 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Ensuring a Trust for the Selection of Web Services Based on QoS with Optimization Algorithms

[1]E.S. Shamila, [2]Dr. V.Ramachandran
[1]Research Scholar ,Faculty of CSE, Sathyabama University, Chennai, India
[2]Director, National Institute of Technology, Nagaland, India

Abstract
There are number of similar functional Web services are available. Trusting to find the most appropriate services from the similar functional available services is an issue. Web service composition is a new software development paradigm, and can done service oriented computing successfully. Web services composition problem can solve with optimization algorithms with QoS. The preference of various QoS of compound service is users wishes. We have proposed the services composition algorithm based on quality of services for various optimization algorithms. In this paper, to identify the best web service, the various optimization algorithms such as PSO algorithm, Cuckoo search and Cuckoo modified algorithm, Gravitational search algorithm and Intelligent water drop algorithm are used and compared. The experimental comparison result shows the best algorithm for web service composition is based on the access rate, speed of the system. This paper presents the QoS calculation of non functional requirements such as availability, response time, reliability and access rate.

Keywords
QoS, Particle Swarm Optimization, Cuckoo Search, Gravitational Search, Web Services

I. Introduction

A Web service is a programmable Web application which describes a collection of operations that are network accessible through XML messaging. With the increasing number of functionally similar Web services, finding the most suitable Web service for a particular task or set of task is a demanding issue in the area of dynamic Web service composition. Quality of service (QoS) is one of the most important non-functional criteria for selection of an appropriate Web service for a particular task. QoS for web services allows consumers to have confidence in the use of services by aiming to experience good service performance. It is difficult to choose services from service registries for service consumers, which contain hundreds of similar web services, given that the selection is only based on functional properties. Trust establishment for selecting Web Services by evaluating QoS Parameters with optimization algorithms. Web Service Selection problem is transformed into a multiobjects optimization problem with constrained: four attributes of QoS are treated as objects, that is, Access rate, less response time, higher availability and reliability. The results are compared with various optimization algorithms and the experimental comparison result shows the best algorithm for web service composition is based on the access rate, speed of the system.

This paper is structured as follows:

Related work for optimization techniques has been focused in section II. In Section III, proposed runtime QoS parameters are discussed to evaluate different nonfunctional parameters at runtime. In Section IV optimization techniques are described. Section V provides the proposed various optimization algorithms for evaluation of QOS non functional parameters. Section VI provides an example for service selection algorithms based on optimization. Section VII provides the experimental evaluation of proposed parameters. Section VIII provides the conclusion.

II. Related Work

The paper [1] explains about composition of web services by two various algorithms: Evolutionary and Non- Evolutionary Algorithms.The evolutionary algorithms find optimal solution only when the business processes are complex and distribute the service candidates to obtain best results. The non-evolutionary algorithms converge much faster but it produces efficient result under small scale environment and candidates are limited. The paper [2] enables user experience and involvement, simplifies service implementation and optimizes the service lifecycle. But, it doesn't focus on efficient service discovery and selection algorithms and lack of distributed storage mechanisms. So, we are in necessity to enhance access rate, reliability, and availability by fully automating the search by using optimization techniques.

A third party broker works as a third reliable Web service coordinating on measurement, monitoring and updating QoS database in time to guarantee that the discovered services based on QoS are reliable and trustable[4-5][7-8]. A QoS model includes response time, price, reliability and throughput is proposed [3,5, 7]. The paper [6] explains semantic model checking algorithm which is sound and complete and it is a basic tool for web service selection, validation and composition.

III. Proposed Non-Functional Parameters of QoS

Parameters of Qos Selecting the best services based on QoS for simple and complex task requires continuously updating the published QoS information. QoS information can be updated only by monitoring the QoS parameters performed by trusted third party brokers. To perform monitoring and evaluation of QoS parameters associated with Web services, Access Rate, Response Time, Reliability and Availability are used with optimization algorithms.

A. Access Rate(Ar):

Access rate [9] is an accessibility parameter that requires continuous monitoring of Web services by broker to provide updated QoS information. Access rate is directly related with the availability from the host location. Access rate can be defined as the rate of total number of Web service request (Ar) requested by the service consumer through broker interface.

