, and complement in this study.

In this paper, for the cloud users' preference, we use a appraisal grade set to obtain the personalized evaluation score of the cloud service provider by defuzzification. Defuzzification means calculate the crisp value of fuzzy number. The crisp value can approximately represent the deterministic characteristics of the fuzzy reasoning process based on the assessment matrix, and help convert the uncertainty into an applicable action in solving real world problems. The defuzzification way is as follows:

$$iw_1 = F_1 \cdot S_1, \tag{3}$$

where iw_1 is a defuzzification score, F_1 is the decision making set, and S_1 is the appraisal grade set. For instance, excellent, good, normal, bad and very bad can be defined in appraisal grading as 1, 0.8, 0.6, 0.3, and 0, respectively. Hence, by using (3), the personalized evaluation score on the cloud service provider can be obtained according to cloud users' preferences. Similarly, we also can obtain the personalized evaluation score of cloud service providers by this way.

Uncertainty evaluation

In this module, we adopt cloud model to compute the uncertainty by transforming quantitative QoCS values (transaction logs) to qualitative QoCS concept (uncertainty level).

According to the uncertainty level, a cloud service with consistently good QoCS can be distinguished from those services with a large QoCS variance. It mainly contains three steps: Cloud model selection, Compute uncertainty, and Distinguish uncertainty.

Cloud model selection

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

Definition 1 Let U be the set as the universe of discourse, and C a qualitative concept associated with U. The membership degree of quantitative numerical representation x (x is a random realization of the qualitative concept C) in U to the concept C, $\mu(x) \in [0, 1]$ ($\mu(x)$ is the membership degree of x to C), is a random number with a stable tendency, that is as in (4):

$$\mu : U \rightarrow [0, 1],\ \forall x \in U,\ x \rightarrow \mu(x) \tag{4}$$

The distribution of x in the universe of discourse U is called cloud $C(X)$, and x is called a cloud drop. The overall characteristics of cloud model may be reflected by its three numerical characteristics: Expected value (Ex), Entropy (En) and Hyper-Entropy (He).

Because a lot of uncertainty concepts behave normal clouds in social and natural phenomena (Li et al. 1998), normal cloud is the most basic and important cloud, and it is an efficient tool to express linguistic atom. Hence, as shown in Fig. 4, we mainly apply normal cloud model. Figure 4

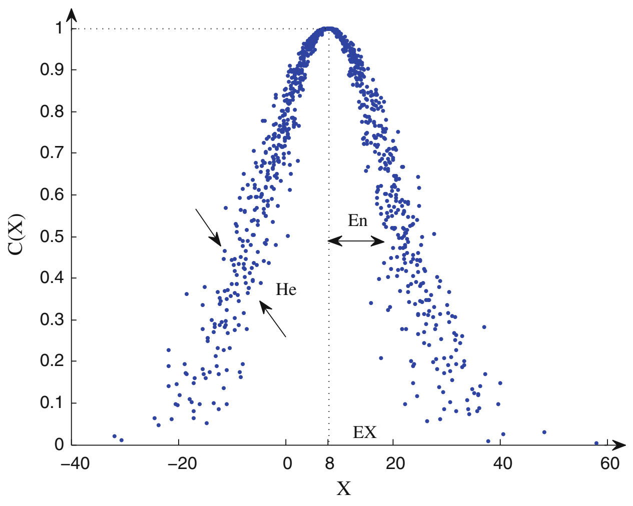


Fig. 4 Three numerical characteristics of cloud model

shows the three numerical characteristics of cloud model, where the number of cloud drops is 1,000. In the discourse universe, Ex is the expectation of the cloud drops distribution, and is the position representing the qualitative concept best, corresponding to the center of the cloud gravity. En is the uncertainty measurement of the qualitative concept, which is decided by the randomness and the fuzziness of the concept. The randomness reflects the discrete degree of cloud drops which can represent the concept. The fuzziness reflects the interval of cloud drop accepted by the concept. He is the uncertain measurement of entropy, that is, the entropy of the entropy En, which is decided by the randomness and the fuzziness of entropy En. Then, a vector $NC = Ex, En, He$ is called the eigenvector of cloud model, which integrates the fuzziness and random of language value represented by quality approach.

