Available online at www.ijpe-online.com

**International Journal of Performability Engineering**

vol. 13, no. 7, November 2017, pp. 1039-1047
DOI: 10.23940/ijpe.17.07.p5.10391047

# Service Selection Method based on Skyline in Cloud Environment

Yanpei Liu*, Rui Yang, Suzhi Zhang

*School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, 450002, China*

**Abstract**

In view of the fact that SaaS service cannot currently guarantee reliability and instantaneity, a dynamic SaaS service selection strategy based on Skyline (SSBS) is proposed. In this method, to select the matching services in the SaaS services library, the Skyline computation is used to reduce redundancy. Then, service selection can be made using mixed integer programming. The experimental results show that the proposed method can accurately select the most suitable could service.

*Keywords*: skyline computation; data uncertainty; service selection

(Submitted on July 25, 2017; Revised on August 30, 2017; Accepted on September 15, 2017)

(This paper was presented at the Third International Symposium on System and Software Reliability.)



## 1. Introduction

SaaS (Software as a Service) [2,6] is an application of cloud computation that provides leased online software service to achieve rapid development in the current open network environment. Thus, various kinds of applications aimed at services have emerged. With an increasing number of SaaS services deployed on cloud platforms, many of which have identical or similar functions to meet user's demand, it is imperative to be able to select SaaS services or services combinations from among numerous services in a timely and accurate manner.

Because QoS can distinguish services well, many service selection methods for QoS have been proposed by current scholars. Resource virtualization and service encapsulation of a logistics centre was studied in [5] and focuses on those technologies. The advantages of global and local choices are combined in [7] to implement a hybrid selection method that resolves global QoS constraints into local constraints and considers global factors in making a local choice. Achieving the implementation of optimal service selection can reduce the cost of service selection. In [3], the authors proposed a dynamic service selection method based on the global QoS constraint decomposition. The method is based on adaptive adjustment of fuzzy logic and the adaptive particle swarm optimization algorithm. Adaptive global QoS constraints can be disintegrated to satisfy the local constraints preferred by the user, and then this method is used to select the local optimization. Candidate services were screened with different QoSs in [8] to choose the Skyline service to participate in a composite service choice. Weeding out the candidate atomic services with poor QoS reduces the scale of candidate services and improves the efficiency of the selection method. In [9], the candidate services were screened in advance according to certain rules. This filtering process reduced the number of available services. Then, linear programming based on the global optimization method is adopted to produce the final combination scheme. Trust was introduced into the service selection in [12], proposing a dynamic service selection algorithm based on ant colony optimization to avoid the phenomenon in which all composite services implementations go through the same path. The authors of [1] studied the optimal combination problem based on per-flow and created a unified plan for multiple service selection schemes in the prediction of the user task arrival rate.

* Corresponding author.
*E-mail address*:liupeiok123@gmail.com.

The previous research improved the accuracy and reliability of service selection but still has the following deficiencies. First, it does not take the uncertain problems of SaaS service QoS data into account. Second, it matches and chooses all services in the SaaS services library. Third, it does not consider the SaaS services dynamic problem. To make up for these shortages, this paper will divide the SaaS service selection process into two phases: the service selection design phase and the service selection execution phase. The selection design phase does offline calculation by employing the Skyline calculation method and is responsible for the updating the Skyline service. Thus, offline process never affects the time performance of the service selection phase. The selection execution phase is responsible for selecting the optimal service combination that meets users' QoS global constraint.

## 2. Service selection method based on the skyline

### 2.1. Cloud model

The cloud model is a transformative model that considers the uncertainty between qualitative linguistic values and quantitative values. It was proposed by Li Deyi, who primarily considers the relevancy between randomness and fuzziness on the basis of probability theory and fuzzy set theory. It has been employed in intelligent control, fuzzy evaluation, evolutionary computation and many other fields. The next section of this paper gives parts of the related concepts and definitions for cloud model.

**Definition 1**(cloud and cloud droplet) Suppose $U$ is a quantitative domain theory denoted by a numerical value, and $C$ is the qualitative concept on space $U$ . If the quantitative value $x \in U$ is a Random implementation of the qualitative concept $C$ , the certainty degree $\mu(x) \in [0,1]$ that $x$ has for $C$ is a random number with a stable tendency $\mu : U \to [0,1]$ $\forall x \in U, x \to \mu(x)$, then $x$ 's distribution on domain theory $U$ is called as cloud and denoted as $C(x)$, in which every $x$ is called a cloud droplet.