There are three types of Access rate

- Success access rate i.e., (S(Ar))
- Failure access rate i.e., (F(Ar)) and
- Bounce access rate i.e., (B(Ar))

Access rate is denoted by Ar and it can be calculated as follows:

$$Ar = \Sigma (S(Ar)) + \Sigma(F(Ar)) + \Sigma(B(Ar))$$

B. Response Time (WSresp_time):

Response time is the total time duration spent request and response for a particular Web service between from the side of service

consumer broker and service provider.
WSresp_time=(Tcons_req+Tbro_req)(Tbro_res+Tpro_res)
where Tcons_req is consumer request, Tbro_req is broker request, Tbro_res is broker response and Tpro_res is provider response

C. Availability (WSavail):
Availability deals with the capability to perform the task and not being able to perform the task which is referred as uptime and downtime of web services.

D. Reliability (WSreliable):
The reliability of service(WSR) represents the probability that a request is correctly responded within the specific of time and is presented by the success rate of invoked service.

IV. Optimization Techniques
Optimization is the act of obtaining the best result under given circumstances. Optimization can be defined as the process of finding the conditions that give the maximum or minimum of a function. The purpose of optimization is to choose the best one among the many that is available. In some situations, there may be more than one criterion to be satisfied simultaneously. An optimization problem involving multiple objective functions is known as a multiobjective programming problem.
With multiple objectives there arises a possibility of conflict, and one simple way to handle the problem is to construct an overall objective function as a linear combination of the conflicting multiple objective functions.
Thus, if f1 (X) and f2 (X) denote two objective functions, construct a new (overall) objective function for optimization as:
where a1 and a2 are constants whose values indicate the relative importance of one objective function to the other.
Trust establishment in selecting web services is mainly achieved in the following way :

- There are hundreds of similar web services are available.
- Based on the multiobjective programming problem of optimization we have to to construct an overall objective function as a linear combination of the conflicting multiple objective functions.
- Web Service Selection problem is transformed into a multi-objects optimization problem with constrained: four attributes of QoS are treated as objects, that is, Access rate, less response time, higher availability and reliability

The Complete Successful Quality of Service (CSQoS) for each web service is calculated as
CSQoS= (WSresp_time + WSavail + WSreliable)/N
Where N is the total number of QoS parameters used for the evaluation of average score of all QoS parameters.

V. Proposed Various Optimization Algorithms For Evaluation of Qos Non Functional Parameters
To identify the best web service decomposition algorithm, the various optimization algorithms such as PSO algorithm, Cuckoo search and Cuckoo modified algorithm, Gravitational search algorithm and Intelligent water drop algorithm are used and compared.

A. PSO Algorithm
Service selection algorithm implementation steps based on particle swarm as following steps,
Select any N combinations randomly calculate the CSQoS as follows,
CSQoS= (WSresp_time + WSavail + WSreliable)/N
Step 1 : Set global fitness value as zero.
Step 2 :Randomly select N combination using random function. Calculate the CSQoSas above.
Step 3 :Take the first combination and calculate CSQoS. The best fit value has to assign global fitness value.
Step 4 :Take the second combination and calculate CSQoS. Check the second combination CSQoS with global fitness value.
If second combination best fit is less than
the global fitness value then
replace global fitness value with this second
combination CSQoS
else
keep the global fitness value as it is.
Step 5. Repeat the step 4 for all N combinations.
Step 6. Take the final global fitness value's combination is the best combination.

B. Cuckoo Search Algorithm
Service selection algorithm implementation steps based on cuckoo search is as follows, Select any n combinations randomly calculate the CSQoS as follows,
CSQoS= (WSresp_time + WSavail + WSreliable)/N
Step 1. Set global fitness value as zero.
Step 2. Randomly select n combination using random function. Calculate the CSQoS as above.
Step 3. Take the first combination and calculate CSQoS. The CSQoS value has to assign global fitness value.
Step 4. Take the second combination and calculate CSQoS. Check the second combination CSQoS with global fitness value.
Step 5. If second combination CSQoS is less than the global fitness value then replace global fitness value with this second combination CSQoS
else
keep the global fitness value as it is.
Step 6. Repeat the step 4 for all n combinations.
Step 7. Take the final global fitness value's combination is the best combination.
Step 8. Compare the global fitness value with all randomly selected combinations CSQoS values. The combinations which is greater than the global fitness value should be removed from the list considering they are worst fitness. So that in future we can avoid the worst combinations.

