ARTICLE IN PRESS

Future Generation Computer Systems ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Efficient and reliable service selection for heterogeneous distributed software systems

Shangguang Wang [a], Lin Huang [a], Lei Sun [a], Ching-Hsien Hsu [b,c,*], Fangchun Yang [a]

[a] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[b] Department of Computer Science and Information Engineering, Chung Hua University, Hsinchu, Taiwan
[c] Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, 300191, Tianjin, China

HIGHLIGHTS

- Design a service composition middleware for heterogeneous distributed systems.
- An efficient and reliable service selection approach based on information and variance theory is proposed.
- Experiments with real-world dataset show that the proposed technique is superior to other existing approaches.

ARTICLE INFO

Article history:
Received 12 September 2015
Received in revised form
16 November 2015
Accepted 14 December 2015
Available online xxxx

Keywords:
Service selection
Service composition
QoS uncertainty
Entropy
Variance

ABSTRACT

The service-oriented paradigm is emerging as a new approach to heterogeneous distributed software systems composed of services accessed locally or remotely by middleware technology. How to select the optimal composited service from a set of functionally equivalent services with different quality of service (QoS) attributes has become an active focus of research in the service community. However, existing middleware solutions or approaches are inefficient as they search all solution spaces. More importantly, they inherently neglect QoS uncertainty owing to the dynamic network environment. In this paper, based on a service composition middleware framework, we propose an efficient and reliable service selection approach that attempts to select the best reliable composited service by filtering low-reliability services through the computation of QoS uncertainty. The approach first employs information theory and probability theory to abandon high-QoS-uncertainty services and downsize the solution space. A reliability fitness function is then designed to select the best reliable service for composited services. We experimented with real-world and synthetic datasets and compared our approach with other approaches. Our results show that our approach is not only fast, but also finds more reliable composited services.

1. Introduction

Services are commonly regarded as black boxes with well-defined interfaces that can be aggregated recursively into new services by *service composition* technology [1]. An important aspect of service composition is the finding and binding of services in order to compose them into a composite application. Service composition has become the kernel technology in the domain of service-oriented architecture (SOA) which is able to meet the business requirements of heterogeneous distributed software systems.

According to the SOA paradigm, composite applications are specified as abstract processes composed of a set of abstract services (called the *service class*). Then, at the service's run time, for each service class a concrete service (called the *service candidate*) is selected and invoked. This case ensures loose coupling and design flexibility for many business applications distributed within and across organizational boundaries [2].

It is well-known that QoS (e.g., response time, reliability, and throughput) plays an important role in determining the performance of selected services for service composition middleware [3]. Traditional service discovering and matching approaches (e.g., UDDI, Bluetooth, etc.) only focus on searching services with functionalities. However, with dramatic growth in the number of

* Corresponding author at: Department of Computer Science and Information Engineering, Chung Hua University, Hsinchu, Taiwan.
E-mail addresses: sgwang@bupt.edu.cn (S. Wang), huanglin@bupt.edu.cn (L. Huang), sunlei@bupt.edu.cn (L. Sun), chh@chu.edu.tw (C.-H. Hsu), fcyang@bupt.edu.cn (F. Yang).

http://dx.doi.org/10.1016/j.future.2015.12.013
0167-739X/

Please cite this article in press as: S. Wang, et al., Efficient and reliable service selection for heterogeneous distributed software systems, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2015.12.013

services, there are typically many services that are functionally equivalent for the user, leading to the user not knowing which service should be selected. To satisfy users' QoS requirements, concrete services have to be selected and instantiated for each business process's abstract task. Giving the growing number of service candidates of each service class that can offer different QoS values, the selection of the optimal combination of services that fulfills users QoS constraints becomes a very complex and time consuming task [4]. Hence, a QoS-based service selection approach is proposed, aiming at finding the best combination of services that satisfy a set of end-to-end QoS constraints from the user's requests.

Some notable service selection approaches include Hybrid [5], GA [6], Replanning [7], CAR & AR [8], MIP [9], and Heuristic [10]. Although these approaches have been shown to perform well in their respective contexts, they are not suited to composited services because of violent QoS fluctuation in services (i.e., the response time of the service changes over time) [8,11,12]. They lack consideration of QoS uncertainty, so they cannot provide reliable services for users in composition systems owing to the dynamic service environments involved. Generally, service candidates participating in service selection are widely distributed in the network. These services come from different organizations/systems and run on different platforms. Hence, any slight changes in location, network environment, service requirement time, and other aspects affect the reliability of these service candidates [13]. Therefore, it is worth noting that a service with consistently good QoS is typically more reliable than a service with a large variance in QoS. Therefore, consistency should be considered as an important criterion for reliable service selection.

