Swarm and Evolutionary Computation 10 (2013) 31–40

Contents lists available at SciVerse ScienceDirect

Swarm and Evolutionary Computation

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

Regular Paper

Memetic algorithms applied to the optimization of workflow compositions

Simone A. Ludwig

Department of Computer Science, North Dakota State University, Fargo, ND, USA

ARTICLE INFO

Article history:
Received 1 April 2012
Received in revised form
29 October 2012
Accepted 9 December 2012
Available online 11 January 2013

Keywords:
Genetic algorithms
Particle swarm optimization
Munkres algorithm
Quality of service
Memetic algorithms

ABSTRACT

The selection of services of a workflow based on Quality of Service (QoS) attributes is an important issue in service-oriented systems. QoS attributes allow for a better selection process based on non-functional quality criteria such as reliability, availability, and response time. Past research has mostly addressed this problem with optimal methods such as linear programming approaches. Given the nature of service-oriented systems where large numbers of services are available with different QoS values, optimal methods are not suitable and therefore, approximate techniques are necessary. In this paper, we investigate Genetic algorithms and particle swarm optimization for the service selection process. In particular, both methods are combined with an optimal assignment algorithm (Munkres algorithm) in order to achieve higher solution qualities (success ratios) and to form a so called memetic algorithm. Experiments are conducted to investigate the suitability of the approaches and to compare the memetic algorithms with their non-memetic counterparts. The results reveal that the memetic algorithms are very suitable for the application to the workflow selection problem.

1. Introduction

The distribution of software systems has increased over the last decade. There are many different reasons for this increase, and one of them is the need to integrate and connect heterogeneous applications and resources within organizational, but also across organizational boundaries. In particular, most legacy systems and applications were designed for their specific purpose, but not with the view in mind that they need to be integrated with and adapted to different application scenarios. This lack in the design of these legacy systems require new paradigms and approaches to be integrated to cope with these challenges.

Service-orientation provides the conceptual principle necessary to deal with the integration challenges as well as with the increasing complexity by providing adaptive software units referred to as services. Services are characterized by properties such as loose coupling, well-defined service contracts, and standardization that allows them to be independent of any particular implementation technology [1].

The idea behind service-oriented computing is that businesses offer their application functionality as services over the Internet, and users or companies can make use of these services by composing and integrating these services into their applications. Service-oriented architecture is the concept that combines this idea. Fig. 1 describes the basic components of the service-oriented architecture that are the providers, requesters and registries. The service provider publishes the service description in a service registry, and a service requester can query a service from the registry, and dynamically bind it to one of the services that are returned by the search query.

The main idea of service-orientation is to compose these services by discovering them and then dynamically invoking them when building applications, rather than building them from scratch or reusing other applications. Service composition (also called orchestration) enables the development of building service-oriented applications using existing services. The result of this composition process is referred to as a composite service. In this paper, we assume that all available services are validated pre-runtime, so that failure cannot occur during the composition process due to incompatibility. The standard orchestration language is WS-BPEL (Web Service Business Process Execution Language) [2]. Choreography is another term used in service-oriented environments that describes the message interchanges between the different participants in service-oriented systems. It provides the global distributed model of message exchanges without the need of a central coordinator [3]. The Web Service Choreography Description Language (WS-CDL) [4] is one of the first languages for describing the global model of service interactions.

One primary requirement of service-oriented systems is the ability of self-adaptivity [5]. Self-adaptivity in the area of service-oriented environments means that the system should be able to adapt its behavior depending on the changes within its environment. A possible solution for this adaptability, in particular for service composition, is the concept of Quality of Service

E-mail address: simone.ludwig@ndsu.edu

2210-6502/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.swevo.2012.12.001

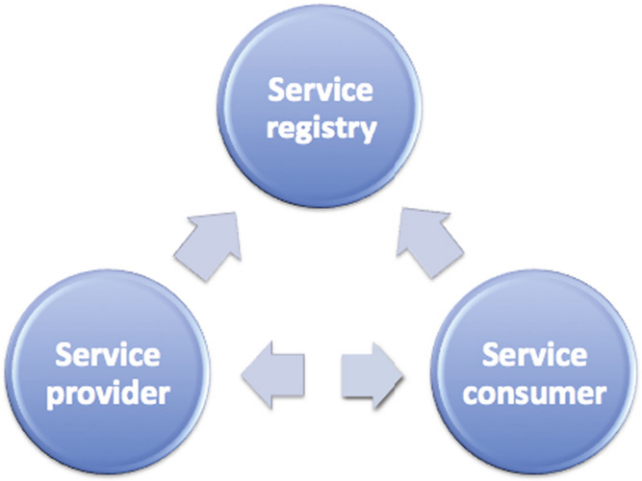

Fig. 1. Service-oriented architecture.

