

## Intelligent Management of Remote Facilities and Quality of Cloud Services

T.A.Deepak Kumar<sup>1</sup>, Dr. G. Sumathi<sup>2</sup>

<sup>1</sup>Department of Computer Science

<sup>2</sup>Department of Information Technology

Sri Venkateswara College of Engineering, Sriperumbudur – 602105,India

### Abstract

*Cloud Computing is a new and vast area that is impacting new business model and is heavily using the network as a platform. As the number of cloud services grow , it will be difficult to find the appropriate Cloud service suitable for our needs. We combine the service selection algorithm and ranking mechanism along with particle swarm optimization algorithm to find the appropriate service matching our needs. We are using Quality of Service as our main criteria. We consider travelling as our subject and take availability, reliability and price as our measure of Quality of Service . We use multi-objective optimization strategies to find the best Cloud service.*

**Key Words:** *Cloud Computing, Service Ranking, Service Selection, Particle Swarm Optimization.*

### 1. Introduction

Cloud computing is the next stage in evolution of the Internet. The cloud in cloud computing provides the means through which everything - from computing power to computing infrastructure, applications, business processes to personal collaboration - can be delivered to us as a service wherever and whenever we need. In this paper, we use an efficient approach to implement optimal web service selection and ranking for fulfilling service requesters functional and non-functional requirements[9]. The given work is distinguished from other related research in the following three aspects. a) A web service description model for describing web services where non-functional attributes are taken into account. b) An overall framework of service selection and ranking with QoS is used based on previous given description model. c) We respectively concentrate on a specific service selection algorithm, a service ranking algorithm, and quality updating mechanism.

### 2. Related Works

The work was proposed in the reference [3], which presented a model of reputation-enhanced QoS-based web service discovery that combines an augmented UDDI registry to publish the QoS information and a reputation manager to assign reputation scores to services. However, it only described an abstract service matchmaking, ranking and selection algorithm. Moreover, they failed to give an efficient metrics method for QoS computation, which was only evaluated by the dominant QoS attribute. In order to enable quality-driven web service selection, the authors in [4] proposed a QoS computation model by implementation and experimentation with a QoS registry in a hypothetical phone service provisioning. Unfortunately, as a result of their measurement way of QoS values normalization, it is very

difficult to make a uniform evaluation for all quality criteria because their QoS metrics values are not limited in a definite range. Therefore, it will bring about a problem that a quality attribute even has a higher weight, while its internal impact is decreased by its smaller QoS value.

### 3. Proposed Framework

In our proposed work, we combine the particle swarm optimization along with service selection and service ranking algorithm as shown in Figure1. Based on the user request we search the cloud service repository. If the function matches, we perform service selection and ranking along with the particle swarm optimization. We then update the QoS database, based on the results we obtain. We derive results from both the system and take a weighted average to obtain the needed results.

#### 3.1 Cloud Service Repository

The cloud service repository contains the cloud services functional and non-functional attributes of the cloud services. The Cloud service providers provide details about them to the cloud service repository.

#### 3.2 QoS Database

The QoS database contains the updated information about cloud service providers which is obtained from the feedback provided by the cloud users.

### 4. Cloud Service Description Model

This model is proposed in [9]. In order to facilitate providers to publish service information with QoS, it is necessary to model service description, as well as provides a mechanism for requesters to submit service requirements. An efficient cloud service description model has been given in [8]. However, it does not include QoS registry. We consider the service description model concerning QoS called *WSDM-Q*, which contains two parts of definitions: cloud service and service request.