The overall features that the cloud model expresses can be reflected by the digital characteristics of the cloud. The cloud expected value $Ex$, entropy $En$ and hyper entropy $He$ are three characteristics that represent the whole concept (this paper uses this concept to express the stability of QoS, namely, the uncertainty). The expectation of the cloud $Ex$ is located at the centre of gravity in the cloud. It is formed by the typical user values used for service property evaluations and is measured by the service QoS attribute feedback data of an average level. Entropy $En$ is the distance a cloud droplet is from $Ex$. This is indicated by user feedback data with an average deviation degree, namely, the fluctuation range of the service QoS attribute data that users can accept. $He$ expresses the thickness of the cloud and the random size of cloud droplets' uncertainty as measured by the stability of users' QoS attribute data. These three numerical characteristics are used to express the overall qualitative concepts of $C(Ex, En, He)$, which is known as the characteristic vector of the cloud model.

### 2.2. Redundant services eliminating algorithm based on Skyline computation

Skyline computation is a process that extracts data objects to form a collection of objects not dominated by any other data objects from a database. Its purpose is to choose the best or the most significant objects from huge amounts of information [4,10,11]. In recent years, Skyline has attracted more attention and has been applied to data mining, data visualization, peer-to-peer networks and many other fields. This paper specifically presents parts of the relevant Skyline concepts and definitions.

**Definition 2** (Skyline computation) The Skyline computation chooses points that are not dominated by another point s from a given $d$ -dimensional space. The so-called control means for points $\overline{p}(p_1, \ldots, p_d)$ and $\overline{q}(q_1, \ldots, q_d)$, if $\forall i \in [1, d], p_i \geq q_i$ ( $\geq$ means better than or equal to) and $\exists i \in [1, d], p_i > q_i$ ( $>$ means better than), we say $\overline{p}$ dominates $\overline{q}$ .

**Definition 3**(service domination) Supposing there is a service class $S = \{s_1, s_2\}$ that contains two candidate services $s_1$ and $s_2$, and each service has a number of $k$ QoS attributes. If $\forall i \in [1, k], s_i \geq s_i$ and $\exists i \in [1, k], s_i > s_i$, then $s_i \prec s_i$ ( $\prec$ means service domination). Each QoS attribute in service $s_1$ is better than or equal to the same attribute in service $s_2$, and at least one QoS attribute is better than that of $s_2$.

**Definition 4** (Skyline service) Skyline service refers to a candidate set of services $SkyS$ that is not dominated by other services in a service class $S$ .

For example, in a Skyline service, this paper shows how to eliminate redundant candidate services using a Skyline calculation to keep those potential candidate services that might become the service component. This process aims to reduce both the searching space of service selection methods and the computing time.

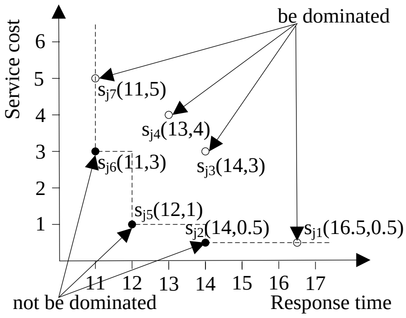



Figure 1. An example of Skyline service

Figure 1 shows an example of Skyline service in service class $S_j$. In this case, the service class $S_j$ contains seven candidate services. Each candidate service has two QoS attributes, response time and service cost (the value is the expected value $Ex$ of the backward cloud algorithm output), and each candidate service can be thought of as a single point in the two-dimension space. As defined by definitions 3 and 4, because service $s_{j2}$ is not dominated by any other service in the service class ( there is no other service with a shorter response time and lower service cost), then service $s_{j2}$ is better, so service $s_{j2}$ is the Skyline service ( $s_{j2} \in SkyS_j$ ). Similarly, $s_{j5}$ and $s_{j6}$ are also Skyline services. Therefore, the Skyline service of service class $S_j$ is $SkyS_j = \{s_{j2}, s_{j5}, s_{j6}\}$ . Among these, service $s_{j2}$ dominates $s_{j1}$ and $s_{j3}$, service $s_{j5}$ dominates $s_{j4}$ , and service $s_{j6}$ dominates $s_{j7}$ .