Compute uncertainty

In this paper, we apply these three numerical characteristics of backward cloud generator (Li et al. 1998) (Algorithm 1) to denote the uncertainty of QoCS by transforming QoCS quantitative values to qualitative concept.

We first take two cloud store services CSS and CST that offer the similar cloud store service as an example to illustrate the different implications for QoCS evaluation. In this example, the performance of CSS and CST is recorded by a series of transaction logs, which helps capture the actual QoCS delivered by each provider in practical application. Because in dynamic environment, these cloud 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 CSS and CST, respectively. These transactions are represented as $(css_1, \ldots, css_5)$ and $(cst_1, \ldots, cst_5)$ as shown in Table 1. Table 1 gives response time values of these transactions. The aggregated QoCS value ($\overline{css}$ and $\overline{cst}$), which are obtained by averaging all transactions, are given in the last row of Table 1.

As shown in Table 1, the aggregate QoCS values of CSS is larger than that of CST. In traditional evaluation approach, CST is usually selected as a good service in service-oriented cloud computing. However, after analyzing each transaction of these two cloud services, we find this way may be unreasonable. For example, although the average response time of CST is slightly less than that of CSS, the three transactions (css_2, css_3, css_4) of CSS is less than (cst_2, cst_3, cst_4) of CST. The response time of CST is more volatile than that of CSS. Hence, the response time of CSS is less than that of CST in most transactions. Moreover, CST is with a large variance on its QoCS, while CSS is with consistently good QoCS.

Algorithm 1 Compute uncertainty via backward cloud generator

```
Input: $n$ transactions of a cloud service, i.e., n cloud drops $\{x_1, x_2, \cdots, x_n\}$.
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 cloud service on its QoCS can be calculated by $Ex = \bar{X}$;
3. The Entropy of the cloud service on its QoCS 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 $He = \sqrt{S^2 - En^2}$.
```

Table 1 A set of service transactions

Cloud store Service: CSS		Cloud store service: CST	
ID	Response time (ms)	ID	Response time (ms)
css_1	29	cst_1	16
css_2	26	cst_2	40
css_3	30	cst_3	31
css_4	26	cst_4	34
css_5	26	cst_5	12
$\overline{css}$	27.4	$\overline{cst}$	26.6

In this study, if CST is selected as a business application, the actual execution result of CST may deviates from cst, leading to poor service quality. Thus, it may be obvious that CSS is more stable than CST, and CSS as a business application may be more suitable than CST. So we adopt the backward cloud generator algorithm of cloud model to distinguish theses cloud services.

Distinguish uncertainty

The section is to distinguish one cloud service with a consistently good QoS from another cloud service with a large variance on its QoCS.

By using Algorithm 1, these response time values can be seen as quantitative QoCS values expressed by five cloud drops, i.e., $(css_1, \ldots, css_5)$ or $(cst_1, \ldots, cst_5)$. The qualitative QoCS concept (uncertainty level) of each service can be expressed by its eigenvector. Then these eigenvectors of CSS and CST can be calculated. Because En of CSS is smaller than that of CST and He of CSS is also smaller that that of CST ($2.11 < 12.63$ and $3.10 < 144.25$), the uncertainty level of CSS is smaller than that of CST. Thus, the QoCS of CSS is consistently good but CST is with a large variance on its QoCS. So the CSS should be selected as a business application rather than CST.

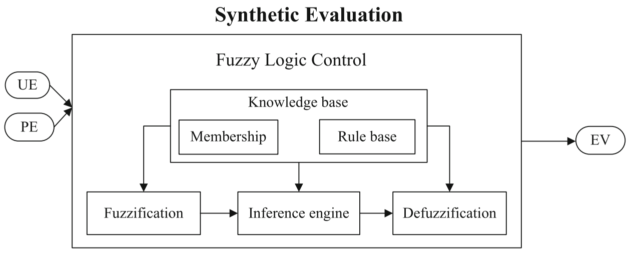


Fig. 5 Procedures of synthetic evaluation

Because cloud model run over collected data, this process can be expensive in terms of computation time. Given that for the problem considered here the process of cloud model is independent of any individual service request, it does not need to be conducted online at request time. Hence, in order to apply the cloud model to QoCS evaluation, we set the parameters λ and h as the thresholds of En and He according to different cloud service environments. For example, if $En \leq \lambda$ and $He \leq h$, the cloud services with a large QoCS variance and the cloud services with a consistently good QoCS can be distinguished in service-oriented cloud computing.