C. Modified Cuckoo Search Algorithm
Service selection algorithm implementation steps based on Modified Cuckoo Search as following steps,
Both, original, and modified code use random step sizes. Compared to the original code, we use different function set for calculating this step size.In the original code, step size is calculated using following code expression:
r*nests [permute1 [i]][j]- nests [permute2 [i]][j] (1)

- where r is random number in [0,1] range, nests is matrix which contains candidate solutions along with their parameters, permute1 and permute2 are different rows permutation functions applied on nests matrix.

In order to calculate the step size, instead of Equation (1) , we used:
r*nests [sorted [i]][j] - nests [permute [i]][j] (2)
This method keeps the selection pressure (the degree to which highly fit solutions are selected) towards better solutions and algorithm should achieve better results.

 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

D. Intelligent Water Drop algorithm

Service selection algorithm based on Intelligent Water Drop is as follows,

The IWD algorithm employs a number of IWDs to find the optimal solutions to a given problem. The problem is represented by a graph (N, E) with the node set N and edge set E.This graph is the environment for the IWDs and the IWDs flow on the edges of the graph. Each IWD begins constructing its solution gradually by traveling between the nodes of the graph along the edges until the IWD finally completes its solution denoted by T IWD. Each solution T IWD is represented by the edges that the IWD has visited. One iteration of the IWD algorithm is finished when all IWDs complete their solutions. After each iteration, the iterationbest solution T IB is found. The iterationbased solution T IB is the best solution based on a quality function among all solutions obtained by the IWDs in the current iteration. T IB is used to update the total-best solution TTB . The total-best solution TTB is the best solution since the beginning of the IWD algorithm, which has been found in all iterations.

E. Gravitational Search Algorithm (GSA)

Service selection algorithm implementation steps based on Gravitational Search Algorithm (GSA) as following steps, Gravitational Search Algorithm (GSA) is a new optimization algorithm based on the law of gravity proposed. The goal function is defined at two different equations. The fitness of each service is calculated. Since each mass is a sequel of web services, for evaluating each mass, we sum fitness of involved web services in the sequel.

VI. An Example For Service Selection Algorithms Based On Optimization

Consider the proposed optimization based service selection algorithms by an example of a Airline Reservation System.

There will be three service providers. Each are linked together.

- Ticket Reservation Service
- Hotel (Room Booking) Service
- Travels (Car) Service

When client booking flight ticket from source to destination(eg. From chennai to delhi), the reservation service provider should list out all the available flights along with available hotels provider and travels provider in destination.

Eg.

Flights

Flight1 Flight2 Flight3 Flight4 etc.

Hotels

Hotel1 Hotel2 Hotel3 Hotel4 Hotel5 Hotel6 etc.

Travels

Car1 Car2 Car3 Car4 Car5 Car6 etc.

When the client is requesting a reservation service it should provide all possible combinations as follows,

- Flight1 Hotel1 Car1--- with all tariff
- Flight1 Hotel1 Car2--- with all tariff
- Flight1 Hotel1 Car3--- with all tariff
- Flight1 Hotel1 Car4--- with all tariff
- Flight1 Hotel1 Car5--- with all tariff
- Flight1 Hotel1 Car6--- with all tariff
- Flight1 Hotel1 Car7--- with all tariff
- Flight1 Hotel2 Car1--- with all tariff
- Flight1 Hotel2 Car2--- with all tariff
- Flight1 Hotel2 Car3--- with all tariff
- Flight1 Hotel2 Car4--- with all tariff
- Flight1 Hotel2 Car5--- with all tariff
- Flight1 Hotel2 Car6--- with all tariff
- Flight1 Hotel3 Car1--- with all tariff
- Flight1 Hotel3 Car2--- with all tariff
- Flight1 Hotel3 Car3--- with all tariff
- Flight1 Hotel3 Car4--- with all tariff
- Flight1 Hotel3 Car5--- with all tariff
- Flight1 Hotel3 Car6--- with all tariff

We have to give all possible combinations. Minimum of 64 combinations.