Through the above example, we can see that the number of candidate services in service class $S_j$ decreased from the original 7 to 3, eliminating four redundant candidate services, which achieves the purpose of reducing the service selection search space. Of course, in determining the Skyline service of each service class, it is necessary to compare multiple QoS attributes of all the candidate service. When the number of candidate services is large, the calculation process may take more time. However, because service selection and the Skyline computation function independently, this paper uses the existing Skyline computation method to perform offline computations (it is responsible for the update of Skyline services). Because it belongs to the service selection design stage, it will not affect the time performance of service selection stage. However, the SaaS service in the library service is constantly changing. In other words, new services join the library and service failures or QoS changes will occur. These changes lead to positional changes of service in the service space and thus affect the Skyline service.

### *2.3. Skyline service dynamic maintenance algorithm*

The dynamic SaaS service environment includes service additions, service failures and service QoS changes. Service QoS changes can be viewed as the integration of two processes: original service invalidation and new service generation. Therefore, for convenience, this paper considers only two cases: service additions and service failures. The Skyline maintenance algorithms for service addition and service failure are described below.

(1) Skyline maintenance algorithm for service addition

In a two-dimensional tape model, there are two service attributes for response time and service cost. There are two tapes R and C, and the position of service $s_t$ on the tapes can be respectively defined as $R_t$ and $C_t$ . Algorithm 1 is the Skyline maintenance algorithm for service adding.

| Algorithm 1. DSCA_AddService. |
|---|
| Input: R-Tape, C-Tape, $s_t(q_r, q_c)$ |
| Output: R- Tape, C-Tape |
| 1. $R_t \leftarrow$ Compare $q_r$ with values in R-Tape |
| 2. $C_t \leftarrow$ Compare $q_c$ with values in C-Tape |
| 3. if $R_t < C_t$ |
| 4. Delete services belong to $[R_t, C_t]$ |

5. Insert $s_t(q_r, q_c)$ and Update R-Tape, C-tape
6. end if
7. else if $R_t = C_t$
8. Insert $s_t(q_r, q_c)$ and Update R-Tape, C-tape
9. end else if
10. else
11. end else

(2) Skyline maintenance algorithm for service failure

Before describing the Skyline maintenance algorithm for service failure, we first need to introduce the concept of the only dominant area. As shown in Figure 2, the services in the grey rectangle area are dominated only by service $s_2$ on the Skyline rather than any other services. We call it the only dominant area of service $s_2$. However, the services on the upper right two sides of the rectangle do not belong to the only dominant area of $s_2$, since they are dominated by service $s_1$ and service $s_3$, respectively.

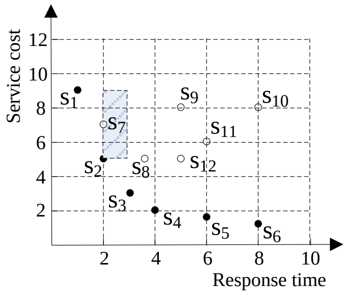



Figure 2. Example of the only dominant area

**Definition 5** (the only dominant area) If area $r$ is the only dominant area of service $s$, then all services in area $r$ are dominated by service $s$ only.

If the failure service $s$ is the dominated service, there is no effect on Skyline services. However, if the failure service is on Skyline, then we need to calculate the local-Skyline in the only dominant area of service $s$ and insert it into the original Skyline to form a new Skyline. Because services in the only dominant area of service $s$ are dominated by service $s$ only, when service $s$ loses efficacy, other services on the original Skyline cannot dominate these services. Therefore, the local-Skyline in the only dominant area becomes a part of the new Skyline and will be added to the original Skyline. Take Figure 2 for example. The local-Skyline computation process is as follows. Use the coordinates of $s_2$ for the lower left corner and the synthetic coordinates of $s_3$ abscissa and $s_1$ ordinate as the upper right corner to form the only dominant area of a rectangular. We can use the BBS (Branch and Bound Skyline) or BNL (Block Nested Loops) method to calculate the Skyline in this area, namely, the local-Skyline of service $s_2$. If the dynamic probability of service change is not high, we can take the local-Skyline computation process of every Skyline service as a pretreatment process. Algorithm 2 shows the pseudo code implementation for this process.