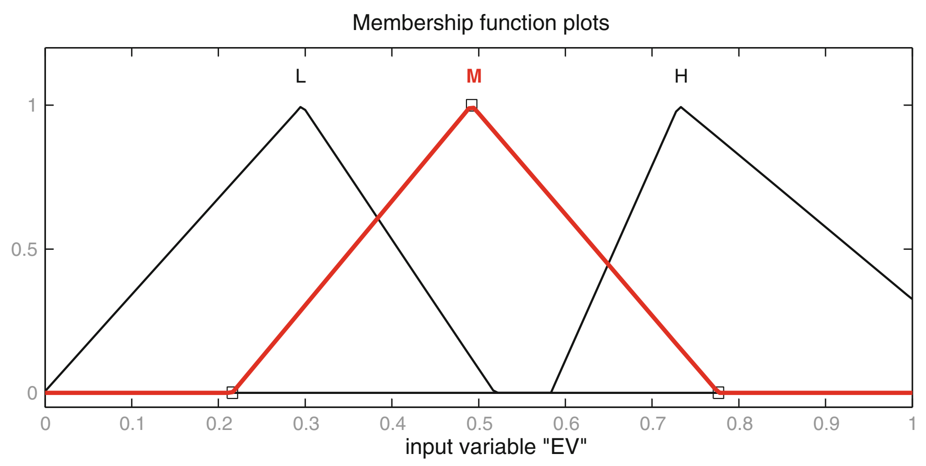
Synthetic evaluation

Fuzzy Logic Control (FLC) (Freeman 1994) is a development of fuzzy logic that allows for extremely precise control of robotic systems. A FLC is a control system based on fuzzy logic (a mathematical system) that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical or digital logic, which operates on discrete values of either 1 or 0 (true or false respectively). Hence, FLC accounts for ambiguities in the data by giving it a level of confidence rather than declaring the data simply true or false.

As shown in Fig. 5, based on the outputs of the two modules above, we adopt FLC to obtain accurate QoCS evaluation. In the module, we take the output (UE) of Uncertainty Evaluation module and the output (PE) Personalized Evaluation module as the inputs of Synthetic Evaluation module, with EV as its output where EV is the evaluation value of QoCS. By using the technology of (Chuang and Chan 2008), EV should be accurately evaluated with the following four steps:

(1) *Memberships* We set UE and PE as inputs, with the EV (QoCS evaluation value) as output. In this study, we adopt triangular membership function to design the synthetic evaluation. Figure 6 gives one of the triangular membership functions as an example to understand the steps.

Fig. 6 Membership function



(2) *Fuzzification* By using the defined membership functions, we translate the input values into a set of linguistic values and assign a membership degree for each linguistic value by using triangular membership functions (extra experimental results shown that three membership functions and the triangular membership function are very suitable to our proposed approach).

(3) *Inference* The inference engine makes decisions based on fuzzy rules.[1] All fuzzy rules that apply are invoked, using the membership functions and truth values obtained from the inputs, to determine the result of the rule. This result in turn will be mapped into a membership function and truth value controlling the output variable. In this study, FLC is represented with three fuzzy sets: "low" (L), "medium" (M), and "high" (H) where L denotes the input/output is a low (little) value, M denotes the input/output is a medium value, H denotes the input/output is a high (large) value. These fuzzy sets determine the shape and location of the membership functions. Each rule is an IF-THEN clause in nature, which determines the linguistic value of CUS according to the linguistic values of all context factors. For example, If (PE is L) and (UE is M) then (UV is M). As shown in Table 2, in this study, we designed 9 rules.

(4) *Defuzzification*. We adopt the most common defuzzification method, called center of gravity (Van Broekhoven and De Baets 2009) to obtain EV with a value in the range [0, 1].

According to the 4 steps above, we obtain the evaluation results of QoCS, and then cloud users can determine which cloud service is suitable to them for their applications. However, note that our approach only perform the current evaluation in a certain period time and it does not match all the phases. Moreover, note that our approach does not match at all with the problem presented. Developing other approaches for all the problems are areas for future research.

