Expert Systems with Applications 42 (2015) 135–145

Contents lists available at ScienceDirect

Expert Systems with Applications

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

# Imperialist competitive algorithm with PROCLUS classifier for service time optimization in cloud computing service composition

Amin Jula [a,*], Zalinda Othman [a], Elankovan Sundararajan [b]

[a] Data Mining and Optimization Research Group, Centre for Artificial Intelligence, UKM Bangi, 43600 Selangor, Malaysia
[b] Centre of Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, UKM Bangi, 43600 Selangor, Malaysia

ARTICLE INFO

*Article history:*
Available online 4 August 2014

*Keywords:*
Cloud computing
Service composition
Service selection
Service time
Quality of service
QoS
Imperialist competition algorithm
Clustering
Proclus

ABSTRACT

Aiming to provide satisfying and value-added cloud composite services, suppliers put great effort into providing a large number of service providers. This goal, achieved by providing the "best" solutions, will not be guaranteed unless an efficient *composite service composer* is employed to choose an optimal set of required unique services (with respect to user-defined values for quality of service parameters) from the large number of provided services in the pool. Facing a wide service pool, user constraints, and a large number of required unique services in each request, the composer must solve an NP-hard problem. In this paper, *CSSICA* is proposed to make advances toward the lowest possible service time of composite service; in this approach, the PROCLUS classifier is used to divide cloud service providers into three categories based on total service time and assign a probability to each provider. An improved imperialist competitive algorithm is then employed to select more suitable service providers for the required unique services. Using a large real dataset, experimental and statistical studies are conducted to demonstrate that the use of clustering improved the results compared to other investigated approaches; thus, *CSSICA* should be considered by the *composer* as an efficient and scalable approach.



## 1. Introduction

In an era where computation complexity is growing dramatically and achieving desired results depends on the processing of big data, the computation world has no choice but to recognize the use of cloud computing (Armbrust et al., 2010; Hayes, 2008). Choosing each of the deployment models of cloud computing (public, community, private, and hybrid clouds (Dillon, Chen, & Chang, 2010; Peter Mell, 2011)) would provide required service models (Ellinger, 2013) with different security policies (Takabi, Joshi, & Gail-Joon, 2010; Wei et al., 2014; Zissis & Lekkas, 2012). According to a widely accepted classification, each service can belong to one of the three categories: software as a service (SaaS), platform as a service (PaaS), or infrastructure as a service (IaaS), which can provide more effective functionalities in cooperation and combination of other services.

Nevertheless, the increasing tendency of applicants to receive services from the cloud has led to an unprecedented increase in the number of providers who want to present their services in a cloud service pool. Hence, we are faced with a large number of unique services provided with similar functionality and different quality of service (QoS) (Jula, Sundararajan, & Othman, 2014).

On the other hand, due to the availability of complicated and varied services, a distinct simple service is unable to meet the prevailing functional prerequisites for several real-world cases. A set of simple atomic services that are able to work together is necessary to perform a complicated service. We also encounter hanging customer requirements from simple services into complicated services, along with a set of constraints, priorities, and QoS requirements (e.g., service time). Therefore, cloud suppliers must provide a package, referred to here as the *composite service composer (CSC)*, which is a set of components that searches for the best composition of pre-provided unique services in the service pool based on customer requirements and constraints. Because of the immense variety of unique services and large number of service providers, as well as the importance of customer-defined requirements and constraints, *CSC* is faced with an NP-hard problem referred to as cloud computing service composition *(CCSC)* (Fei, Yuanjun, Lida, & Lin, 2013; Li, Cheng, Ou, & Zhang, 2010; Wada, Suzuki, Yamano, & Oba, 2012) when seeking the optimal response to any request for a composite service.

Many studies have been conducted and many different heuristic and non-heuristic algorithms (Barney, 2012; Gutierrez-Garcia & Sim, 2010; Kofler, Haq, & Schikuta, 2010; Kofler, ul Haq, & Schikuta,

* Corresponding author.
*E-mail addresses:* amin.jula@gmail.com, aminjula@ftsm.ukm.my (A. Jula), zalinda@ftsm.ukm.my (Z. Othman), elan@ftsm.ukm.my (E. Sundararajan).

http://dx.doi.org/10.1016/j.eswa.2014.07.043
0957-4174/