Algorithm 2. DSCA_DeleteService.

Input: R-Tape, C-Tape, $s_t(q_r, q_c)$

Output: R- Tape, C-Tape

1. $R_t \leftarrow$ Compare $q_r$ with values in R-Tape
2. $C_t \leftarrow$ Compare $q_c$ with values in C-Tape
3. if $R_t = C_t$
4. $\sigma \leftarrow$ Compute the local Skyline of $s_t$ 's only dominate region
5. Delete $s_t(q_r, q_c)$ and insert $\sigma$ into Skyline and Update R-Tape, C-tape
7. end if
8. if $R_t \neq C_t$
9. end if

### *2.4. Service selection based on mixed integer programming*

This process is the service execution phase. Service selection based on the global service QoS constraints chooses a service combination that has the maximum QoS utility function value and can meet the global service QoS constraints from all possible service portfolios. For example, if the global QoS constraints $CS = \{C_1, \ldots, C_m\}, 0 \le m \le r$ , we can calculate that the optimum service combination is $S = \{s_1, \ldots, s_n\}$ . Then, it must meet the following two conditions:

(1) The QoS utility function value $U(S)$ is the maximum in all service classes.

(2) The QoS aggregate value of the service combination $q(S) \le C, \forall C_m \in CS$ .

Since this study blocks services with large QoS uncertainty using the cloud model and eliminates redundant services using the Skyline computation, it reduces the search space for service selection and guarantees the security of services. Therefore, this paper only needs to make a service selection from the Skyline service to obtain the optimal combination of global service that can meet the users' global QoS constraints.

Mixed integer programming was used to solve the optimal combination service that satisfies above two conditions. Here, $x_{ji}$ denotes the corresponding binary variables of the candidate service $s_{ji}$ in service class $S_j$ . If $x_{ji} = 1$, then $s_{ji}$ was selected as the optimal component of the combination service. Otherwise $x_{ji} = 0$ and the service will be discarded. At this point, the method that can obtain the optimal combination service is the solution to the following multi-objective optimization problem:

A combination service utility function:

$$\sum_{j=1}^{n}\sum_{i=1}^{l} U(s_{ji}) \cdot x_{ji} \tag{1}$$

Global QoS constraints:

$$\begin{cases} \sum_{j=1}^{n}\sum_{i=1}^{l} q_m(s_{ji}) \cdot x_{ji} \le C_m, 1 \le m \le r \\ \sum_{j=1}^{l} x_{ji} = 1, \qquad 1 \le \mathrm{j} \le \mathrm{n} \end{cases} \tag{2}$$

In equation (2), $n$ is the number of service classes, $l$ is the number of candidate services in the service class, $r$ is the number of QoS attributes, $m$ is the number of QoS constrained attributes, and $C_m$ is the value of QoS constraints.

Therefore, by using mixed integer programming, we can quickly and reliably select the optimal global combination service that can satisfy users' QoS constraints from the Skyline service of each service class.

## 3. Comparison and analysis of algorithms

### *3.1. Test indicators*

To evaluate the SaaS services dynamic selection strategy based on the backward clouds and Skyline, this paper uses two evaluation indexes: execute success ratio and calculation time.

**Definition 6** Execute Success Ratio (ESR) includes the running $t$ times in a combination service. If the ratio of its QoS total utility function (overall utility, OU for short) and its actual operation monitoring results (monitoring result, MR for short) is not less than the set threshold, and the ratio of the user's QoS constraints attribute values and monitoring aggregate attribute values is not less than the percentage of the number of times $(h)$ of the set threshold and $t$ .

$$ESR = \frac{srn}{t} \times 100\%, srn = \sum_{i=1}^{100} \begin{cases} 1, \bigcap_{i=1}^{m} \frac{C_i}{U(S)} \ge H \cap \frac{OU}{MR} \ge h \\ 0, otherwise \end{cases} \tag{3}$$

From equation (3), the greater $ESR$ is, the higher its success rate is and the better the service selection method is. In this paper's experiment, $h = 0.9$ and $t = 200$.

**Definition 7** Calculation Time, (CT ) refers to the average computation time that the combination algorithm needs to generate the combination service. This indicator reflects the efficiency of the algorithm and indicates its capacity for running in a large-scale environment.