[1] http://en.wikipedia.org/wiki/Fuzzycontrolsystem.

Table 2 Fuzzy rules

ID	Fuzzy rulers
1	If (UE is L) and (PE is L) then (UV is H)
2	If (UE is L) and (PE is M) then (UV is H)
3	If (UE is L) and (PE is H) then (UV is M)
4	If (UE is M) and (PE is L) then (UV is H)
5	If (UE is M) and (PE is M) then (UV is M)
6	If (UE is M) and (PE is H) then (UV is L)
7	If (UE is H) and (PE is L) then (UV is M)
8	If (UE is H) and (PE is M) then (UV is L)
9	If (UE is H) and (PE is H) then (UV is L)

Simulation evaluation

In this section, we present an experimental evaluation of our approach by conducting two simulations. The first simulation results indicate that our approach has much higher accuracy than other approaches, and the second simulation results show that the time cost of our approach is much shorter than other approaches.

Simulation setup

It is worth noting that due to the limited availability of service data currently, some traditional quality measure approaches (Ardagna and Pernici 2007; Hwang et al. 2007; Pei et al. 2009) in service computing, used simulation data for performance evaluation. Hence, in our simulation, we focus on simulation data for QoCS evaluation based on web service cloud environment.

To evaluate our approach, Some comparisons have been performed on QoCS evaluation with two existing approaches

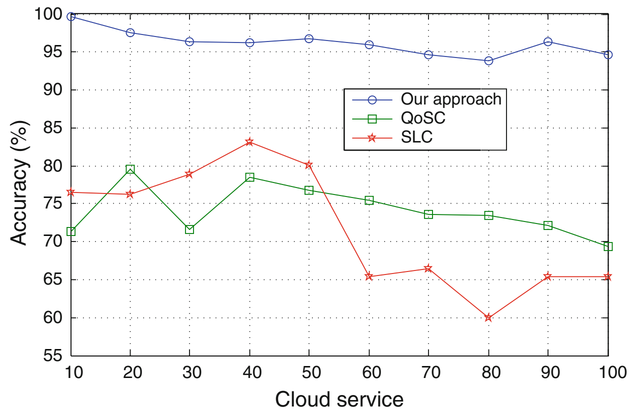


Fig. 7 Comparison results on accuracy

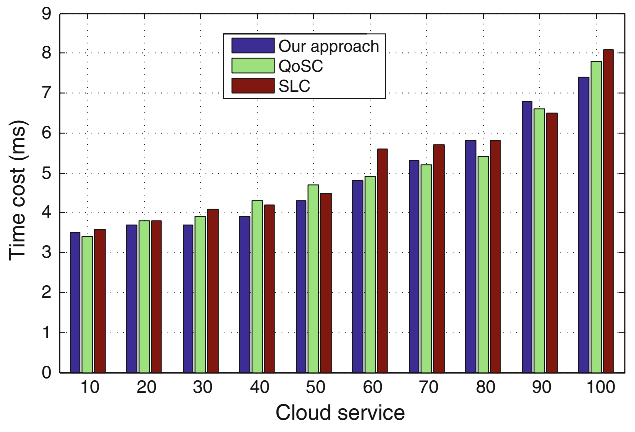


Fig. 8 Comparison results on time cost

of Chazalet (2010a) and Ferretti et al. (2010) in cloud computing. For the purpose of illustration, we use the letters "QoSC" to represent the model in Ferretti et al. (2010). Similarly, the letters "SLC" represent the approach in Chazalet (2010a).

All the simulations are conducted on the same computer with Intel Core2 2.8 GHz processor, 2.0 GB of RAM, Windows XP SP3, and Matlab 7.6. All approaches are run for 10 times and all results are reported, on average.

Comparison on accuracy

Definition 2 We define the accuracy of QoCS evaluation as the difference between the actual value (AV_i) and the evaluated value (q_i), i.e., $100\,\% \times |q_i - AV_i| \big/ AV_i$. The higher the accuracy is, the evaluation approach is better.

As shown in Fig. 7, we investigate the accuracy of the three approaches based on 100 cloud services. From Fig. 7, the accuracy of our approach is 96.2 %, which is much higher than 74.2 % of QoSC and 71.7 % of SLC. These results mean that our approach is the best. Hence, the comparison results show that our approach can obtain the most accurate QoCS evaluation.

