Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 1170 – 1175

Procedia Computer Science

The Eleventh International Symposium on Intelligent Techniques for Ad hoc and Wireless Sensor Networks (IST-AWSN16)

Distributed Approach for QoS Service Selection in Web of Objects

Nacera TEMGLIT[a], Abdelghani CHIBANI[b], Karim DJOUANI[b], Mohamed AHMED NACER[c]

[a]*Ecole nationale Suprieure d'Informatique (ESI), Algiers 16000 ALGERIA*
[b]*LISSI Laboratory, Paris Est University (UPEC), Paris 94000 FRANCE*
[c]*LSI Laboratory, USTHB University, Algiers 16000 ALGERIA*

Abstract

In the Internet of Things (IoT), web objects and services can be seen as distributed and cooperative agents that need to collaborate in order to reach advanced functionalities and also to optimize the overall quality offered to the end users. For this purpose, we propose in this paper a distributed and optimal QoS services selection approach based on Multi-Agents paradigm and Distributed Constraints Optimization Formalism (DCOP). Hence, we revisited a DCoP technique for developing a new complete algorithm for service selection: the 'Synch4QoS' implemented under a multi-agent architecture using a real-time Web protocol for communication. The proposed algorithm takes into accounts the specificities of the service composition context and the satisfaction of the global user's constraints.

Peer-review under responsibility of the Conference Program Chairs

Keywords: ubiquitous web services, QoS web service selection, DCOP formalism, multi-agent paradigm

1. Introduction

Our work is part of the European project ITEA2 WoO, which aims to propose a new architecture that connects, through web protocols and services, objects like sensors, actuators, robots or any physical or virtual object, as well to solve the central problem of services composition. The objective is to select the adequate services from those available in the ubiquitous environment to optimize the entire quality of a composite service expected by the end users. However, in the case of robotic services, the complexity of service selection will increase as those services are characterized by their dynamic aspect due to robots mobility, limited connectivity in some situations, intensive use of battery to handle actuation and also limited memory. Thus, we need a sound model of QoS parameters to capture the descriptions of the important aspects of an ubiquitous service. Hence, we propose a model that allows for specifying generic QoS attributes like *response time*, *availability* and the *price* of the service as well as domain-specific QoS

* Nacera TEMGLIT. Tel.: +213-673-636-236
E-mail address: n_temglit@esi.dz

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.240

attributes, e.g., the *Energy Level* of the ubiquitous objects. QoS attributes of ubiquitous services are also qualified as static or dynamic. The dynamic attributes represent the variable characteristics of a given service such as the *response time*, *energy level*, *availability* and their values are provided at the service invocation time by a monitoring process.
In this paper, a distributed optimization approach is proposed to solve the problem of service selection in the context of service Choreography instead of a central service Orchestration. In the literature, various techniques have been proposed for optimal or near-optimal web service selection problem in the context of centralized and static environment [1,2,3,4]. However, very few approaches have been proposed to solve the problem of optimal service selection in decentralized environment, most of them focused on a local service selection at each service provider component of the distributed environment by avoiding communication between objects [5,6]. However the optimality is never achieved especially when the constraints of end users are globally expressed. In the present work, we aim to tackle the problem of global service selection in distributed environment by taking advantage of the multi-agent paradigm and DCOP formalism [7]. In other word, the objective is to implement an approach for selecting the best candidate services that maximize the global service quality while satisfying the global QoS user's constraints. Hence, we revisited a DCOP algorithm based on the well known Branch and Bound optimization technique for developing new complete algorithm 'the Synch4QoS' seeking to maximize or minimize the QoS attributes, such as minimizing *response time* and maximizing *availability*. To address the trade-off between different objectives, we take advantage of Simple Additive Weighting (SAW) techniques and use utility function method. The 'Synch4QoS' algorithm has a simple control structure that makes it easy to be implemented on low computing capability systems but requires intensive messages exchanges. Two techniques are discussed for reducing the number of exchanged messages and pruning unpromising candidates services. In addition, the *Synch4QoS* takes into account the composition patterns (sequence, parallel and loop) that are considered as means to compute the QoS aggregations (Constraints on QoS aggregations).
The rest of the paper is organized as follows: some related works of the domain will be reviewed in section 2. In section 3, we give a DCOP formulation of the services selection problem, we describe also our proposed algorithm 'Synch4QoS'. We present our initial experiment and results about the solution in section 4. We conclude and give some perspectives to the work in section 5.