### *3.2. Experiments settings*

These experiments perform a quantitative comparison among SSBS, the classic typical Global optimal service selection method [5] (used in this paper) and the Hybrid optimization method proposed in recent years [7]. For the experimental dataset, this paper adopts two kinds of data collection, namely, public effective data collection QWS (Quality of Web Service) and synthetic datasets. The first dataset is a QWS real dataset, in which all data were collected from public services on the Internet. This dataset contains 2500 services and their 6 corresponding QoS attribute values: response time, throughput, usability, accessibility, interoperability and service costs. The second dataset is generated by the Skyline dataset generator, in which each service generates the same six QoS attributes as those from the QWS. It also contains three different types of two-dimensional QoS data types. The first is a correlation data collection, in which one dimensional attribute of the object is good when another dimension is also very good. The second is independent data collection, in which there is no connection between the two dimensional parameters and they are subject to independent, random distributions. The third is reverse correlation data collection, in which one dimensional attribute of the object is good, the other dimensional attribute is poor, and there is a relative balance between them. Figure 3 shows examples of the number of Skyline objects in these three different types of data collection. In this case, each type of data contains 35 SaaS services in which the black point indicates the Skyline service that dominates other services (white points). In the three different types of data collection, the Skyline objects in the correlation data collection have the smallest scale, as shown in Figure 3(a). The Skyline objects in the reverse correlation data collection have the largest scale, as shown in Figure 3(b). The Skyline objects in the independent data collection have a medium scale, as shown in Figure 3(c).

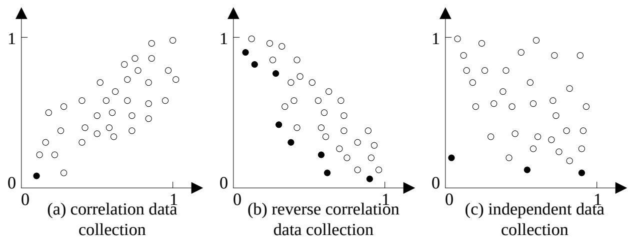



Figure 3. Example of the numbers of Skyline objects in three different types of data collection

### *3.3. Comparative experiments with other service selection algorithms*

Experiment 1: the comparison of execution success ratio

The purpose of experiment 1 is to verify the success ratio of service selection by the method proposed in this paper and present it in contrast with the Global and Hybrid methods. Figure 4 shows the experimental results of the success ratio in four types of QoS data using the SSBS. Figure 5 shows a comparison of the success ratio with the Global and Hybrid methods.

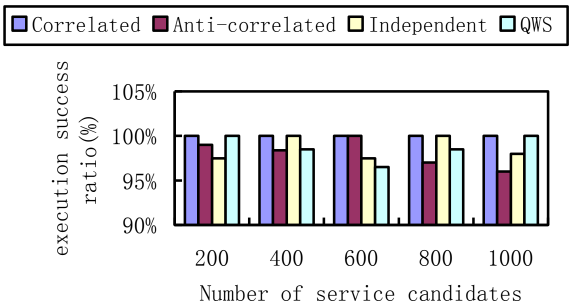
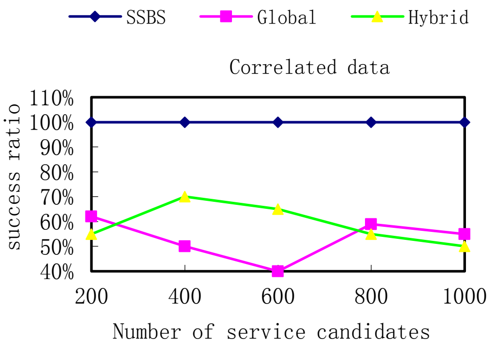
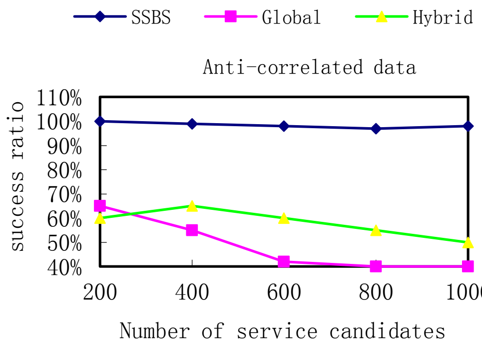