(QoS). All non-functional attributes of a service, such as performance-specific attributes, are described by QoS. QoS attributes can be categorized into deterministic and non-deterministic attributes [6]. Non-deterministic QoS attributes, such as response time, have uncertain values during service invocation. It is necessary to provide as accurate as possible values for all QoS attributes for composite applications and their execution.

A Service Level Agreement (SLA) is a contract between a service provider and a service consumer that captures the agreed-upon terms with respect to QoS parameters. Considering a service-oriented computing environment, capabilities are shared via the implementation of web services exposed to by a service provider. When a service requester requires a specific functionality, which cannot be provided by one single service, the composition of multiple services needs to be done thereby creating a workflow. The composition of web services should not only be functionally compatible, but should also be compatible with regards to the defined service levels. In particular, QoS attributes need to be considered for the dynamic binding process of the concrete services available. Therefore, QoS-aware composition is necessary given the changing and dynamic environment of service-oriented systems (services come online or go offline, new services become available, or existing services change their characteristics).

This paper addresses the workflow selection using approximate algorithms such as Genetic algorithms (GA) and Particle Swarm Optimization (PSO) for the optimization process. Furthermore, the selection of services is based on QoS parameters as well as on service level agreements. Preliminary results have shown that the approximate algorithms achieved an optimized service selection that were higher than that of a random selection and took less time than the optimal algorithm, however, the solution quality needs to be further improved. In order to address this, we apply the memetic algorithm idea of combining an evolutionary algorithm or swarm intelligence algorithm with a local search technique in order to balance the exploration and exploitation of the search space, and therefore, achieving higher solution quality. GA as well as a PSO approach is combined with an optimal search method called the Munkres algorithm. Both algorithms are implemented and experiments show that the combination achieves much higher solution qualities applied to the workflow selection problem than the basic GA and PSO. In particular, it balances the high computational complexity of the Munkres algorithm is balanced with the stochastic advantages of the GA and PSO.

This paper is structured as follows. Section 2 describes related work in the area. In Section 3, background information on workflows, quality of service, and service level agreements are given. Section 4 outlines and describes the approaches implemented and the experiments conducted. The results are displayed in Section 5. And last but not least, Section 6 summarizes the findings and gives an account to future work.

2. Related work

Related work in the area of service and workflow selection can broadly be classified into the following categories: web service infrastructure, agent-based, fuzzy-based, trust-based and optimization approaches. Some but not all of these approaches use QoS as a measure for the selection process.

A dynamic web service selection and composition approach is described in [7]. The approach determines a subset of web services to be invoked during runtime in order to orchestrate a composite web service successfully. A finite state machine model is used to describe the permitted invocation sequences of the web service operations, and a reliability measure is aggregated for the web service operations.

A transactional and QoS-aware selection algorithm is proposed in [8]. The composition of services is recursively constructed based on diverse functionalities, transactional properties and QoS thereby considering the user's requirements. This approach addresses the composition of a workflow based on transactional properties and QoS characteristics using proofs as well as an experimental analysis.

An adaptive hybrid semantic matchmaker for services is proposed in [9]. The matchmaker determines three kinds of semantic service similarity that are logic-based, text-based and structural-based. The degree of structural similarity is computed with the help of the SWDL-analyzer tool by means of XMLS tree edit distance measurement, string-based and lexical comparison of the respective services.

An open, fair, dynamic and secure framework to evaluate the QoS of services is outlined in [6]. The fair computation and enforcement of QoS of web services should have minimal overhead, but yet should be able to achieve sufficient trust by both service requesters and providers. A case study of a phone service provisioning market place application shows the idea of the approach.

The area of agent-based service selection has looked at ways to formulate the problem of service selection. In [10], inspiration from traditional recommender approaches is taken and a new agent-based approach in which agents cooperate to evaluate service providers is proposed. The agents rate each other, and decide on the weight to place on each other's recommendations. The algorithm is devised to work in the context of a concept lattice that enables to find relevant agents.

Another approach [11] developed a multi-agent framework based on an ontology for QoS and a new model of trust. The ontology provides the basis for the providers to advertise their service capabilities and for consumers to express their preference in order for ratings of services to be gathered and shared. The ratings give an empirical measure for the selection of services, and the ratings are quality-specific and obtained by automatic monitoring or user input.

A trustworthy service selection and composition framework based on Bayesian networks and a beta-mixture model is presented in [12]. This approach was devised since existing approaches either failed to capture the dynamic relationships between services or assumed that the environment is fully observable. Experimental results show that their approach punish and reward services in terms of the quality criteria they offer, also