In addition, there is a further question we must face: are there high-overhead services with the same functional attributes but different QoS? The statistics published by the web services search engine Seekda! indicate that the number of web services increased exponentially in recent years. Before cloud computing, many researchers asked whether web services can be used as service candidates for each service class. Some researchers were pessimistic on this point. Now, however, the pay-per-use business model promoted by the cloud computing paradigm may enable service providers to offer massive services (e.g., infrastructure as a service, platform as a service, and software as a service) to public, private, or hybrid cloud platforms [14]. Hence, in one vision for the future, there will be massive services. However, most existing approaches suffer from a concentrated workload with increasing number of services, causing poor real-time performance. The main reason for this is that they focused too much on optimization of selection approaches to reduce time costs within the service selection process. They neglected a basic principle: reducing the search space for service candidates (called the *solution space*) is more important than focusing only on the seeking or optimization of service selection approaches.

Different from most existing approaches, we propose an efficient and reliable approach to consider not only the QoS uncertainty of services, but also to pay close attention to downsizing solution spaces for the service selection process. The QoS uncertainty is used to filter low-reliability services by information theory and probability theory. The higher the QoS uncertainty of a service, the lower the reliability of the service; if unacceptable, it must be filtered from among the service candidates. Why do we use information theory and probability theory in this paper? Entropy is used to measure the expectations of a random variable and its numerical value can reflect very well the degree of a service's disorder. Furthermore, the main role of the variance is as a measure of the stability of a sample. Using these two aspects to prune low-reliability services could help make up for defects in existing service selection approaches. Compared with previous QoS-based service selection approaches, our main contributions can be summarized as follows.

Middleware Framework: Aimed at efficient and reliable service selection in heterogeneous distributed software systems, a service composition framework is presented with three distinct components, i.e., *Discovery Engine*, *Selection Engine*, and *Composition Engine*.

High Reliability. We adopt entropy and variance to compute QoS uncertainty. Low-reliability services are then pruned, and high-reliability services can be selected by our designed reliability fitness function for composited services.

Low Computation Time. Because many low-reliability services are filtered, the solution space of service selection is downsized sharply. This yields lower computation time in the service selection process than existing techniques.

Extensive Experiments. We implemented our approach and experimented with 5825 real-world services and 10,000 synthetic services. Our results show that our approach is superior to others. We also report results on a parameter study of our approach.

The remainder of this paper is organized as follows. In Section 2, we introduce the background of service selection, including related definitions and related work. Section 3 introduces the proposed service composition framework. Section 4 describes our approach in detail, including computing QoS uncertainty, filtering uncertain services, and the service selection process. The evaluation in Section 5 demonstrates the benefits of our approach. Finally, Section 6 concludes the paper.

2. Background

2.1. Related concepts

In this section, we explain some concepts related to service selection and service composition. The purpose of a composition service is to achieve a particular function that can satisfy the user's requirements and preferences. It is obtained by combining a plurality of service candidates which are selected from each service class (which consist of a number of service candidates). We can understand the concepts of a composite service thoroughly through the following example. In a composite service $S = \{s_1, s_2, \ldots, s_n\}$, any $s_i \in S$ and $s_i = \{s_{i1}, s_{i2}, \ldots, s_{il}\}$ refers to a service class and contains l ($l > 1$) functionally equivalent service candidates with different QoS values. The QoS values for a specific web service are $\{q_1, q_2, \ldots, q_l\}$.

The QoS affects the performance of a web service and is a nonfunctional attribute of the web service. A service's QoS has many attributes such as response time, reliability, throughput, delay, availability, and so on. Generally, QoS attributes can be divided into two categories: positive and negative. Positive QoS attributes (e.g., *reliability, availability*, etc.) means that the larger the attribute value, the better the quality of the web service. Conversely, negative QoS attributes (e.g., *response time, delay*, etc.) should be kept as low as possible. In this paper, we consider both positive and negative QoS attributes.

Generally, a service's QoS contains multiple attributes. We could obtain the corresponding attribute value through quantitative calculation. For example, the service s_{ij} has r attributes and its attribute vector can be expressed as $Qs_{ij} = \{q_1(s_{ij}), q_2(s_{ij}), \ldots, q_r(s_{ij})\}$, where the value of $q_k(s_{ij})$ $(1 < k < r)$ represents the kth attribute value in service s_{ij}. Similarly, the composite service's attribute vector can be expressed as $QS = \{q_1(S), q_2(S), \ldots, q_r(S)\}$, where the value of $q_k(S)$ is aggregated by the kth attribute values from all the selected service candidates. Table 1 lists the QoS aggregation functions of the sequential composition model. Other models (e.g., parallel, conditional, and loop) can be transformed into the sequential model using techniques described in existing papers [15].

Please cite this article in press as: S. Wang, et al., Efficient and reliable service selection for heterogeneous distributed software systems, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2015.12.013