2. Related Work

Several solutions to the Web service selection problem have been reported in the literature including, integer programming (IP), genetic algorithms and CSP (Constraint Satisfaction Problem) formalism based techniques. The work of Zeng at al. [1] focuses on dynamic and quality-driven selection of services. The authors use Mixed Linear Programming (MLP) techniques to find the optimal selection of component services. But, these methods assume linearity of the constraints and the objective function and suffer from poor scalability. In [8,9,10,11] Genetic Algorithms (GA) based methods are proposed to solve the selection problem in a reasonable time. In [12], the author proposes an improved version of the standard genetic algorithm approach by making use of fuzzy logic during the stochastic genetic search process to adjusts dynamically the crossover and mutation rates of the generations evolution. The result shows improvement of the quality solution but the optimality still not guaranteed. The work in [13] uses also the crossover and mutation operators of genetic algorithm to the hybrid particle swarm optimization algorithm used for service selection in the Cloud. However, as users have often to fix a priori a constant number of iterations or a time deadline to stop the Genetic algorithm, this does not give any guarantee about the quality of the generated solution. Therefore, the Genetic Algorithm offers a better scalability but deemed non useful for selecting the optimal composition plan. There has been very few works in the area of distributed QoS selection. In [14], the authors combine global optimization with local selection techniques. First, global constraints are decomposed into local constraints using mixed integer programming (MIP) such that the satisfaction of local constraints guarantees the satisfaction of global constraints. The second step is to use a local selection on each component independently to find the best web services that satisfy these local constraints. But in this work, the method of extracting QoS levels from the QoS information of service candidates is greedy, this leads to very restrictive decomposition of the global constraints to local constraints that cannot be satisfied by any of the service candidates. [15] and [16] base their approaches on a decomposition of the global objective on the local domains on the Cloud and use a Genetic Algorithm to find the best composition, however, this can only produce locally approximate solutions. Regarding the existing tools within the domain of DCOPs, the Framework FRODO [17] has been considered. It is an open source framework dedicated to

distributed combinatory optimization problems supporting the majority of DCOP algorithms. However, we could not use it for the case of the present work as it only allows simple and integer values to be exchanged between agents, whereas in our approach the agents are exchanging vectors of real values representing the parameter's values. In addition, some other domain specificities have to be considered like defining new aggregation operators in order to take into account the patterns of composition when calculating the cost.

3. The Distributed Service Selection Approach

3.1. QoS Model

In the context of ubiquitous environment, the QoS parameters can be dynamic or static [18]. Static parameters (SQP) are generally known at the deployment time and are usually not updated during the execution time, for example the *Price* and the *Security Level* of a service. Dynamic parameters (DQP), represent the variable characteristics of a given service such as the *Response time*, *Availability*, *Reliability* and *Energy level* of the ubiquitous objects. DQP attributes are determined at the invocation time and their values are provided by a monitoring process. Therefore, the QoS of a concrete service cs_i is denoted by a vector of the five QoS attributes that can be represented as follows: $QoWS(cs_i) = (Price(cs_i), Secur(cs_i), AV(cs_i), EL(cs_i), RT(cs_i), RE(cs_i))$. QoS values of a composite web service are

Table 1. Aggregation of QoS values based on the pattern: S, P, L

QoS parameter	S (Sequential)	P (Parallel)	L (Loop)
Price	Price(a)+Price(b)	Price(a)+Price(b)	N*Price(a)
Secur	min(secur(a),Secur(b))	min(secur(a),Secur(b))	Secur(a)
EL	EL(a)+EL(b)/2	min(EL(a),EL(b))	$\sum_{i=0}^{N} EL_i$
RE	RE(a)*RE(b)	RE(a)*RE(b)/2	$\prod_{i=0}^{N} RE_i$
RT	RT(a)+RT(b)	max(RT(a),RT(b))	$\sum_{i=0}^{N} RT_i$
AV	AV(a)*AV(b)	AV(a)*AV(b)	$\prod_{i=0}^{N} AV_i$

calculated according to the candidate QoS values and composition patterns: Sequential, Parallel and Loop (see table 3.1). The aggregated quality $c_{a,b}$ between two concrete services a and b is computed according to SAW (Simple additive Weighting) technique used for multi-objective problems:

$$c_{a,b} = \sum_{(k=1)}^{L} Agg(q_k(a), q_k(b)) * w_k \tag{1}$$

with $\sum_{i=1}^{L} w_i = 1$, L is the number of QoS attributes and Agg denotes the aggregation operator (see table 3.1).

