HQPSO Algorithm for Improving QoS in Web Services Selection and Composition

D.Palanikkumar
Asst Professor
Department of Computer Science and Engineering,
Anna University of Technology Coimbatore, INDIA

N. Thillaiarasu
PG Scholar
Department of Computer Science and Engineering,
Anna University of Technology Coimbatore, INDIA

Abstract
Web services composition is the latest software progress pattern and its main point to accomplish service-oriented computing presently. For gathering the QoS requirements of consumers, this paper presents QoS calculation rules and the formal standing to the services composition as the combinatorial optimization problem. In order to solve services composition, a new algorithm based on PSO algorithm is proposed, which is called Hybrid QPSO algorithm. The accuracy, viability, and usefulness of the algorithm are demonstrated using the experiments.

Keywords: web service, services composition, QoS, Hybrid QPSO algorithm

I.INTRODUCTION

During service composition, selection of appropriate services for composition is the chief task. Many sophisticated tools are required for a web consumer to search for the best service that satisfies his needs. Several web services may exist that provide the same functionality. In such a case, Quality of Service (QoS) is the decisive factor in distinguishing the functionally similar services. An efficient method is PSO-based service selection which can be performed by considering user's functional and QoS constraints. Composing atomic web services to provide the value-added service, i.e., web services composition, attracts widespread concern, because more and more enterprises and organizations provide their business in the form of web service. Web services composition is not only the difficult and key point to implement the service-oriented architecture, but also a new software development paradigm in the area of software engineering. This paper presents a hybrid QPSO algorithm to solve service selection and composition with QoS constraints.

II.RELATED WORKS

The QoS property of services composition has accumulated a quantity of precious results, such as in literature [1, 2, 3, 4] and so on. A self-healing approach for web service composition proposed in [1]. It is an integration of backing up in selection and reselecting in execution, and a way of performance prediction. A QoS prediction approach proposed in [2], has complete similarity mining among consumers and QoS data and then predict the QoS of the unused web services from other consumers' experiences. Few researchers proposed an approach to compose services automatically based on backward tree, which composes services for a user all the way through three steps: 1) builds a complete backward tree on-line; 2) searches for optimal valid generation sources; 3) composes generation path [3]. A greedy algorithm to arrange all web service calls of a query into a pipelined execution plan, which be capable of exploit parallelism among web services to minimize the query's total running time [4].

III.PROCEDURE RULES OF SERVICES COMPOSITION

A service composition is an aggregative service without restriction to the granularity. It comprises of one or more services, achieves a certain business logic functions, and finishes after a inadequate-pace procedure. The defined control structure, combined with the initial definition, we explain the services composition using BNF (Backus Normal Form) as follows:

SC=WS|SC©SC|SC*SC|SC◊SC|SC||SCμSC

1. WS1©WS2: QoSsc=QoS (ws_1©ws_2) = (T_1+T_2, $C1+C_2$, min {A_1, A_2})
2. WS1*WS2: QoSsc=QoS (ws_1*ws_2) = {A_1, A_2}) or {QoSws_1, QoSws_2}, QoSsc
3. WS_1◊WS2:QoS_{sc}=QoS(ws_1◊ws_2)=(min{T_1,T_2},min {C_1,C_2},max{A_1,A_2})
4. WS_1||WS2:QoS_{sc}=QoS(ws_1||ws_2)=(max{T_1,T_2},C_1+ C_2,min{A_1,A_2}) μWS_1: If the number of iteration is K,
5. QoS_{sc}=$QoS_{(\mu ws1)}$$QoS_{ws1}$©$QoS_{ws1©.........©}$$QoS_{ws1}$ =(kT_1,Kc_1,A_1)

IV.SERVICES COMPOSITION MODEL

A .Services composition Definition

Similar Services Composition: With the same function, services composition has the different attributes of quality called Similar Services Composition. Figure 1 is a construction diagram of travel services composition. In the construction diagram, SFStart and PTStart are the respective start state of the flow and sub-flow, SFEnd and PTEnd be the respective end state of flow and sub-flow, which be supplemented for the reliability of the services composition. WS1 is the Book Flight service, WS2 is the Book Car service, WS3 is the Book Hotel a service, WS4 is the Book Flight service, WS5 is the BookCar service, WS6 is the Book Hotel service; servicesWS11…WSn1,WS21…WSn2,……,WS61…WSn6 are the same service; services WS1,WS2,…,WS6 are selected form their similar services respectively. The services composition travel can be expressed as:

$SC=WS_{1j}1©(WS_{2j2}||(WS_{3j3}◊WS_{4j4}◊WS_{5j5}))©WS_{6j6}(ji=1,2,\ldots\ldots n_{i,}, i=1,2,3,4,5,6)$

The reason of a huge number of parallel services, services composition can be constructed using a variety of different patterns, in which they are many similar services compositions.

