Information Sciences 433–434 (2018) 255–273

Contents lists available at ScienceDirect

Information Sciences

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

ISAT: An intelligent Web service selection approach for improving reliability via two-phase decisions

Weidong Wang[a,b,*], Zhangqin Huang[a,b,*], Liqiang Wang[c]

[a] Faculty of Information Technology, Beijing University of Technology, Beijing, China
[b] Beijing Engineering Research Center for IoT Software and Systems, Beijing University of Technology, Beijing, China
[c] Dept. of Computer Science, University of Central Florida, Orlando, FL, USA

A R T I C L E   I N F O

Article history:
Received 3 March 2017
Revised 17 December 2017
Accepted 25 December 2017
Available online 26 December 2017

Keywords:
Web service
Optimization selection
Reliability
Decision approach

A B S T R A C T

Due to stochasticity and uncertainty of malicious Web services over the Internet, it becomes difficult to select reliable services while meeting non-functional requirements in service-oriented systems. To avoid the unreliable real-world process of obtaining services, this paper proposes a novel service selection approach via two-phase decisions for enhancing the reliability of service-oriented systems. In the first-phase decision, we define the problem of finding reliable service candidates as a multiple criteria decision making (MCDM) problem. Then, we construct a decision model to address the problem. In the second-phase decision, we define the problem of selecting services based on non-functional requirements as an optimization problem. Finally, we propose a convex hull based approach for solving the optimization problem. Large-scale and real-world experiments are conducted to show the advantages of the proposed approach. The evaluation results confirm that our approach achieves higher success rate and less computation time to guarantee the reliability when compared to the other state-of-the-art approaches.

1. Introduction

Service- oriented architecture (SOA) patterns provide a flexible support for building software applications that use Web services available in typically large-scale and complex networks [32]. Web services are self-contained and loosely coupled reusable software components distributed and invoked over the Internet [31]. Compared with traditional stand-alone environments, the stochastic and unpredictable nature of open distributed environments based on SOA introduces new challenges in improving service-oriented software reliability [11]. Actually, the challenges are twofold. First, it is difficult to build fully reliable or fault-free service-based softwares under limited development cost and the pressure of time to market [28]. Then, it is uncertain whether users previously know which Web services are malicious. This is because the internal designs and implementation details of remote Web services from the third party are unclear to some extent [47].

In service-oriented software reliability researches [21], there are four possible approaches for improving reliability, which are fault prevention [36], fault removal [45], fault prediction [40], and fault tolerance [12]. Fault prevention and fault removal approaches need to revise the source code of Web services. They may be confined on the limited development cost and the pressure of time to market. Fault prediction approaches may result in inaccurate prediction because of the unpredictability of open distributed environments. Instead of removing faults or predicting faults, fault tolerance approaches

* Corresponding authors.
E-mail addresses: wangweidong@bjut.edu.cn, weidong_bjtu@126.com (W. Wang), zhuang@bjut.edu.cn (Z. Huang), lwang@cs.ucf.edu (L. Wang).

https://doi.org/10.1016/j.ins.2017.12.048
0020-0255/© 2017 Elsevier Inc. All rights reserved.

have been widely applied in building reliable service-oriented softwares. Such approaches can be divided into three categories based on the roles in the SOA, *i.e.,* provider-, registry- and requester-based approaches, respectively. Provider-based approaches such as FT-Web [17] and FT-CORBA [15] mainly focus on developing functionally equivalent components running in parallel or operating another alternative backup component previously designed when the primary component crashes. Registry-based approaches such as an active UDDI (Universal Description, Discovery and Integration) [33] apply continuous access and discovery strategies to guarantee that users can obtain the available services needed. Requester-based approaches employ complex mediation strategies such as ws-reliability [24] to ensure the service to be successfully invoked by other services when their interface mismatches. Owing to the cost and time of developing redundant components in the provider-based approaches or designing complex fault tolerance strategies in registry- and requester-based approaches, the above three fault tolerance approaches are usually only used for critical softwares.