VII. Experimental Evaluation

The performance of QoS parameters can be analyzed through a set of functionally similar Web services stored into database. These Web services have been collected from different sources such as, servicerepository. com, xmethod.net and webservicex. net. The database WSdatabase for these Web services is created in MS Access, which includes number of Web services with their Web service id (WSID), Web service name (WSname) and URLs. These Web services were invoked several times for a period of week.

The access rate, availability, reliability and response_time can be calculated. Web services have stored in MS Access database. The value of QoS parameter can be evaluated by using these evaluated nonfunctional parameter values shown in Table 1.

Table 1. Web Services With Their Monitored Non-Functional Parameter Values

ws	url	Total	Failure	Success	Bounced_Rate	Success_Rate	Failure_Rate	Availability	Response_Time	Price
Jet Lite	http://localhost:1045/JetLite/Service.asmx	6	5	4	0.6	0.8	0.3333333	12	44 ms	463
Jet Lite	http://localhost:49176/JetLite/Service.asmx	7	7	5	0.5833333	0.7142857	0.368421	3	24 ms	291
Jet Lite	http://localhost:49176/JetLite/Service.asmx	8	6	6	0.5714286	1	0.3	9	37 ms	407
Jet Lite	http://localhost:49176/JetLite/Service.asmx	13	10	7	0.65	0.7	0.3333333	8	34 ms	379
Jet Lite	http://localhost:49176/JetLite/Service.asmx	14	9	8	0.6363636	0.8888889	0.2903226	11	41 ms	440
Spice Jet	http://localhost:49183/SpiceJet/Service.asmx	649	12	8	0.9878234	0.6666667	0.01793722	1	21 ms	263
Jet Airways	http://localhost:49164/JetAirways/Service.asmx	94	20	1	0.9894737	0.05	0.173913	8	34 ms	380
Jet Airways	http://localhost:49164/JetAirways/Service.asmx	113	19	2	0.9826087	0.1052632	0.141791	19	58 ms	586
King Fisher	http://localhost:49181/kf/Service.asmx	173	20	1	0.9942529	0.05	0.1030928	1	20 ms	253
Jet Lite	http://localhost:49176/JetLite/Service.asmx	35	9	14	0.7142857	1.555556	0.1551724	1	20 ms	251
Jet Lite	http://localhost:49176/JetLite/Service.asmx	36	8	15	0.7058824	1.875	0.1355932	13	45 ms	476
Jet Lite	http://localhost:49176/JetLite/Service.asmx	39	7	16	0.7090909	2.285714	0.1129032	19	58 ms	585
Jet Lite	http://localhost:49176/JetLite/Service.asmx	42	9	17	0.7118644	1.888889	0.1323529	15	50 ms	516
Jet Lite	http://localhost:49176/JetLite/Service.asmx	45	9	18	0.7142857	2	0.125	1	20 ms	258
Jet Lite	http://localhost:49176/JetLite/Service.asmx	48	8	19	0.7164179	2.375	0.1066667	12	43 ms	459
Jet Lite	http://localhost:49176/JetLite/Service.asmx	54	7	21	0.72	3	0.08536585	13	45 ms	471
Jet Airways	http://localhost:49164/JetAirways/Service.asmx	199	25	3	0.9851485	0.12	0.1101322	19	58 ms	586
Spice Jet	http://localhost:49184/SpiceJet/Service.asmx	1100	19	9	0.9918846	0.4736842	0.01684397	17	55 ms	557

A Web interface is developed as shown in Fig. 1, to implement the proposed QoS parameters on Windows 2007 server platform using Microsoft Visual Studio .NET development environment and ASP.NET, VB.NET as a programming language. The interface helps to evaluate the values of above parameters such as access rate, availability, reliability, response_time for Airline Reservation System.

With the help of interface as shown in fig. 1, the broker can evaluate the value of CSQoS parameter by aggregating different non-functional parameters value.