Comparison on time cost

For the purpose of illustration, the letters "Time1" represent the computation time of approach. Similarly, the letters "Time2" and "Time0" represent the computation time of QoSC and SLC, respectively.

As shown in Fig. 8, we investigate the time cost results of all approaches with respect to the number of cloud services. From the results, regardless of the number of services, time cost of our approach is the lest. It is about 5 milliseconds only.

In a word, based on the Figs. 7 and 8, the evaluation performance of our approach is best. This means that our approach can achieve the accurate evaluation of QoCS for cloud users.

Related work

Cloud computing has gained much attention in the last few years. However, to the best of our knowledge, the topic of QoCS provision in cloud computing environments has not received much attention as yet. Nevertheless, some scientific papers recently published reveals a growing interest in this topic in both the industrial and academic research communities. Although in our previous work (Wang et al. 2010; Li et al. 2008; Shangguang et al. 2010), we proposed some QoS evaluation approaches, it may not be suitable to cloud computing environment. In this section, we briefly review a few papers on this topic.

Chazalet (2010a) presented an innovating architecture for Service Level Checking that has been applied to the cloud Computing context. The approach proposed is feasible and functional, which allowed the separation of concerns between information collection and monitoring on one side and contract verification on the other. It also allowed the precise specification of data and data exchanges between the data collection and the SLC layers.

Ferretti et al. (2010) described a middleware architecture that they have designed in order to enable platforms, constructed out of cloud computing resources, to meet the QoS requirements of the applications they host. An initial evaluation of this architecture, carried out through simulation, has provided them with very encouraging results that confirm the adequacy of our design approach.

Ardagna and Pernici (2007) proposed multiple QoS measure models (aggregation functions) to five QoS attributes such as availability, execution time, data quality, price and reputation. These models are simple and effective to service composition application, but they are lacking in considering existing malicious QoS data from service providers, which makes the services obtained fail to satisfy customers' requests. When customers' QoS requests cannot be met (no solution), the authors adopted QoS negotiating between customers and service providers to perform the second optimization (reoptimization) for satisfying their requests.

Hwang et al. (2007) analyzed the QoS metrics for web services and proposed a probability-based QoS model. A QoS measure of an atomic or composite web service is quantified as a probability mass function. The authors described algorithms to compute the QoS measures of a web service workflow from those of its constituent web services, introduced the problem of computing the least error QoS probability mass function during composing a web service workflow, and provided a dynamic programming formulation for the optimal solution and an efficient approximation heuristic.

Pei et al. (2009), based on the fact that users and providers can express their QoS in very flexible ways, the authors made the management of QoS a very complex task, the authors proposed a computing-oriented description of QoS and an approach to QoS-based service evaluation based on hierarchical constraint logic programming (HCLP). This approach allows web service designer to describe the real values that non-functional properties (NFPs) will expose at run time by means of proper mathematical functions, and allows users to specify their preferences on NFPs by exploiting HCLP and introducing tendency functions.

Ghosh et al. (2010) quantified the effects of variations in workload (e.g., job arrival rate, job service rate), faultload (e.g., machine failure rate) and system capacity (PMs in each pool) on IaaS cloud service quality. By using interacting stochastic models, they described a fast method suitable for analyzing the service quality of large sized IaaS clouds. The approach is tractable and capture many realistic features such as provisioning decision, VM provisioning etc. of a large sized cloud, with reduced complexity of analysis.

Yigitbasi et al. (2009) presented a portable, extensible and easy-to-use framework for performance analysis of cloud computing environments. It is portable in the sense that it is implemented in Python which is a platform-independent programming language, and it is extensible in the sense that it can be extended to interact with many cloud computing environments and it can also be extended with different scheduling algorithms.

Jackson (2010) quantitatively examined the performance of a set of benchmarks designed to represent a typical high performance computing workload run on Amazon EC2. They provided the broadest evaluation to date of application performance on visualized cloud computing platforms and analyzed the impact of virtualization based on the communication characteristics of the application. Newton and Arockiam (2011) considered four primary parameters are such as reliability, delay, jitter, bandwidth which together determine the QoS and proposed a technique to predict reason(s) for deterioration in the QoS and to identify the algorithm(s)/mechanism(s) responsible for the deterioration. There are some other related studies such as search quality improvement (Qi and Bouguettaya 2010), trusted Software dissemination system (Pyshkin and Kuznetsov 2010), distributed search engines (Chuan et al. 2011), and integrated management platform (Dominguez-Sal et al. 2010).