Figure 4. The success ratio of four types of QoS data using SSBS

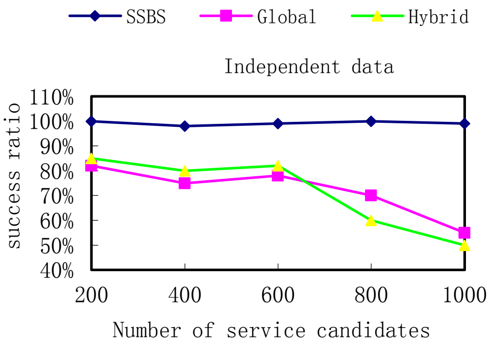
As Figure 4 shows, regardless of the number of candidate services and the SaaS service QoS, the success ratio of SSBS approaches 98.85%. This effectively overcomes problems such as the combination of poor quality, poor reliability, and service selection failure due to the uncertainty of QoS. In the experimental results shown in groups $4 \times 5$ of Figure 4, for the QoS associated data types, the service selection success ratios are 100%. Although the success ratios for the other QoS data types do not achieve 100%, their average never falls below 98%.



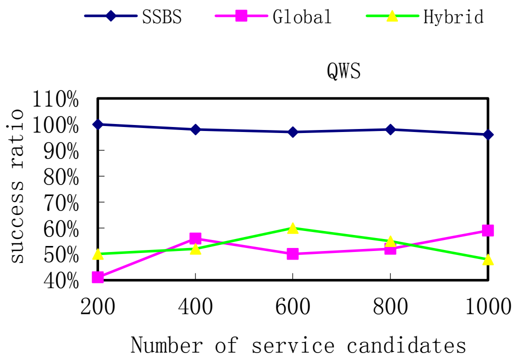

(a) correlation data




(b) reverse correlation data




(c) independent data




(d) QWS

Figure 5. Comparison of success ratio among SSBS, Global and Hybrid in four types of data collection

As shown in Figure 5, regardless of the number of candidate services and the SaaS service QoS, the success ratio of SSBS is considerably better than the Global and Hybrid methods. A comparison of the success ratio on QoS positive phase data is shown in Figure 5(a). The success ratio is 100% when using the method proposed in this paper, while the average success ratios of the Global and Hybrid methods are 53.2% and 59%, respectively. A comparison of the success ratio on QoS inverted phase data is shown in Figure 5(b). The success ratio is 98.4% using the method proposed in this paper, while the average success ratios of the Global and Hybrid methods are 48.4% and 58%, respectively. A comparison of the success ratio on QoS independent data is shown in Figure 5(c). The success ratio is 99.2% using the method proposed in this paper, while the average success ratios of the Global and Hybrid methods are 72% and 71.4%, respectively. A comparison of the success ratio on QWS data is shown in Figure 5(d). The success ratio is 97.8% using the method proposed in this paper, while the average success ratios of the Global and Hybrid methods are only 51.6% and 53%, respectively. Therefore, these results show that the success ratio of the proposed method is obviously higher than that of both the Global and Hybrid methods. The main reason is that, in the proposed approach, the cloud model is adopted to calculate the uncertainty problem of SaaS service QoS and the Skyline is used to calculate eliminate redundant services, thus reducing the search space for service selection methods and reducing the calculation time. This reduction greatly improved the execution success ratio of service selection.

Experiment 2: the comparison of calculation time

Experiment 2 verifies the time cost of service selection by SSBS and make a comparison with that of the Global and Hybrid methods. The experimental results on 4 different QoS data type by the proposed method are given in Figure 6. Figure 7 shows the comparison results with those of the Global and Hybrid methods.

