The Journal of Systems and Software 90 (2014) 191–203

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Accurate sub-swarms particle swarm optimization algorithm for service composition

Jianxin Liao [a,b,*], Yang Liu [a,c], Xiaomin Zhu [a,b], Jingyu Wang [a,b]

[a] State Key Lab of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[b] EB Information Technology Co., Ltd., 7-8th Floor, Kunxun Masion, No. 9 Zhichun Road, Haidian District, Beijing 100191, China
[c] The Export-Import Bank of China, No. 77, BeiHeYan Street, DongCheng District, Beijing 100009, China

ARTICLE INFO

Article history:
Received 10 January 2012
Received in revised form
27 November 2013
Accepted 27 November 2013
Available online 4 December 2013

Keywords:
Service composition
Particle swarm optimization
Multi-constraint optimal service

ABSTRACT

Service composition (SC) generates various composite applications quickly by using a novel service interaction model. Before composing services together, the most important thing is to find optimal candidate service instances compliant with non-functional requirements. Particle swarm optimization (PSO) is known as an effective and efficient algorithm, which is widely used in this process. However, the premature convergence and diversity loss of PSO always results in suboptimal solutions. In this paper, we propose an accurate sub-swarms particle swarm optimization (ASPSO) algorithm by adopting parallel and serial niching techniques. The ASPSO algorithm locates optimal solutions by using sub-swarms searching grid cells in which the density of feasible solutions is high. Simulation results demonstrate that the proposed algorithm improves the accuracy of the standard PSO algorithm in searching the optimal solution of service selection problem.

1. Introduction

In a service overlay network (SON) (Duan et al., 2003), legacy and new applications are encapsulated into atomic services and implemented on different service nodes (SNs). Unlike the typical provision of developed applications, SON provides a novel method to develop and provide applications in the mean time. In a SON, the system executes several services to compose them to a composite application. This process is known as service composition (SC).

Service composition can be divided into three steps in process: (1) submission of a composite request; (2) service selection; (3) service execution and result acquirement. In the first step, the user submits the functional descriptions like the goal of the composite application and the non-functional specification like the QoS constraints of the composite application. The composite application can be modeled as a workflow or a directed attribute graph (DAG). In the second step, the system selects the optimal service instances based on the QoS constraints, e.g. delay, cost, reliability. Then the system invokes these services and returns the result to the user in the last step.

Because services in a SON are implemented and distributed on diverse SNs independently, duplicated service instances (SIs) are inevitable. Duplicated implies that SIs with the same function are implemented by different developers, deployed on different SNs. Consequently, for a composite application, multiple SIs with different non-functional parameters exist for each service. The selection of optimal service instance for each service can barely guarantee the optimal QoS parameters of the whole composite application. Therefore, it is a challenge work to select optimal SIs compliant with the requirements of whole composition request.

Particle swarm optimization (PSO) is known as an effective and efficient algorithm which can support multi-constraint multi-objective problems, whatever linear or nonlinear. In previous literatures (Guha and Ludwig, 2008; Xia et al., 2009; Cao et al., 2010), PSO has been proposed to solve the service selection problem properly. In our previous research (Liao et al., 2011), Niching PSO has been adopted for multi-constraints single-objective service selection problems in SON. However, PSO's premature convergence and diversity loss always results in suboptimal solutions regardless of true optimal solutions. Sub-Swarm techniques have been proposed to improve the accuracy of PSO. However, these techniques all concentrate on the improvement of particles' neighbor topology. They neglect the reason of premature convergence, i.e. suboptimal solutions near true optimal solutions form traps what pull particles to converge to them. Inspired by this phenomenon,

* Corresponding author at: State Key Lab of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China. Tel.: +86 137 2005 5806.
E-mail addresses: jxlbupt@gmail.com (J. Liao), sunovo2008@gmail.com (Y. Liu), zhuxiaomin@ebupt.com (X. Zhu), wangjingyu@ebupt.com (J. Wang).

0164-1212/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2013.11.1113

we propose an accurate sub-swarms particle swarm optimization (ASPSO) algorithm which locates optimal solutions by using sub-swarms searching grid cells in which the density of feasible solutions is high. It adopts parallel and serial niching techniques to compare suboptimal solutions found in diverse sub-swarms which also avoid diversity loss in PSO. Furthermore, it uses an exterior file to record feasible solutions found in the whole solution space. Simulation results reveal that the accuracy of proposed algorithm is higher than the ones compared in searching the optimal solution of service selection problem.

The rest of paper is organized as follows. Section 2 presents the related work. Section 3 presents the system model and problem formulation. The proposed algorithm is described in Section 4. Section 5 presents simulation results and analyses the performance of the proposed algorithm. Finally, Section 6 concludes this paper.

2. Related work

Nowadays, extensive studies have been conducted related to service composition. Papers of this domain mainly refer to two aspects: semantic discussion and non-functional properties discussion of service composition.

