Future Generation Computer Systems 25 (2009) 199–212

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Workflow-based resource allocation to optimize overall performance of composite services☆

BangYu Wu, Chi-Hung Chi *, Zhe Chen, Ming Gu, JiaGuang Sun

School of Software, Tsinghua University, Beijing, 100084, China
Key Laboratory for Information System Security, Ministry of Education of China, Beijing, 100084, China

ARTICLE INFO

Article history:
Received 28 January 2008
Received in revised form
2 June 2008
Accepted 4 June 2008
Available online 25 June 2008

Keywords:
Service oriented architecture
Performance
Composite service
Resource allocation

ABSTRACT

In software service provision, the overall performance of a composite service is often the ultimate focus of concern rather than those of its individual components. This opens new opportunities for resource allocation because with its service workflow definition, more accurate prediction of its individual components' dynamic workload is possible, thus resulting in better utilization of resources. In this paper, we propose to improve resource allocation through tracing and prediction of workload dynamics of component services as requests traverse and pipeline through the workflow. Factors affecting service workload such as service time, transition probability, replication overhead for additional service etc. as well as the uncertainty in request arrival time are all taken into consideration in our model. The goal is to maximize the number of requests completed under the constraints of limited available resources. Experimental study on TPC-W and synthetic workflow shows that our dynamic workflow-based resource allocation scheme is much more efficient in enhancing the overall performance of composite services than current resource allocation schemes do.

1. Introduction

Composition is one key service design principle under SOA (Service Oriented Architecture). With composition, new complicated composite services can be formed by aggregating component services together in a workflow. Due to its importance, numerous research efforts have been spent on effective service matching and selection, trying to find the "best" component services available in the registry for composition [1–6,25,28]. With the argument that "service without quality is not a true service", increasing attention is now being given to the non-functional QoS aspects of services. One big challenge now is on how to allocate minimum resource to individual services within a workflow so that the overall agreed (or pre-defined) service quality can be provided at runtime. Consider the situation in Fig. 1, in which the workflow for a given composite service is defined, achieving or failing the pre-defined agreement of the overall service will result in different business values.

The goal of resource allocation for services is for all services in the workflow to meet their respective performance targets. However, due to the dynamic nature of service invocation, resource allocation and QoS provisioning of services are much more difficult than that of traditional software/servers environment. This dynamism comes from three main sources: (i) non-deterministic arrival of service requests, (ii) non-constant demand for resource capacity of requests (due to different inputs and service level agreements (SLAs)), and (iii) execution uncertainty within a service workflow (e.g. parallel exclusive operations defined in Section 3), which results in only some component services inside to be used. All these dynamism result in the need for runtime adjustment of resources allocated to individual component services.

To address this problem, previous researches on quality provisioning for services often perform static allocation first, and then use replication technologies [7–12] to adjust the instant resource capacity of individual component services at runtime. While they lay down important foundations for capacity planning in the traditional provider-centric computing environment, most of them have not taken full advantage of information about the workflow structure and the SLA specifications in service quality provisioning. For examples, SLAs of some services might tolerate certain percentages of request failure without incurring penalty. Furthermore, clients often only care about the overall QoS of the requested composite service and not those of its individual components. If we only enhance the performance of one hotspot component service, the performance bottleneck might be shifted to subsequent component services in the same workflow, thus resulting in little (if any) improvement in the overall service

☆ A preliminary version of this paper is published in the 2007 IEEE Service Computing Conference. Substantial modification of the content, including the algorithm and experimental results, is made.

* Corresponding author. Tel.: +86 10 62773368.
E-mail addresses: wby03@mails.tsinghua.edu.cn (B. Wu), chichihung@mail.tsinghua.edu.cn (C.-H. Chi).

0167-739X/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2008.06.003

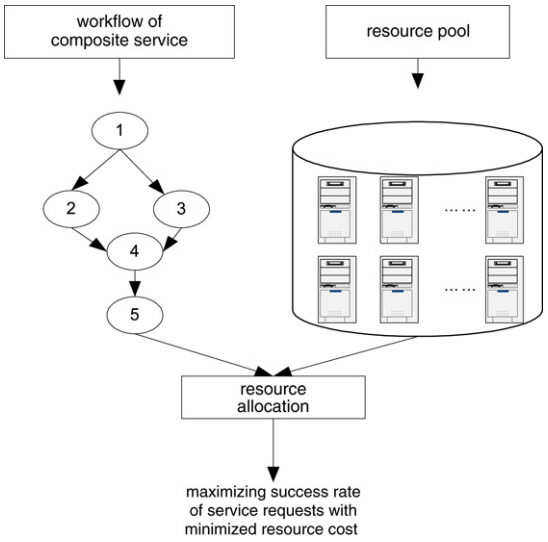

Fig. 1. An example of resource allocation for composite service.

quality. Also, in the decision making of resource allocation, not many schemes take the code/data replication and setup times for services into consideration. In the real time environment of service provisioning, such overhead will be very important in determining the effectiveness of the schemes.

