ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 9, No. 3, 2014, *pp*. 163-168

An Effective QoS-aware Web Service Selection Approach

Tongguang Zhang

Department of Computer and Information Engineering
Xinxiang College, Xinxiang, Henan 453000, China
jsjoscpu@163.com
(*Received November 10, 2012, accepted January 3, 2014*)

Abstract. Service computing provides the capability of binding each service invocation in a composition to a service selected among the same function attributes but different QoS attributes to achieve a QoS goal. Hence, service selection approach plays a very important role in service composition. However existing approaches little considered the computation time. In this study, we propose a effective QoS-aware web service selection approach. This approach adopts a particle swarm optimization algorithm select the most service with users' QoS requirements Experimental results show that our approach can find best suitable services with low time cost in web service selection.

Keywords: Service computing, Web service selection, QoS

1. Introduction

At present, there has been a more than 130% growth in the number of published web services in the period from October 2006 to October 2007 [5-6]. Therefore, service requesters will be faced with a huge number of variations of the same services offered at different QoS. In addition, the QoS requirements dynamic changes can occur at run-time, which means that a quick response to adapt to the requests is important in web service selection [5-6]. Hence, the need of an effective and efficient service selection approach will increase.

The problem of QoS-aware web service selection has received a lot of considerable attention. In [7], the authors proposed two heuristic algorithms based on linear programming to find near-optimal solutions that may be more efficiently than exact solutions suitable for making runtime decisions. The improvement of the two algorithms is significant compared with exact solutions, but both algorithms do not scale when web services increase, which remains out of real time requirements. The authors of [8] adopted the global optimization approach to find the best service components for the composition by linear programming. Soon afterwards the authors of [9] also used linear programming to optimize user's end-to-end QoS constraints, but it differs in the solution of the optimization problem. Furthermore, the work of [10] extends the linear programming model to include local constraints. These linear programming approaches are effective when the size of the problem is not very large. However, their scalability is very poor due to the exponential time complexity of the applied search algorithms with the increasing size of the problem [11].

Although efforts above have been made and obtained in web service composition area, existing technologies on web service selection little considered the computation time or time complexity. They are still not mature yet and require significant efforts. In this paper we propose an approach to select web services for composition with particle swarm optimization. We evaluate proposed approach experimentally on real QoS data. Experimental results show that our approach significantly improves the time performance of web service selection process in service composition system.

The remainder of this paper is organized as follows. Section II introduces web service selection. Our research approach is proposed in Section III. Experiments are presented in Section IV. Finally, Section V is our conclusions.

2. Web Service Selection

In service composition, service candidates have a number of different QoS attributes, leading to large variation in their QoS attribute values or scopes. Considering global QoS constraints, computing or evaluating QoS is different for each service candidate. So QoS utility function was proposed in [7]. The function maps the quality vector Qs into a single real value, enabling sorting and ranking of service candidates and simplifying choosing to satisfy QoS constraints of the service components. The QoS utility function in this paper is similar to [8]. For example, in the sequential composition model, the overall utility of a composite service S is computed as

$$U(s)=\sum_{k=1}^{r}\frac{Q_{j,k}^{\max}-q_k(s)}{Q_{j,k}^{\max}-Q_{j,k}^{\min}}.w_k \tag{1}$$

$$U(\mathrm{S})=\sum_{k=1}^{r}\frac{Q_{k}^{\max}-q_k(\mathrm{S})}{Q_{k}^{\max}-Q_{k}^{\min}}.w_k \tag{2}$$

with

$$Q_{j,k}^{\max}=\max_{\forall s_{ji}\in S_j} q_k(s_{ji}),Q_k^{\max}=\sum_{j=1}^{n}Q_{j,k}^{\max}$$

$$Q_{j,k}^{\min}=\min_{\forall s_{ji}\in S_j} q_k(s_{ji}),Q_k^{\min}=\sum_{j=1}^{n}Q_{j,k}^{\min}$$

where $w_k\in R^+(\sum_{k=1}^{r}w_k=1)$ represents users preferences, $Q_{j,k}^{\min}$ is the minimum value of the k-th attribute in all service candidates of the service class S_j and similarly, $Q_{j,k}^{\max}$ is the maximum value, $Q_k^{\min}$ is the minimum value of the k-th attribute of S and similarly, $Q_k^{\max}$ is the maximum value. Due to space limited, some concepts about service selection are omitted here. More information is in [7].

As it is well known that the service selection with global QoS constraints is an optimization process. The optimal selection for a given service composition S must meet the following two conditions:

1) An given vector of global QoS constraints $\mathrm{CS}=\{C_1,\cdots,C_m\},0\le m\le r$, $q(\mathrm{S})\le C,\forall C_k\in\mathrm{CS}$ ($q(\mathrm{S})$ is the aggregated QoS value of the composition service).

2) The maximum overall utility value $U(\mathrm{S})$ in the composition service.

However, finding the optimal composition requires enumerating all possible combinations of service candidates, which can be very expensive in terms of computation time. Therefore, we propose an approach to solve this problem.

3. PROPOSED SERVICE SELECTION APPROACH

Particle swarm optimization (PSO) as a parallel optimization algorithm can be used to solve a large number of complex and non-linear problems, and has been widely used to science and engineering for example function optimization, pattern classification and resource allocation fields [8]. PSO is most frequently applied to solve continuous optimization problems. At present, how to apply PSO to discrete optimization problems, especially combinatorial optimization problems, is an important research direction. In recent years, many researchers have proposed improved PSO for example in [12] for combinatorial optimization problems and obtained many good optimization solutions. Therefore, we also use PSO to find optimal quality control lines (local constraints).

For a canonical PSO, the velocity of each particle is modified iteratively by its best personal position, and the position of best particle from the entire swarm. As a result, each particle searches around a region defined by its best personal position and the best position of the population. For instance, a swarm consists of Np particles moving around in a D-dimensional search space. Every particle in the swarm has a position $x_{id}\,(i\in Np)$, a velocity v_{id} and a memory p_{best}, i.e., its best personal position (local optimization) for its best found position, which are all updated in every iteration step of PSO. The best solution p_{gbest} (the best position it has found so far, i.e., global optimization) depends on the iteration number (i.e. $it\max$). At the beginning, Np particles are initialized with a random position. The fitness of all initial position is evaluated by the fitness function, leading to an initial p_{gbest}. During every iteration step of PSO, The position of the i-th particle at the next iteration will be calculated according to the following equations[13]:

$$\begin{aligned}v_{id}^{t+1}&=w\times v_{id}^{t}+c_1\times rand()\times(p_{best}(t)-x_{id}^{t})\\&\quad+c_2\times rand()\times(p_{gbest}(t)-x_{id}^{t})\end{aligned} \tag{3}$$

$$x_{id}^{t+1}=x_{id}^{t}+v_{id}^{t+1} \tag{4}$$