B. Mathematical Model of Services Composition

Services composition may not have the highest QoS properties at the same time, so their relations need to be coordinated by a trade-off factor. Let us assume that the number of services which services composition needs to achieve goal is m, and has the number of similar services for each WSi is ni. Services composition is selected for the similar services compositions which the number is $\prod_{i=1}^{m} n$, and it have to satisfy the following condition, $QoSsc=\max\{QoS_{sc1},QoS_{sc2},\ldots..QoS_{sck}\}$, $K=\prod_{i=1}^{m} n$. $QoS_{sc1},QoS_{sc2},\ldots..QoS_{sck}$ can be completed respectively according to operation rules of services composition. Conversely, as QoS is a set of attributes, the maximum values QoS_{SC} cannot be obtained directly. We have to do the following transformations. We argue that QoS_{SC} cannot be obtained by comparison of their values which is the sum of each attribute, there are several reasons for this: 1) QoS is a tuple, not a specific numerical value; 2) Four attribute values have different meaning and the extent unit is different; 3) The four QoS attribute values may be a difference of several orders of magnitude, and the simple sum of them may result in loss of constraint function of the small orders of magnitude attribute; 4) For each value of the four attributes is not larger on behalf of the higher QoS, for instance the T and C are precisely the smaller on behalf of the higher QoS. We make the following normalization of the QoS to transform it into a as good as value:

$QoS=\frac{Aj-E(Aj)}{\Delta(Aj)}-\frac{Tj-E(Tj)}{\Delta(Tj)}-\frac{Cj-E(Cj)}{\Delta(Cj)},(1\leq j\leq k)$,Where E is the mathematical expectation, Δ is the standard deviation. In this way, the larger value of QoS'SFj implies the higher quality of service, and condition is expressed as $QoS'_{SC}=\max\{QoS'_{sc1}, QoS'sc2_{,\ldots\ldots,} QoS'_{sck}\}$.

V.SOLUTION TO THE SERVICES COMPOSITION

Services composition is a combinatorial optimization problem, and the solution to the problem is a complex process. HQPSO algorithm is called the general problem solving method, but it is a time-consuming and very low efficiency algorithm for solving services composition. Using the backtracking algorithm to solve services composition needs to traverse the complete solution space tree, and the solution space tree of services composition is very large, which has leaf nodes, there are paths have to be traversed. PSO gets the answer by virtue of a series of the optimal choices at the current state, and it is with less time but more efficient. Using it to solve the services composition needs to local optimal choices in the solution space tree of services composition, and the choices are not necessarily optimal path for the global, as calculating for each QoS value of atomic service may leads to lose some data and information. For these reasons, we try to improve the QPSO (Quantum-behaved Particle Swarm Optimization) algorithm to solve services composition with the optimal global solution.