Conclusion

In this paper, we present an accurate evaluation approach of QoCS in service-oriented cloud computing. Its core is to combine the performance evaluation of cloud service providers with monitored QoCS data to obtain an accurate evaluation of QoCS for cloud users.

We first adopt fuzzy synthetic decision to evaluate cloud service providers according to cloud users' preferences, and then based on monitored QoCS data, we employ cloud model to calculate the uncertainty of cloud services. Finally, fuzzy logic control is used to obtain QoCS evaluation. In order to evaluate our proposed approach, we conduct a simulation. The simulation results demonstrate our proposed approach can effectively achieve an accurate evaluation of QoCS for service-oriented cloud computing.

In our future work, we will continue to investigate more efficient evaluation approaches based on fuzzy control (Jeguirim et al. 2011; Erginel 2010; Adali et al. 2009) to help real-world cloud service users find appropriate cloud services according to their QoCS requirements in the near future. The other focuses on the application in ubiquitous environment (Kryvinska et al. 2010; Lee et al. 2011; Lim et al. 2010; Oh 2010), especially,ubiquitous Web service-based manufacturing environment (Wang et al. 2009; Lee et al. 2009).

Acknowledgments The work is supported by RFDP (20110005130 001); 863 program (2011AA01A102); China Postdoctoral Science Foundation (2011M500226) and NCET (100263).

References

Adali, M. R., Taskin, M. F., & Taskin, H. (2009). Selecting the optimal shift numbers using fuzzy control model: a paint factory's facility application. *Journal of Intelligent Manufacturing, 2*, 267–272.

Ardagna, D., & Pernici, B. (2007). Adaptive service composition in flexible processes. *IEEE Transactions on Software Engineering, 33*, 369–384.

Chazalet, A. (2010a). Service level checking in the cloud computing context. In *3th IEEE international conference on cloud computing* (pp. 297–304).

Chazalet, A. (2010b). Service level agreements compliance checking in the cloud computing: architectural pattern, prototype, and validation. In *5th International conference on software engineering advances* (pp. 184–189).

Chuan, D., Lin, Y., Linru, M., & Yua, C. (2011). Towards a practical and scalable trusted software dissemination system. *Journal of Convergence, 2*, 53–60.

Chuang, S. N., & Chan, A. T. S. (2008). Dynamic QoS adaptation for mobile middleware. *IEEE Transactions on Software Engineering, 34*, 738–752.

Dominguez-Sal, D., Perez-Casany, M., & Larriba-Pey, J. L. (2010). Cooperative cache analysis for distributed search engines. *International Journal of Information Technology, Communications and Convergence, 1*, 41–65.

Erginel, N. (2010). Modeling and analysis of packing properties through a fuzzy inference system. *Journal of Intelligent Manufacturing, 6*, 869–874.

Ferretti, S., Ghini, V., Panzieri, F., Pellegrini, M., & Turrini, E. (2010). QoS-aware clouds. In *3th IEEE international conference on cloud computing* (pp. 321–328).

Freeman, A. (1994). Fuzzy systems for control applications: The truck backer-upper. *The Mathematica Journal, 4*, 64–69.

Ghosh, R., Trivedi, K. S., Naik, V. K., & Kim, D. S. (2010). End-to-end performability analysis for infrastructure-as-a-service cloud: An interacting stochastic models approach. In *16th IEEE Pacific Rim international symposium on dependable computing* (pp. 125–132).

Hoi, C., & Trieu, C. (2010). Ranking and mapping of applications to cloud computing services by SVD. In *1th IEEE/IFIP intenational workshops on network operations and management symposium* (pp. 362–369).

Hwang, S. Y., Wang, H., Tang, J., & Srivastava, J. (2007). A probabilistic approach to modeling and estimating the QoS of web-services-based workflows. *Information Sciences, 177*, 5484–5503.