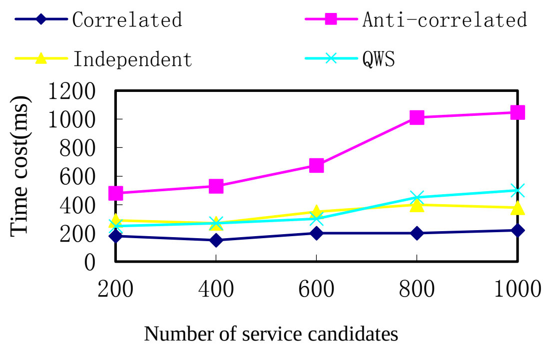
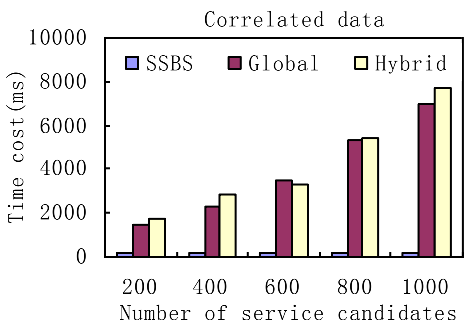
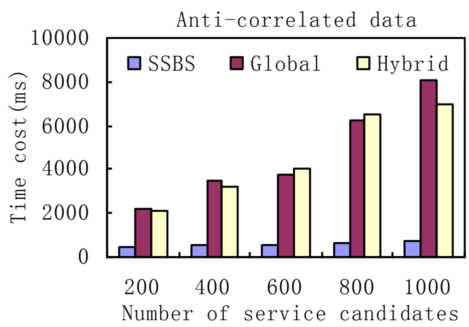



Figure 6. The time cost of four types of QoS data using the method proposed in this paper

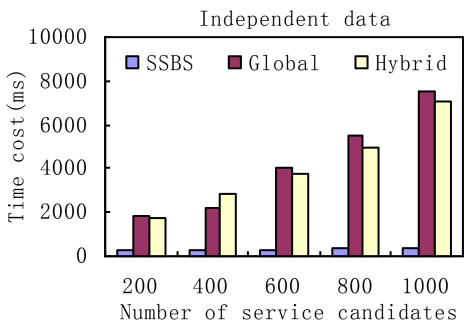
As shown in Figure 6, regardless of the number of candidate services, the time costs of SSBS are all quite low; their overall average is just 407.5 ms. SSBS can fully meet the user's time requirements in practical applications for SaaS service selection. Skyline service selection on the QoS correlation data requires the least time, averaging less than 200 ms, and the time spent by SSBS is relatively stable and little affected by the changing number of candidate services. Next, is the independent QoS data and the Skyline on the QWS data service selection, which consumes an average of less than 400 ms. Skyline service selection on the QoS reverse correlation data requires the most time. When the number of candidate services exceeds 800, the time cost is more than 1 s. However, the maximum time cost is only 1046 ms, and the average time cost is less than 800 ms; therefore, it will still be able to meet the user's time requirements for service selection.



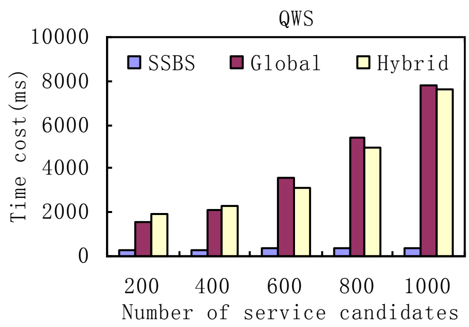

(a) correlation data




(b) reverse correlation data




(c) independent data




(d) QWS

Figure 7. Comparison on time cost among SSBS, Global, and Hybrid in four types of data collection

From Figure 7, we can see that regardless of the number of candidate services and the type of SaaS service QoS, the time cost of SSBS is much lower than that of the Global and Hybrid methods. A comparison of the time cost on QoS correlation data is shown in Figure 7(a). The time cost is 177.6 ms using the SSBS, while the time costs of the Global and the Hybrid methods are 3920 ms and 4180 ms, respectively—almost 20 times slower than the proposed method. A comparison of the time cost on QoS reverse correlation data is shown in Figure 7(b). The time cost is 590 ms using the method proposed in this paper, while the time costs of the Global and Hybrid methods are 4760 ms and 4560 ms, respectively—almost 7 times slower than the proposed method. A comparison of time cost on QoS independent data is shown in Figure 7(c). The time cost is only 319 ms using the method proposed in this paper, while the time costs of the

Global and Hybrid methods are 4200 ms and 4080 ms, respectively—almost 12 times slower than the proposed method. A comparison of time cost on QWS data is shown in Figure 7(d). The time cost is 338 ms using the method proposed in this paper, while the time costs of the Global and Hybrid methods are 4100 ms and 3980 ms, respectively—almost 11 times slower than the proposed method. Therefore, the comparison of time costs on the above four groups of data among the proposed method and the Global and Hybrid methods clearly shows that this paper's proposed method requires the least time compared with the Global and Hybrid methods. The main reason why the proposed method requires less selection time than the Global and Hybrid methods is that the proposed method adopts the Skyline computation for candidate services in each service class. Through the Skyline computation, it eliminates the redundant services in each service class and retains only the Skyline service. Moreover, the tape model can quickly locate the positional relationship between a service and Skyline and dynamically maintain Skyline services, thereby significantly reducing the time cost of the service selection process.