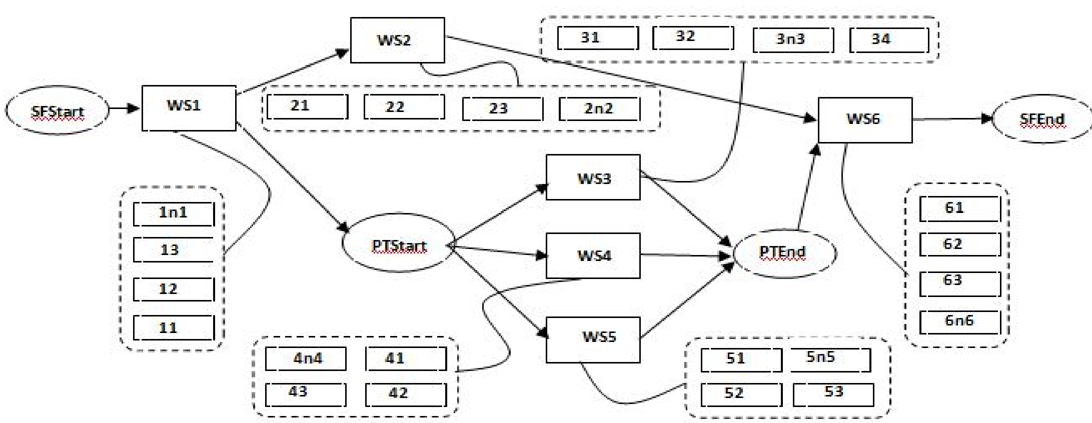

Figure 1.Composition Diagram of Services Composition Travel

A.QPSO ALGORITHM

QPSO Algorithm [5] is a novel PSO algorithm model, which is presented from the perspective of quantum mechanics. QPSO algorithm is the search strategy changes of PSO algorithm, which based on DELTA potential well and the particles having the quantum behavior. QPSO algorithm not only has less number of parameters, but also the search capability of algorithm is superior to PSO algorithm. It is in accordance with the following evolution equation of initialization and updating the particle's velocity and location: mbest=$1/M\sum_{i=1}^{M} pi = (1/M\sum_{i=1}^{M} pi\,1, 1/M\sum_{i=1}^{M} pi2, \ldots\ldots, 1/M\sum_{i=1}^{M} pid)$, $P_{id}=\phi * P_{id}+(|mbest - xid|\, 1-\phi) * P_{gd}$, ϕ=random(0,1) $X_{id}=P_{id}+\beta * \left|\left(\frac{1}{u}\right)\right|$,u=random(0,1) Where M is no of particles, mbest is mean best position among the particles ,pid is the random point of P_{id} and P_{gd} , β is exception and contraction coefficient.

B. Solving services composition with HQPSO algorithm

QPSO algorithm with the formula cannot meet the requirements for solving services composition because of the solution vector specification; the original formulas need to give the new meaning of computing. When introduce the cross operation and mutation operation into the QPSO algorithm: $P_{id}=\phi \Delta Pid \varDelta (1-\phi)\Delta Pgd$, ϕ=random(0,1) formula (1) X_{id}=Pid$\varDelta\beta\Delta\left(\left|\ln\left(\frac{1}{u}\right)\right|\Delta|mbest\varDelta xid|\right)$,=random (0,1) formula (2) Where operator Δ is the order cross operation, operator $\varDelta$ is the order mutation operation, q is the number of sequence element, $\alpha\Delta S_q$ indicate the position of element is$|\alpha * q|$ +1.in formula (2) non factor of P_{id} , but we required form last position of the sequence $P_{id,}$

Algorithm steps

Step1: Setup the number of particles Np, the provisions of iterations Num, randomly generated initial solutions Np, which is the initialization of services composition SF_0.

Step2: Calculate the fitness value of each particle based on the current positions.

Step3: Supposing that the current fitness value of particle is the local best fitness value, and the current position is the local best position Pl_{best}, using the solving formula for services composition to calculate the global best fitness value $Gf_{best}(QoS_{SC})$ and the global best position $Gl_{best}(SC)$.

Step4: Calculate the value of mbest using round method according to the formula (1).

Step5: For each particle, calculate the new positions according to the formula (1) and (2) that are the formation of the new services compositions.

Step6: If the calculations do not achieve the maximum number of iterations, repeat Step2 to Step5; else return Step7.

Step7: Output the global best fitness value (QoS_{SC}) and the global best positions (SC).

The time complexity of solving services composition with HQPSO algorithm can be predictable as,O(Num *Np*$\sum_{i=1}^{m} ni$).

VI. EXPERIMENTS

Experimental results has been taken for 100M LAN communication system, network topology is a star schema with seven nodes, each node is a single CPU (Intel Pentium 4 2.8GHz, Memory 2GB) structure, and operating system is Windows2003 Server.ASP.NET and MySQL DataBase. If the property QoSj of services composition is obtained by algorithm, $\lambda = 1 - |(QoSsc - QoSj)/QoSsc|$ λ is called success ratio of solving services composition with HQPSO algorithm, in this web logic, ratio large than .83 to be considered to successfully solve the services composition. As shown in Table 1 and Figure 2 and 3, we compare the success ratio and execution time PSO and HQPSO algorithm to solve services composition.