2.1. Researches mainly focusing on semantic discussion

Sirin et al. (2004) studied applying hierarchical task network (HTN) planning to automatic web service composition. This paper is the first research proposing an algorithm which translates OWL-S (Ontology Web Language for Services) service descriptions to the HTN planning system SHOP2's (simple hierarchical ordered planner 2) domain. Kalasapur et al. (2007) went further in the semantic research of service composition. They proposed a pervasive information communities organization (PICO) middleware to support service composition on semantic and syntactic layers. The semantic layer represents services' input and output types. The syntactic layer represents services' input and output formats. In service composition, the first step is to check whether there is a path between the input type and the output type of tasks in the semantic layer. If there is a path, the second step is to find the shortest path in the syntactic layer.

Previous literatures always separate semantic service selection and QoS optimization in the service composition. Fujii and Suda (2005) proposed a semantic-based service composition architecture. This architecture consists of a semantic representation module (CoSMoS), a service discovery middleware (CoRE) and a service composition mechanism (SeGSeC). Yue et al. (2007) presents an approach for automating geospatial Web service composition by employing geospatial semantics in SOA. This paper shows how ontology-based geospatial semantics are used in a prototype system for enabling the automatic discovery, access, and chaining of geospatial web services. Ngu et al. (2010) proposed a method to enable progressive composition of non-Web-service-based components. Their approach exploits semantic annotation together with the standard semantic Web matching algorithm for finding sets of functionally equivalent components out of a large set of available non-Web-service-based components. Lécué and Mehandjiev (2011) discussed a novel service composition model combining user-centric quality and semantic matching quality together. Then they also proposed a genetic algorithm based approach for this model in the service selection optimization. Pastrana et al. (2011) present a framework to specify services' QoS properties by connectors defined by web service client and automatically generated by the COMPOSITOR tool they have developed. Connectors use contracts based on OWL ontology of the server domain to express the non-functional requirements and the behavior desired by the client of a service. These self-adaptive connectors could promote the interoperability and cooperation of the service selection process.

2.2. Researches mainly focusing on non-functional properties discussion

In another aspect, there are a few papers addressing non-functional properties optimization in service composition. Zeng et al. (2004) proposed local and global algorithms for QoS-aware service composition. This paper uses execution paths and plans to model all possibilities of the composite application. The local algorithm selects the optimal service for each task in the composite application. The global algorithm selects the optimal plan for all paths based on integer programming. Later papers applied this basic approach to diverse scenarios. Estévez-Ayres et al. (2009) proposed off-line and on-line algorithms for real-time service composition. The off-line algorithm computes all possible combination for selecting the optimal service path using a depth first search algorithm. The on-line algorithm improves the previous one by using a pruning technique. Gu and Nahrstedt (2006) presented a decentralized service composition framework, called SpiderNet. SpiderNet maps the function graph into the best qualified ServFlow, which minimizes the utility function and conforms to QoS constraints in the meantime. The bounded composition probing (BCP) algorithm is proposed to probe service instances hop by hop. At last, SpiderNet sorts all qualified ServFlows and then selects the optimal one for execution. Yu et al. (2007) studied two structure of service selection problem: sequence structure and general structure. Then they proposed several algorithms based on combinatorial and graph models for each structure. Oh et al. (2008) developed an AI planning-based heuristic Web service composition algorithm WSPR to minimize the cost of composite web service. This paper also studied existing public web service sets, and developed a web service discovery and composition benchmark tool based on WSPR. Ko et al. (2008) applied tabu search and simulated annealing meta-heuristics to the proposed constraint satisfaction web service composition algorithm. They also presented a QoS-oriented web service composition planning architecture composed of a composition broker and an execution plan optimizer. Hwang et al. (2008) focused on the reliability of service composition. This paper proposed two dynamic web service selection strategies which select the candidate composite web service with max aggregated reliability. Then they applied these strategies to a BPEL service composition prototype. Park and Shin (2008) proposed three power-aware algorithms for service composition on mobile devices: (1) dynamic QoS control for mobile applications; (2) reconfiguration of the Service Composition Graph; (3) Service Discovery and Routing. Huang et al. (2009) proposed an integer programming approach with pruning and branch-and-bound methods. This algorithm can be used in finding the optimal services which maximize the objective function, sum of services' attribute value. Haddad et al. (2010) proposed a service selection algorithm considering not only QoS properties but also transactional characteristics. This paper specified transactional attributes of web services and composite web services. The transactional properties are mainly related to fault tolerance and rollback. Lin et al. (2011) proposed a relaxable QoS-based service selection algorithm which takes effect when there is no candidate solution for specific service composition constraints. This algorithm could enhance the reliability and availability of Service composition systems.

In researches on non-functional properties discussion, there are some literatures using PSO. Discrete PSO can be directly adopted to solve web service matching problems between requesters and providers, whatever single-objective (Ludwig and Schoene, 2011)