#### Definition 4.1 Cloud Service.

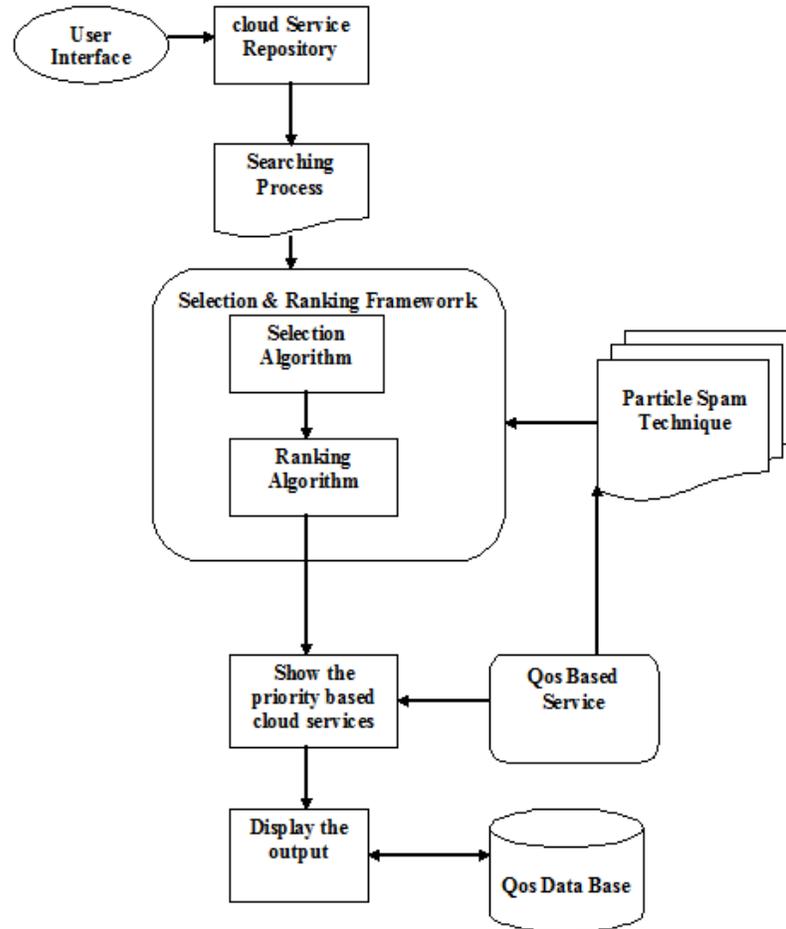
A cloud service in cloud service repository is defined as a five tuple:

$$ws = \{ServiceKey, wsName, wsDesp, QP, OprSet\} \quad (1)$$

- ✚ *ServiceKey* is the unique identifier;
- ✚ *wsName* represents cloud service name;
- ✚ *wsDesp* is service functional description;
- ✚ *QP* is published QoS information that is denoted as  $QP = QN\_QD$ . Where *QN* represents necessary quality criteria set for all web services and *QD* represents domain-specific quality criteria set for specific web services.
- ✚ *OprSet* is web operation set denoted as  $OprSet = \{opr1, opr2, \dots, oprs\}$ . Where each  $opri (1 \leq i \leq s)$  can be executed for a specific function task. Similarly, for the requirements of service requester, a corresponding service request description is given.

**Definition 4.2 Service Request.**

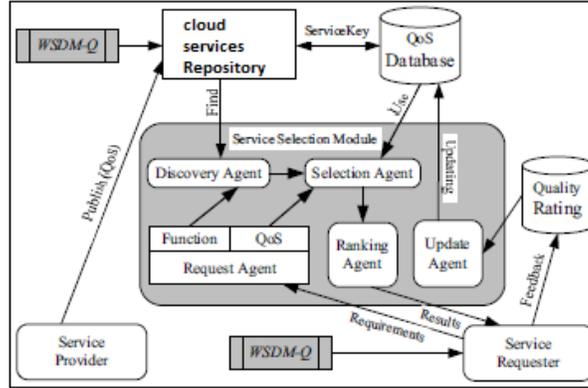
A service request is defined as a four tuple:  $sq = \{wsName, InSet, OutSet, QR\}$  Where,  $wsName$ ,  $InSet$  and  $OutSet$  have the same meaning as in *Definition 4.1* The difference is that these are the request information.  $QR$  includes necessary and domain-specific quality criteria set, which is defined as  $QR = QNUQD$  similar with the *Definition 4.1*. *WSDM-Q* is used to publish web services or submit service requirements in the following framework.



**Figure 1.Overall Architecture**

**5. Service Selection and Ranking Framework**

In the service selection framework, which extend the overall architecture to support service selection, ranking and quality updating, which consists of web services repository, QoS database, service selection module, quality rating database, service provider and service requester. The general framework for web service selection and ranking with QoS called *WSSR-Q* is shown in Figure 2. The kernel of the framework lies in the Service selection module, which involves five correlative agents as shown in Figure 2.



**Figure 2. The General Framework of Cloud Service Selection and Ranking with QoS**

## 6. Web Service Selection, Ranking and Updating

Some classic web service discovery algorithms have been proposed, such as in reference [2, 8]. Therefore, the task of discovering web services in term of requesters' functional requirements is performed by the existing service matchmaking engine.

### 6.1 Service Selection Algorithm

Some notations are given in the following subsections of this paper as it appears in Table 1:

#### Notations and Explanation

- $S$  - Web services repository
- $s_i$  - A web service,  $s_i \in S$
- $S_D$  - Discovered service set,  $S_D \subseteq S$
- $S_S$  - Selected service set,  $S_S \subseteq S_D$
- $S_R$  - Ranked service set  $S_R \subseteq S_S$
- $q_{ij}$  - QoS name at position  $j$  of  $s_i$
- $v_{ij}$  - Constraint value of  $q_{ij}$
- $Q_P^i$  - Published QoS information set of  $s_i$
- $Q_R$  - Submitted QoS requirement set
- $c_{ij}$  - A constraint relation at position  $j$  of  $s_i$
- $c_k$  - Request ternary relation at position  $k$

The description of Service selection algorithm with QoS (SSA-Q) is shown in Algorithm 1.