However, one of the promising fault tolerance approaches without paying much money and time for building reliable service-oriented softwares, commonly known as design diversity, is to adopt functionally equivalent although independently designed service candidates. The reason is that there are a large number of service candidates with equivalent function yet different QoS (Quality of Service) implemented by different organizations over the Internet and these service candidates can be employed as alternative components for tolerating faults. Complementary to previous fault tolerance approaches, which mainly focus on developing redundant components or designing complex fault tolerance strategies, this paper investigates how to optimally select fault tolerance components (services) to build reliable service-oriented softwares for the final goal of improving reliability.

Actually, the limitations of current service selection approaches are, they usually cannot find a good balance between speed and reliability: either fast and unreliable or slow and reliable. This is because, in an open distributed and service-oriented environment, there exist some unreliable Web services over the Internet that may be poor-quality, expensive, time-consuming, or even malicious. Furthermore, if these unreliable Web services cannot be well filtered, any effective service selection approach will become invalid since these unreliable Web services may result in QoS downgrade, or even lead to an ineffective selection process. Therefore, a filter for removing unreliable services is indispensable in supporting the selection process. In addition, as the number of service candidates increases, it may forbiddingly take large amount of time to obtain the optimal selection result according to the given QoS requirements proposed by users.

To address the issues discussed above, we present an intelligent Web service selection approach for improving reliability via two-phase decisions, namely ISAT. The ISAT considers not only how to avoid the selection of malicious services, but also how to maximize the QoS performance of Web services. Different from the previous version [38], the definition is extended to any number of non-functional constraints including linear and non-linear constraints. Also unlike the definition in [47], we typically consider the utility of a service as the only parameter of objective function since the other specific parameters, such as interest and error rate, are introduced in the first-phase decision process. Next, we summarize the major research contributions of this paper as follows.

- In the first-phase, we define the problem of identifying reliable service candidates as a multiple criteria decision making (MCDM) problem. Then, we propose the first-phase decision based on the technique for order of preference by similarity to ideal solution (TOPSIS) [10] to address this problem. Compared with traditional decision approaches, our customized decision typically considers multiple decision parameters from different dimensions (*e.g.,* user-, condition-, and environment-specific parameters) according to the characteristics of service-oriented systems.
- In the second-phase, we define the problem of selecting services for an optimal execution plan as the specific 0–1 integer programming problem. Then, we propose a convex hull based approach to solve efficiently this problem. Unlike traditional decision approaches, the convex hull based decision can reduce the search space of service candidates to the minimum while ensuring the success rate and accuracy of the solution.

Comprehensive experiments are conducted to study the success rate, computation time, and approximation ratio of our proposed approach compared with other competing approaches. The experimental results show the high success rate and efficiency of our approach. The rest of this paper is organized as follows. In Section 2, we introduce a motivating example to illustrate the research problem of the paper. In Section 3, we present the details of the two-phase decisions for Web service selection including the framework and QoS model of Web services. In Section 4, we give an illustrative example to show the computation process of the proposed approach. Section 5 discusses the experiment results. Section 6 reviews the state-of-the-art relevant to this work. Finally, Section 7 concludes this paper and outlines the future work.

2. Motivating example

We begin by a motivating example to illustrate the research problem. In this paper, an execution plan is an abstract description for the activities in a commodity trade process, which includes a group of tasks to execute according to a certain workflow. Fig. 1 indicates that the execution plan has six tasks and each task can execute by invoking a concrete component service. In the example, we assume that there are multiple functionally equivalent component Web service candidates provided by service communities. Meanwhile, we can use universal technologies to ensure the consistency of programming interface.

As the example in Fig. 1, there are several challenges to be addressed. (1) There are so many component Web services for the Task 1. We use Task 1 to make an order for the purpose of trade. Thus we need to identify which service candidates are