3.2. Problem formulation as DCoP

In our approach, we deal with two services conceptualizations, namely: concrete services and abstract services. A concrete service is any peace of software that encapsulates a given functionality like computations or access to raw data of sensors. An abstract service is a set of functionally-equivalent services (with similar input and output data) but with distinct QoS properties. An abstract composite service consists of a set of n abstract services denoted as $AS = \{as_1, as_2, ..., as_n\}$. For each service $as_i \in AS$, there are m candidate services proper to implement it (concrete services), which are represented by $CS_i = \{cs_1^{asi}, cs_2^{asi}, ..., cs_m^{asi}\}$. The problem is to select distributively one concrete service cs_j^{asi} for each as_i of AS that optimize the global QoS of the whole composite service. This is formulated as a DCOP problem as depicted in table 3.2.

Within the DCOP framework, the agents have to communicate in order to converge to an optimal solution, i.e. the best value of their respective variables leading to the minimal(optimal) value of the global cost $\phi(A)$, which is the summation of local costs (c_{ij}) for each couple of variables (services i, j) sharing one constraint.

Table 2. Service selection problem as DCOP.

DCOP concepts	Web service composition concepts
X the set of distributed variables $\{x_1, x_2, \cdots, x_n\}$	the abstract services set $AS = \{as_1, as_2, ..., as_n\}$
A the agents set $\{A_1, A_2, \cdots, A_k\}$	the agents set $\{A_1, A_2, \cdots, A_k\}$
$D = \{D_1, D_2, \cdots, D_n\}$ a set of finite sets , where D_i is the domain of the variable x_i	$D(as_i) = \{cs_j^{asi} / 1 \leqslant j \leqslant m\} = CS_i$, the value of each cs_i is the vector of its QoS parameters values
Graph of constraints	The composition plan
$C = \{c_{ij} : D_i \times D_j \rightarrow \mathrm{R}^+,$ with $i, j = 1, \cdots, n$ and $i \neq j\}$ a set of local costs function for each couple of variables x_i, x_j	The local cost c_{ij} between two given services i, j is calculated according to their pattern of compositions (Loop, Sequence, parallel) (see formulas 1)

$$\phi(Af) = \sum_{x_i, x_j \in X} c_{ij}(Af(x_i), Af(x_j)) \tag{2}$$

with Af the assignment function that map each variable x_i a value d_i from D_i.

We also need to extend the DCOP algorithm to support satisfaction of global constraints on attributes specified by the user, these are bounds on the QoS attributes. For example, the user may expect that the entire service availability shall never be under 60% and the price should not exceed 800 dollars. Generally, let V_r denotes the r^{th} QoS attribute value obtained for a feasible solution (feasible plan). The QoS Bounds vector on L' QoS attributes is expressed as: $Bound$=($Bound_1, \cdots, Bound_{L'}$), with L' is the number of constraints. A boolean function CONSTRAINT (V_r,$Bound_r$,Op_r) is defined to verify whether the r^{th} constraint specified by the user is satisfied, the operator op_r can be $>, <, \leqslant, \geqslant, or =$. V_r is obtained like this:

$$V_r = \sum_{a,b \in CS} Agg(q_r(a), q_r(b)) \tag{3}$$

with $q_r(a), q_r(b)$ denote the given r^{th} QoS attribute values of the concrete services a, b.
Therefore, solving the problem of service selection is to find a set of concrete services $CS = \{cs_1, cs_2, \cdots, cs_n\}$ with the cs_i the concrete service selected for abstract service as_i such that:

1. The global cost defined by the function ϕ in formulas 2 is minimized.
2. The global quality of CS on each QoS attribute (V_r) must satisfy the user's constraints ($Bound_r$):

$$\bigwedge_{r=1}^{L'} CONSTRAINT(V_r, Bound_r, Op_r) = true \tag{4}$$

3.3. The Synch4QoS Algorithm

We have revisited the Synchronous Branch & Bound optimization technique to develop an efficient and complete algorithm for our distributed services selection problem called 'Synch4QoS'. In synch4QoS, the agents, associated to a single variable, are sorted sequentially and change the values of their variables synchronously, one after the other. The process works like the centralized branch and bound ($B\&B$) algorithm using depth-first search. In a distributed version of the $B\&B$, the upper bound is chronologically propagated back and forth, together with the forward value assignment messages and the backward backtrack messages. The exchanged messages contain also, the global cost of the partial solution ($\phi(Af)$) which is calculated progressively according to patterns of composition (each agent has a local view of the global composition structure). To deal with user's constraints, we added to the message the vector (V_r)representing the current global quality on each QoS attributes. The agents compare these values to the bounds specified by the end user ($Bound_r$).

3.4. Discussion

SynchBB4QoS turns on a simple configuration of agents (Sequential order) regardless of the structure of the composite service, performs a full search, ie, it guarantees to find the optimal solution and don't require a lot of memory.

However, it delivers at worst an exponential number of small messages and hence an important execution time. To reduce the number of exchanged messages, two preliminary steps are performed locally at each agents:

- Pareto Search for reducing the number of candidate services: Pareto search[19], widely used in multi-criterion problem, performs the pair-wise comparison of candidate services in terms of the Pareto-dominance relationship. In other word, for any two candidate services cs_i and cs_j, if and only if service c_i is better than (smaller than in our case) or equal to service cs_j in terms of QoS values on all dimensions, and better on at least one dimension, cs_i is Pareto dominating cs_j. Dominating services are then the potential candidates for the composition, the rest are pruned from the set. This can reduce significantly the number of candidate services per variable (agent).
- Local ordering of candidate services: How the agents change their values has a great impact on the speed of the search with bound propagation. The objective is to put on the top the most promising services, i.e. those seeking to have a good upper bound on the cost . In our case, those are the services with minimal local costs $C(S_j)$:

$$C(S_j) = \sum_{(k=1)}^{L} Q(k, j) * w_k \tag{5}$$

$Q(k, j)$ is the k^{th} attribute value of service j
Hence, the first ranked concrete services help to get a good upper bound without going deeper into the search tree and hence backtracks are performed earlier in the search.

4. Implementation

Fig. 1. Comparing SynchBB4QoS without/with using the two pruning techniques

The composition architecture includes a set of agents that communicate in an adhoc and asynchronous way by using point to point HTTP communication protocol (using format such as JSON or XML) and publish subscribe messaging middleware (using XMPP protocol[20]). These agents can be deployed in the cloud as well as on services end-user platforms such as smartphones, slates, sensor motes, home appliances or robots according to Ubistruct REF middleware[1]. The experiments has been done on a Windows PC Core i3 with 4 Go RAM with a processor at 2.1 GHZ. Our programs were implemented in Java under eclipse. Each agent knows its own login for the GTalk server (for XMPP communcation), the list of the concrete services (the candidate services) and its own local orchestration plan including its neighbors in the sequential order required by SynchBB4QoS algorithm coupled with their composition pattern (S, P, L). Each agent has also access to the end user's requirements. The curves depicted in figure 4 compare the performance of the SynchBB4QoS algorithm with/without using the pruning techniques. We fixed in the first time, the number of abstract services (the number of agents) m to 10 and varied the number of concrete services n from 2 to 10. In the second time, we fixed the number of concrete services to 10 and varied the number of abstract services. The values of the six QoS parameters have been generated randomly for the n concrete services. In the two problem

[1] http://ubistruct.ubiquitous-intelligence.eu/architecture

instances, we can see clearly that SynchBB4QoS using the pruning techniques has a better scalability of exchanged messages and hence a better execution time.