#### **Algorithm 1:**Service selection with QoS

**Input :**  $S_D$  and QoS requirement  $Q_R$

**Output :** Selected web service set  $S_S$

$S_S \leftarrow S_D$  ;

If  $Q_R = \text{NULL}$  then

Return  $S_S$  ;

Else if  $Q_N \neq \text{NULL}$  then { //  $Q_N \subseteq Q_R$

```

len ← QN.length;
SS ← SelectWithQoS (SD, QN,len,1); }
If SS ≠ NULL then {
  len ← QD.length; // QD ⊆ QR
  SS ← SelectWithQoS (SS, QD,len,0); }
Return SS ; }
Else
  Return NULL

```

### SelectWithQoS(S<sub>X</sub>, Q<sub>X</sub>, lenX, kindX)

```

Sres ← NULL // returned service set
For u ← 1 to SX.length do {
  QPu ← SX[u].QP ;
  counter ← 0;
  For w ← 1 to QX.length do {
    cw(qw,opw,vw) ← QX[w] ;
    cij(qij , opij ,vij ) ← findTerRel(QPu , qw);
    If(cij(qij , opij ,vij ) = NULL ∧ kindX=1) then break;
    Else If(cij(qij , opij ,vij ) = NULL ∧ kindX=0) then
      counter ← counter + 1;
    Else If (cij(qij , opij ,vij ) ≠ NULL) then {
      If(opw.equals('≤') ∧ vw ≥ vij ) then
        counter ← counter + 1;
      Else If (opw.equals('≥') ∧ vw ≤ vij ) then
        counter ← counter +1;
      Else break; }
    If ( counter = lenX) then // SX[u] judgement
      Sres . Append(SX[u]); }
}
Return Sres ;

```

## 6.2 Service Ranking Algorithm

**Algorithm 2:**Service ranking with QoS

**Input:**S<sub>S</sub>,Q<sub>R</sub> and quality criteria weight array W;

**Output :** Ranked web service set S<sub>R</sub> ;

**Step 1:** generate quality criteria matrix M<sub>S</sub>

```

MS ← NULL
For i ← 1 to r do {
  QPi ← SS[i].QP ;
  For j ← 1 to QR.length do {

```

```

    cj(qj, opj, vj) ← QR[j] ;
    cuw(quw, opuw, vuw) ← findTerRel(QPi, qj);
    If cuw(quw, opuw, vuw) ≠ NULL then
        MS[i,j] ← vuw ;
    Else MS[i,j] ← 0 ; }
}

```

## Step 2: Generate normalized quality criteria matrix M'<sub>S</sub>

```

M'S ← NULL ;
For j ← 1 to QR.length do {
    qmax ← Maxrk=1 {MS[k,j];
    qmin ← Minrk=1 {MS[k,j];
    For i ← 1 to r do {
        If QR[j].opj . Equals('≥') then

            M'S[i,j] ← SQRT( MS[i,j] - qmin ) / SQRT(qmax - qmin )
        Else If QR[j].opj . Equals('≤') then
            M'S[i,j] ← SQRT( qmax - MS[i,j] ) / SQRT(qmax - qmin ) }
    }
}

```

## Step 3: Calculate and rank each service's QoS value

```

For i ← 1 to r do {
    qsumi ← ∑k=1n (wk * M'S [i,k]);
    SR.rank(qsumi, si); }
Return SR;

```

## 7. Particle Swarm Optimization Algorithm for Service Selection Based on Quality of Service in Web Service Composition

### 7.1 Particle Swarm Optimization Algorithm

PSO is an algorithm proposed by Kennedy and Eberhart in 1995. As proposed in reference [10], Social behavior of organisms such as bird flocking and fish schooling motivated them to look into the effect of collaboration of species onto achieving their goals as a group. Years of study for the dynamics of bird flocking resulted in the possibilities of utilizing this behavior as an optimization tool. They named it as PSO. In a PSO system, multiple candidate solutions coexist and collaborate simultaneously. Each solution candidate, called a 'particle', flies in the problem search space (similar to the search process for food of a bird swarm) looking for the optimal position to land. A particle, as time passes through his quest, adjusts its position according to its own 'experience', as well as according to the experience of neighboring particles. Tracking and memorizing the best position encountered build particle's experience. For that reason, the PSO algorithm possesses a memory (i.e. every particle remembers the best position it reached during the past). PSO system combines local search methods (through self experience) with global search methods (through neighboring experience), attempting to balance exploration and exploitation. Two factors characterize a particle status on the search

space: its position and velocity. Kennedy and Eberhart explore several models to manipulate these factors to accurately resembles the dynamic of the social behavior of birds, before reaching to the following equations which achieve good performance on optimization problems :