Jackson, K. R., Ramakrishnan, L., Muriki, K., Canon, S., Cholia, S., Shalf, J., et al. (2010). Performance analysis of high performance computing applications on the Amazon web services cloud. In *IEEE second international conference on in cloud computing technology and science* (pp. 159–168).

Jeguirim, S. E. G., Dhouib, A. B., Sahnoun, M., Cheikhrouhou, M., Schacher, L., & Adolphe, D. (2011). The use of fuzzy logic and neural networks models for sensory properties prediction from process and structure parameters of knitted fabrics. *Journal of Intelligent Manufacturing, 6*, 873–884.

Kryvinska, N., Thanh, D. V., & Strauss, C. (2010). Integrated management platform for seamless services provisioning in converged network. *International Journal of Information Technology, Communications and Convergence, 1*, 77–91.

Kuo, Y. F., & Chen, P. C. (2006). Selection of mobile value-added services for system operators using fuzzy synthetic evaluation. *Expert Systems with Applications, 30*, 612–620.

Lee, M., Yoon, H., Shin, H., & Lee, D. G. (2009). Intelligent dynamic workflow support for a ubiquitous Web service-based manufacturing environment. *Journal of Intelligent Manufacturing, 20*, 295–302.

Lee, M., Lee, J., Kim, K., & Park, S. S. (2011). Evaluating service description to guarantee quality of U-service ontology. *Journal of Information Processing Systems, 7*, 287–298.

Li, D., Cheung, D., Shi, X., & Ng, V. (1998). Uncertainty reasoning based on cloud models in controllers. *Computers and Mathematics with Applications, 35*, 99–123.

Li, F., Yang, F., Shuang, K., & Su, S. (2008). A policy-driven distributed framework for monitoring quality of web services. In *6th IEEE international conference on web services* (pp. 708–715).

Lim, H., Jang, K., & Kim, B. (2010). A study on design and implementation of the ubiquitous computing environment-based dynamic smart on/off-line learner tracking system. *Journal of Information Processing Systems, 6*, 609–620.

Newton, P. C., & Arockiam, L. (2011). A novel prediction technique to improve quality of service (QoS) for heterogeneous data traffic. *Journal of Intelligent Manufacturing,, 6*, 867–872.

Oh, S. (2010). New role-based access control in ubiquitous e-business environment. *Journal of Intelligent Manufacturing, 21*, 607–612.

Pei, L., Comerio, M., Maurino, A., & De Paoli, F. (2009). An approach to non-functional property evaluation of web services. In *7th IEEE international conference on web services* (pp. 1004–1005).

Pyshkin, E., & Kuznetsov, A. (2010). Approaches for web search user interfaces: How to improve the search quality for various types of information. *Journal of Convergence, 1*, 1–8.

Qi, Y., & Bouguettaya, A. (2010). Computing service skyline from uncertain QoWS. *IEEE Transactions on Services Computing, 3*, 16–29.

Saaty, T. L. (1980). *The analytic hierarchy process*. New York: McGraw-Hill.

Shangguang, W., Qibo, S., & Fangchun, Y. (2010). An approach for QoS measure of web service with multifactor support. In *IEEE GLOBECOM workshops on web and pervaisve seccurity* (pp. 1586–1590).

Stantchev, V. (2009). Performance evaluation of cloud computing offerings. In *3th International conference on advanced engineering computing and applications in sciences* (pp. 187–192).

Van Broekhoven, E., & De Baets, B. (2009). Only smooth rule bases can generate monotone Mamdani-Assilian models under center-of-gravity defuzzification. *IEEE Transactions on Fuzzy Systems, 17*, 1157–1174.

Wang, R. C., Chang, Y. C., & Chang, R. S. (2009). A semantic service discovery approach for ubiquitous computing. *Journal of Intelligent Manufacturing, 20*, 327–335.

Wang, S. G., Sun, Q. B., & Yang, F. C. (2010). Towards web service selection based on QoS estimation. *International Journal of Web and Grid Services, 6*, 424–443.

Yigitbasi, N., Iosup, A., Epema, D., & Ostermann, S. (2009). C-Meter: A framework for performance analysis of computing clouds. In *9th IEEE/ACM international symposium on cluster computing and the grid* (pp. 472–477).

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*, 338–353.