## 4. Conclusions

To efficiently select SaaS services that can meet user's non-functional requirements, we proposed a dynamic selection strategy of SaaS service based on Skyline. To address the uncertainty in SaaS service QoS data, the strategy adopts three digital features, including expectations, entropy and hyper entropy, in the backward cloud algorithm for qualitative transformation. To perform matching selection of all services in the SaaS services library, redundancy elimination algorithm based on Skyline computation is proposed. Based on the dynamic characteristics of SaaS services, we propose a dynamic maintenance algorithm for Skyline service. Finally, the results of simulation experiments show that SSBS is well able to address the uncertainty of the QoS problem, reduce the search space for service selection methods, reduce the calculation time and provide a service selection scheme that effectively satisfies the constraints of users' global QoS requests.

## 5. Acknowledgements

This work was partly financially supported through grants from The Open Issues of Beijing city key Laboratory (No.BKBD-2017KF08). The authors thank the 3 anonymous reviewers for their helpful suggestions.

## References

1. Ardagna Danilo, and R. Mirandola, "Per-flow Optimal Service selection for Web services based processes," *Journal of Systems and Software*, vol. 83, no. 8, pp. 1512-1523, August 2010.
2. MLAAnselmi Jonatha, D. Ardagna, and M. Passacantando, "Generalized Nash Equilibria for SaaS/PaaS Clouds," *European Journal of Operational Research*, vol. 236, no. 1, pp. 326-339, July 2014.
3. Guosheng Kang, Jianxun Liu, and Mingdong Tang, "An Effective Dynamic Web Service Selection Strategy with Global Optimal QoS Based on Particle Swarm Optimization Algorithm," in *Proceedings of the International Parallel and Distributed Processing Symposium Workshops & Phd Forum*, pp. 2280-2285, Washington, USA, May 2012.
4. 14. Xiaoyong Li, Yijie Wang, and Xiaoling Li, "Parallel Skyline Queries over Uncertain Data Streams in Cloud Computing Environments," *International Journal of Web & Grid Services*, vol. 10, no. 1, pp.24-53, January 2014.
5. Wenfeng Li, Ye Zhong, and Xun Wang, "Resource Virtualization and Service Selection in Cloud Logistics," *Journal of Network & Computer Applications*, vol. 36, no. 4, pp. 1696-1704, November 2013.
6. Cusumano Michael, "Cloud Computing and SaaS as New Computing Platforms," *Communications of the Acm*, vol. 53, no. 4, pp. 27-29, April 2010.
7. Alrifai Mohammad, T. Risse, "Combining Global Optimization with Local Selection for Efficient QoS-aware Service Composition," in *Proceedings of the 18th international conference on World wide web*, pp. 881-890, New York, USA, April 2009.
8. Alrifai Mohammad, D. Skoutas, and T. Risse, "Selecting Skyline Services for QoS-based Web Service Composition," in *Proceedings of the 19th international conference on world wide web*, pp. 11-20, New York, USA, April 2010.
9. Lianyong Qi, Ying Tang, and Wanchun Dou, "Combining Local Optimization and Enumeration for QoS-aware Web Service Composition," in *Proceedings of the IEEE International Conference on Web Services*, pp.34-41, July 2010.
10. Wanbin Son, Seung-won Hwang, and Hee-Kap Ahn, "MSSQ: Manhattan Spatial Skyline Queries," *Information Systems*, vol. 40, no.1, pp. 313-329, August 2011.
11. George Trimponias, Ilaria Bartolini, and Dimitris Papadias, "Skyline Processing on Distributed Vertical Decompositions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pp. 850-862, April 2013.
12. Yong Wang, Guiping Dai, and Yarong Hou, "Dynamic Methods of Trust-aware Composite Service Selection," *Chinese Journal of Computers*, vol. 32, no. 8, pp. 1668-1673, August 2009.