In this paper, we propose to improve dynamic resource allocation for software service through tracing and prediction of workload dynamics of its component services as requests traverse and pipeline through the workflow. The main goal here is to improve the accuracy of future workload prediction of individual component services of a workflow dynamically, and do the proper adjustment of resource allocation based on this prediction in advance of the actual workload arrival. Factors affecting service workload such as service time, transition probability, replication overhead for service etc. as well as the uncertainty in request arrival time are all taken into consideration in our model. The performance matrix used here is to maximize the number of requests completed under the constraints of limited available resources. Experimental study on TPCW and synthetic workflow shows that our dynamic workflow-based resource allocation scheme is much more efficient in enhancing the overall performance of composite services than current resource allocation schemes do.

The outline for the rest of this paper is as follows. Section 2 gives related previous work on service composition and QoS provisioning. Section 3 explains the basic workflow model and the assumption we make in this paper. In Section 4, we propose a dynamic greedy-based resource allocation scheme for composite service based on the transition probability and arrival time of requests workload. In order to compare our dynamic resource allocation scheme with static resource allocation ones, we present one static allocation algorithm in Section 5. Simulation results are given in Section 6. Results show that our allocation scheme is much more flexible and effective to deal with dynamic request arrival and to maximize request success percentage. Finally, the paper concludes in Section 7.

2. Related work

Lots of research efforts in SOA have been spent on the topic of service composition. Some industrial standard specifications are also proposed. For example, SOAP, WSDL, and UDDI [13] aim at providing infrastructure to support Web service composition. BPEL4WS (Business Process Execution Language for Web services) [14] combines Microsoft's XLANG [15] and IBM's WSFL (Web service Flow Language) [16] to provide language support for the formal specification of business processes and business interaction protocols. WSCI (Web service Choreography Interface) [17] is an XML-based interface description language that describes the flow of messages exchanged by Web services. BPML (Business Process Modeling Language) [18] is designed to express abstract and executable processes that address all aspects of enterprise business processes. Other proposed notations for service description and composition include ebXML [19] and DAML-S [20].

From the viewpoint of actual service composition, there are works on the service modeling and framework. SAHARA [21] proposes two different composition models, the cooperative one and the brokered one. SWORD [22] provides a simple and efficient scheme for Web service composition. FUSION [23] describes a framework for dynamic Web service composition and automatic execution. The SELF-SERV architecture features a service manager and a pool of services [26]. While these efforts often focus on the expansion of service function capacity, new initiatives also start to investigate service selection based on non-functional QoS of services. [24] addresses the topic of dynamic QoS aware service composition. However, the underlying workflow model does not support parallelism nor branching. It just defines workflow as a sequential chain of service operations. WebQ [25] proposes an adaptive framework to maintain QoS by dynamic service binding. [5] gives a quality driven approach to select component services during the execution of a composite service. [29] adds QoS classes to service selection.

Traditional resource allocation solves the problem of limited resource availability when n processes try to access m resources. Some processes might be able to access resource concurrently, while other processes might require exclusive or sequential access to the resource [32]. In distributed systems, how to allocate m resources to n tasks determines the system efficiency, throughput [33,34], or resource utilization [36]. However, their workloads are relatively fixed and static, the resource allocating methods are static as well, and they are not customized to workflow with multiple component services.

With the increasing importance of service computing [35], resource allocation for services that consider the dependency relationships among component services have been proposed. [37] presents a cluster-based decentralized resource allocation policy in mini-grid by computing the dependency relationships of tasks in a given real time DAG (direct acyclic graph) dynamically. However, it deals with the scheduling of relatively fine-grained execution entities such as processes using fork. Uncertainties in either the workload or the execution time is not considered. [38] performs the optimization of execution time of service workflows through grouping services and scheduling them onto the grid infrastructure. [39] presents a novel algorithm that only staticlly maps workflow processes to existing Grid services. [40] considers both the workflow and its performance QoS, but it focuses more on scientific applications, not general Web services.

Our work is related to prior efforts in Web service composition. We focus on improving the overall QoS of composite service workflow after all the individual component services are selected. Unlike the previous work on dynamic service selection, our target is to construct a stable composite service through dynamic resource allocation and service replication. Hence, our work is actually complementary to most of the existing work mentioned above.

3. Workflow model for composite service

In this section, we first give the assumptions that we make in this paper, followed by the basic workflow model for composite service that we use.