Services	PSO Time(ms)	HQPSO Time(ms)
1	260	50
2	340	80
3	400	90.54
4	245	48
5	500.72	150.52
6	380	85

Table 1.Comparation of two algorithms

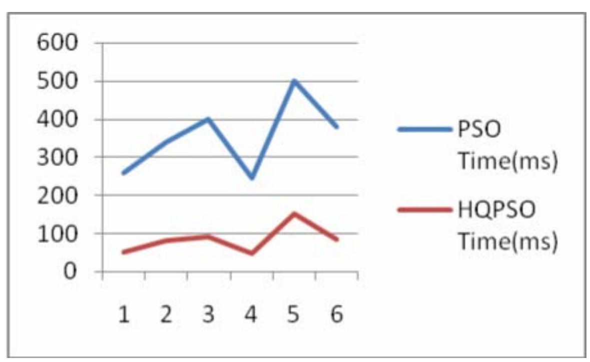

Figure 2.Performance compilation of PSO and HQPSO (Graph)

The result of the experiments conducted for different selection of services is shown in figure2 in terms of time measured in milliseconds. It is noted that the time required for service selection at any threshold level and QoS factors selected by using HQPSO algorithm is less, when compared to the PSO.

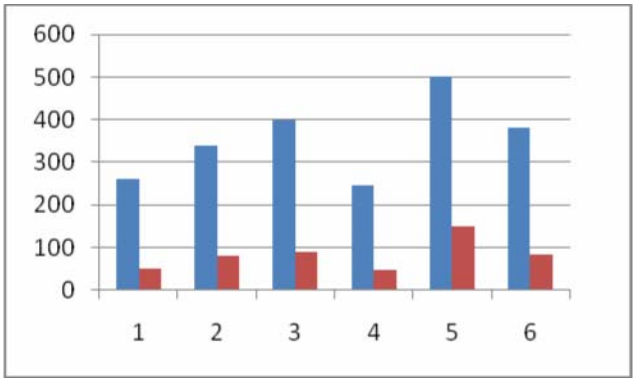

Figure 3.Comparision of two Algorithms (Histogram)

VII. CONCLUSION

Here we analyze the QoS calculation rules, according to the BNF of services composition, and dignified the services selection and composition as the combinatorial optimization problem. Then we presented a new QPSO algorithm to solve the services composition, using the viability and competence of the method. In the future the work may be extended by using parallelized HQPSO algorithm to improve the speed of solving services composition. It can also be aspired to code the adaptive method or stretching technical into HQPSO algorithm to improve the success ratio of solving services selection and composition.

References

[1]. Yn dai, Lei Yang, and Bin Zhang. QoS-Driven Self-Healing Web Service Composition Based on Performance Prediction. Journal of Computer Science and Technology, 2009, 24(2): 250-261

[2]. SHAO Ling-Shuang, ZHOU Li, ZHAO Jun-Feng, XIE Bing, and MEI Hong. Web Service QoS Prediction Approach. Journal of Software, 2009, 20(8): 2062-2073.

[3]. DENG Shui-Guang, WU Jian, LI Ying, WU Zhao-Hui. Automatic Web Service Composition Based on Backward Tree. Journal of software, 2007, 18(8):1896-1910

[4]. XU Shu-hua, JIANG Wen, HUANG Zhi-+gang. Query Optimization Plan of Web Services Based on Greedy Algorithm.ComputerEngineering, 2008, 34(5):69-70.

[5]. SUN J, XU WB. Particle Swarm Optimization with Particles Having Quantum Behavior [A].IEEECongress.EvolutionaryComputation[C], 2004.

AUTHORS PROFILE

Palanikkumar Durai Thirunavukkarasu is an Assistant Professor in Anna University of Technology, Coimbatore. He has received his Master's degree from Government College of Engineering, Tirunelveli. Submitted Synopsis of his PhD programme in the area of QoS in Web Services.His research interest includes QoS in web services, cloud computing, business intelligence and datawarehousing/mining.
Email ID: palanikkumard@gmail.com

Thillaiarasu Nadesan received the Bachelor Degree in Computer science and Engineering from Selvam College of Technology, affiliated to Anna University Chennai, India in 2010. Currently pursuing Masters degree in Software Engineering at the Anna University of Technology, Coimbatore, India.
Email ID: thillai888@gmail.com