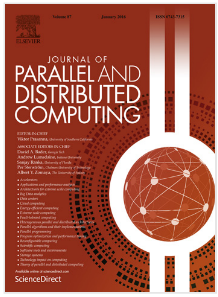
Accepted Manuscript



Combining quantitative constraints with qualitative preferences for effective non-functional Properties-aware Service Composition

Hongbing Wang, Peisheng Ma, Qi Yu, Danrong Yang, Jiajie Li, Huanhuan Fei

PII: S0743-7315(16)30141-1
DOI: http://dx.doi.org/10.1016/j.jpdc.2016.10.013
Reference: YJPDC 3551

To appear in: *J. Parallel Distrib. Comput.*

Received date: 15 March 2015
Revised date: 29 August 2016
Accepted date: 13 October 2016

Combining Quantitative Constraints with Qualitative Preferences for Effective Non-functional Properties-aware Service Composition

Hongbing Wang* Peisheng Ma* Qi Yu† Danrong Yang* Jiajie Li* Huanhuan Fei*
*School of Computer Science and Engineering, Southeast University
Nanjing, China, 211189
hbw@seu.edu.cn, psm@seu.edu.cn
†College of computing and Information Science,
Rochester Institute of Tech, USA
qi.yu@rit.edu

***Abstract*—With the increasing popularity of the service-oriented architecture and web service technologies, service composition has become widely adopted to create value-added services from existing ones. As more web services have been deployed on the Internet, it results in a large number of services providing identical functionalities while differing in their non-functional properties (NFPs). However, most of the existing techniques for NFP-aware service composition only consider quantitative NFPs. In this paper, we present a model that deals with both quantitative and qualitative NFPs. We develop two algorithms, where the first one combines global optimization with local selection and the second one leverages a genetic algorithm. We have conducted extensive experiments to evaluate the effectiveness of our model and algorithms.**

***Keywords*-service composition; non-functional properties; quantitative and qualitative; TCP-nets;**

I. Introduction

Thanks to the wide adoption of the Service Oriented Architecture (SOA), service providers can conveniently describe and deploy their web services to the Internet by using the existing service standards, including SOAP, WSDL and UDDI. Meanwhile, service consumers can easily bind to and invoke any of these services. However, selecting appropriate services from a service repository to fulfill a given user's requirement may still involve tedious manual work that requires the intervention from software developers or service integrators. With the rapidly increasing number of web services on the Internet, it is possible that many of these services provide similar or identical functionalities while differing in their NFPs (non-functional properties) or QoS (quality of service), such as response time, price, availability, throughput, platform, security, SLA level, and so on. *Service Class* is used to denote the collection of services which provide the same functionalities. NFPs of web services has become crucial for providers to meet service users' nonfunctional requirements.

Although the number of web services available on the Internet is very large, it is still possible that no single service can satisfy the user's functional requirements. In order to fulfill the user's request, service composition has been widely used, which leverages functionalities provided by individual component web services to offer a value-added service. The component service may be an atomic service or another composed service. A composed service can be described in a workflow-like language (e.g. BPEL[1]). This trend has triggered considerable research effort on automatic ways of composing web services, both in academia and industry.

The goal of NFP-aware web service composition is to create the best composite service with respect to users' preferences while satisfying their constraints with respect to NFPs. When a user request cannot be satisfied by any single service, the system should create and execute a composite service for it and return the results to the user. The number of service classes involved in this composite service may be large and the size of each service class from which these component services are selected is likely to be even larger. Different users may have different constraints and preferences about NFPs. For example, a search service user may want to optimize the throughput as much as possible, while a real-time application user may want to minimize the response time. The general process of NFP-aware web service composition can be illustrated by Figure 1. When a user request arrives, the system will generate an abstract execution plan (a.k.a., abstract workflow) of the composite service which satisfies the requirements of the request. The execution plan consists of several service classes, each of which corresponds to a set of candidate concrete services. When each service class is replaced by a concrete service, the abstract execution plan is transformed into an *executable business process*. As illustrated in Figure 1, an abstract execution plan ($service\ class\ 1$, $service\ class\ 2$, $service\ class\ 3$) is transformed to the executable business process ($service_{13}$, $service_{22}$, $service_{31}$) by assigning a concrete service of the corresponding service class to each node of the abstract workflow. The task of NFP-aware service composition is to generate the most desirable executable business process while satisfying users' constraints and