5. Conclusion

We presented in this paper an approach to address the problem of distributed service selection in ubiquitous environment based on multi-agent paradigm and DCOP formalism. Hence, we proposed a distributed algorithm, the SynchBB4QoS, revisited from the Branch & Bound optimization technique to meet the need of optimization and satisfaction of global QoS user constraints. SynchBB4QoS has a simple control structure that makes it easy to be implemented on low computing capability systems (limited memory) but requires intensive messages exchanges. We enhanced the SynchBB4QoS by proposing two techniques: pruning the non promising services from each agent service set using the Pareto-Dominance relationship and pruning search by ordering candidate services according to a local cost function. We are currently completing the integration of the selection module in the composition architecture (UbiSCo) and investigating how to extend the solution to manage multiple related service compositions, where each agent has to resolve locally a sub-problem of web service composition while optimizing the global solution. Besides, we are implementing another distributed algorithm of dynamic programming for the problem of service selection which uses a linear number of messages.

References

1. L.Zeng, B.Benatallah, Qos-aware middleware for web services composition, IEEE Transactions on Software Engineering 30 (5) (2004) 311–327.
2. T. Yu, Y. Zhang, K.-J. Lin, Efficient algorithms for web services selection with end-to-end qos constraints, ACM Transactions on the Web TWEB 1 (6).
3. D. Ardagna, B. Pernici, Adaptive service composition in flexible processes, Software Engineering, IEEE Transactions on 33 (6) (2007) 369–384.
4. N. M. Lécué, Towards scalability of quality driven semantic web service composition, in: Web Services. ICWS, 2009.
5. T. R. Alrifai, Combining global optimization with local selection for efficient qos-aware service composition, in: WWW '09: Proceedings of the 18th international conference on World Wide Web, New York, USA, 2009, pp. 881–890.
6. J. Li, Y. Zhao, M. Liu, H. Sun, D. Ma, An adaptive heuristic approach for distributed qos-based service composition, in: Computers and Communications (ISCC), 2010 IEEE Symposium on, 2010, pp. 687–694.
7. W. M. Modi, Adopt: Asynchronous distributed constraint optimization with quality guarantees, Artifcial Intell. (2005) 149–180.
8. M. Gao, H. C., Qos-aware service composition based on tree-coded genetic algorithm, in: COMPSAC '07: Proceedings of the 31st Annual International Computer Software and Applications Conference, Washington, USA, 2007.
9. P. Vanrompay, Y.B, Genetic algorithm-based optimization of service composition and deployment, in: SIPE '08: Proceedings of the 3rd international workshop on Services integration in pervasive environments, New York, NY, USA, 2008, pp. 13–18.
10. X. Wu, J. Y. Xiong, C. Y. C. J, Qos-driven global optimization approach for large-scale web services composition, Journal of Computers 6 (7) (2011) 1452–1460.
11. N.Sasikaladevi, L.Arockiam, Genetic approach for service selection problem in composite web service, International Journal of Computer Applications 44 (4) (2012) 22–29.
12. M.Chen, S.Ludwig, Fuzzy-guided genetic algorithm applied to the web service selection problem, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2012, pp. 1–8.
13. W. Yang, C. Zhang, Y. Shao, Y. Shi, H. Li, M. Khan, F. Hussain, I. Khan, L.-J. Cui, H. He, et al., A hybrid particle swarm optimization algorithm for service selection problem in the cloud, International Journal of Grid and Distributed Computing 7 (4) (2014) 1–10.
14. M. Alrifai, T. Risse, W. Nejdl, A hybrid approach for efficient web service composition with end-to-end QoS constraints, ACM Transactions on the Web 6 (7).
15. X. Ye, B.A., Genetic algorithm based QoS-aware service compositions in cloud computing, Berlin, 2011, Ch. Database Systems for Advanced Applications, pp. 321–334.
16. O. Jula, S.E., A hybrid imperialist competitive gravitational attraction search algorithm to optimize cloud service composition, in: Memetic Computing (MC), IEEE Workshop, 2013, pp. 37–43.
17. Petcu, Frodo: A framework for open/distributed constraint optimization, http://frodo2.sourceforge.net/.
18. Yachir, Composition dynamique de services sensibles au contexte dans les systèmes intelligents ambiants, Ph.D. thesis, University USTHB, Algeria and University UPEC, France (2013).
19. P. Ngatchou, A. Zarei, M. El-Sharkawi, Pareto multi objective optimization, in: Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on, 2005, pp. 84–91. doi:10.1109/ISAP.2005.1599245.
20. Extensible messaging and presence protocol xmpp: http://en.wikipedia.org/wiki/xmpp.