Through the value of CSQoS parameter, the broker can rank the Web service before publishing on host location from where it is accessible by the service consumers. The service consumer can select the best services for discovery and composition of services with the help of this CSQoS score.

The experimental results shown in fig. 2, optimal solution of the execution time spend for single users and multi-users. The experimental comparison result shows the best algorithm for web service composition is based on the access rate, speed of the system. Here cuckoo modify shows the best performance. But it differs based on the access rate, speed of the system.

Fig. 1: Evaluation of Non-Functional Parameters

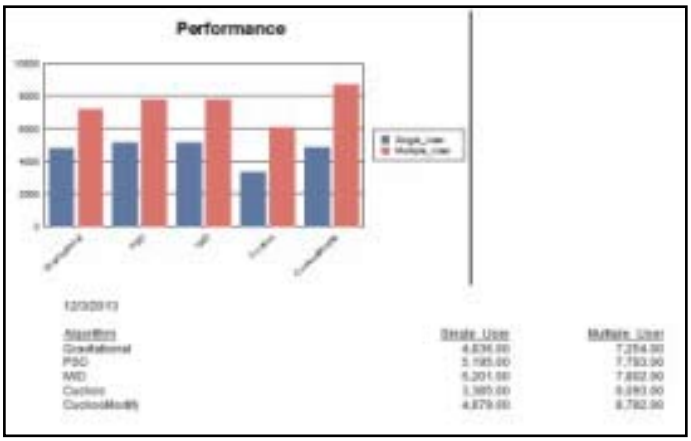

Fig. 2: Optimal Solution of the Execution Time Spend for Single Users and Multi-Users

VIII. Conclusion

Evaluation of QoS parameters of a complex service can be achieved by computation of QoS of every component service for providing trustable and best Web service. In this paper, to identify the best web service, the various optimization algorithms such as PSO algorithm, Cuckoo search and Cuckoo modified algorithm, Gravitational search algorithm and Intelligent water drop algorithm are used and compared. The experimental comparison result shows the best algorithm for web service composition is based on the access rate, speed of the system. This paper presents the QoS calculation of non functional requirements such as availability, response time, reliability and access rate.

References

[1] E.Pejman, Y.Rastegari, P.Majlesi Esfahani, A.Salajegheh, "Web service Composition Methods: a survey", Proceedings of International conference of engineers and computer scientists", March 2012.

[2] Cuiting Huang, Gyu myoung Lee, Noel Crespi , "A Semantic Enhanced Service Exposure Model For A Converged Service Environment", IEEE Communications Magazine, March 2012.

[3] T. Rajendran, Dr. P. Balasubramanie,"An Efficient Architecture for Agent based Dynamic Web Services Discovery with QoS", Journal of Theoretical and Applied Information Technology (JATIT), Pakistan, Vol. 15, No. 2, May 2010.

[4] Gang Ye,"A QoS Aware Model for Web Service Discovery", First International Workshop on Education, Technology and Computer Science (ETCS '09), IEEE Computer Society, Wuhan, Vol. 3, pp. 740-744, March 2009.

[5] T. Rajendran, Dr. P. Balasubramanie,"An Efficient WS-QoS Broker based Architecture for Web Service Selection", International Journal of Computer Applications (IJCA), Foundation of Computer Science, USA, Vol. 1, No. 9, 2010.

[6] Ivan Di Pietro, Francesco Pagliarecci, Luca Spalazzi, "Model Checking Semantically Annotated Services", IEEE Transactions on software engineering, Vol. 38, No. 3, may/ june 2012.

[7] International Journal of Web Services Practices (IJWSP), Korea, Vol. 5,No. 1, pp. 32-40, 2010.

[8] T. Rejendran,"Flexible and Intelligent Architecture for Quality based Web Service Discovery with an Agent based Approach", International Conference on Communication and Computational Intelligence (INCOCCI), Erode, pp. 617- 622, 27-29 December, 2010.

[9] Maya Rathore, Ugrasen Suman,"Evaluating QoS Parameters for Ranking Web Service", Proceedings of International Advance Computing Conference (IACC), Feb 2013.