From reference [11],

$$vid(t+1)=\omega \times vid(t)+c1 \times r1 \times (Pid(t)-xid(t))+c2 \times r2 \times (Pgd(t)-xid(t)) \quad \dots\dots\dots(1)$$

$$xid(t+1)=xid(t)+vid(t+1) \quad \dots\dots\dots(2)$$

In formula  $\omega$  is inertia factor,  $c1$  and  $c2$  are plus acceleration constant  $r1$  and  $r2$  are two range between 0 and 1 random function independent of each other.

### 7.2 Service Selection Algorithm based on Particle Swarm

We use the service selection algorithm in the example of a travel service. As given in [11], Firstly, for booking services and hotel services, then Tourist attractions service, at last formal a comprehensive set of travel program.

$$F_{ij} = (w_{respt} Q_{ij}^{respt} + w_{cost} Q_{ij}^{cost}) / (w_{ava} Q_{ij}^{ava} + w_{rel} Q_{ij}^{rel}) \quad \dots\dots(3)$$

where  $w_{respt}$ ,  $w_{cost}$ ,  $w_{ava}$ ,  $w_{rel}$  are corresponding weights of response time, cost, availability and reliability  $w_{respt} + w_{cost} = 1$  and  $w_{ava} + w_{rel} = 1$  and  $0 \leq w_{respt} \leq 1$ ,  $0 \leq w_{cost} \leq 1$ ,  $0 \leq w_{ava} \leq 1$ ,  $0 \leq w_{rel} \leq 1$

PSO algorithm is used to solve and gain Pareto optimal solution. Formula [3] is the utility function.

Web service selection problem can be described as the following global optimization problem.

$$\begin{cases} \text{Max} \sum_{i=1}^m \sum_{j \in WS_i} F_{ij} x_{ij} \\ \sum_{i=1}^m \sum_{j \in WS_i} Q_{ij} x_{ij} \leq Q_c \\ \sum_{j \in WS_i} x_{ij} = 1 \\ x_{ij} \in \{0,1\} (i=1,2,\dots,m; j \in WS_i) \end{cases} \quad (4) \text{ ,where if service}$$

$WS_{ij}$  selected then  $x_{ij}=1$ , else  $x_{ij}=0$ .  $Q_c$  is the sum of bound of quality of service.

From reference[11], The particle swarm optimization algorithm is given as following steps.

**Step1** Initialize the size of particle swarm, inertia weight, acceleration factor and the maximum allowable number of iterations or the smallest error of the fitness requirements, particle initial position and initial velocity.

**Step2** Evaluate the initial values of the individual particles and store it in  $P_{id}$ , The initial best values of groups are stored in the  $P_{gd}$ , repeat Step3-Step4 until termination condition met, iteration stopped, the output solution set obtained.

**Step3** Using formula (1) and formula (2) update each particle velocity and position.

**Step4** With objective function, re-evaluate each particle fitness value, If the individual particle fitness is better than  $P_{id}$  fitness, then in  $P_{id}$  is set to a new value. If the group particle fitness is better than  $P_{gd}$ , then in  $P_{gd}$  is set to a new value.

## 8. Simulation Results

We consider the services such as travel, stay and food. We consider service providers who provide the above services such as southernrailways, private travels etc., The services are first selected. After which the quality criteria matrix is formed. Then we normalize it to get normalized quality criteria matrix. After which we rank the systems based on the points, which are obtained as a result of ranking.

QoS	Values	Weight
Price $\leq$	5000	0.35
Exec.Duration $\leq$	100	0.05
Res.Time $\leq$	97	0.05
Availability $\geq$	0.20	0.30
Comp.Rate $\geq$	0.15	0.10
Penalty Rate $\leq$	0.90	0.15

Figure 3. Selection Criteria

**Selected Services**

kpntravels  
southernrailways  
airindia  
rajcars  
smcars

Figure 4. Selected Services

Quality Criteria Matrix					
Price	Exec.Dur	Resp.Time	Availability	Comp.Rate	Penalty Rate
500.0	60.0	70.0	0.7	0.7	0.5
400.0	60.0	70.0	0.75	0.75	0.4
1500.0	65.0	75.0	0.55	0.85	0.25
1200.0	60.0	75.0	0.65	0.8	0.35
1500.0	70.0	80.0	0.65	0.75	0.25
Normalized Quality Criteria Matrix					
Price	Exec.Dur	Resp.Time	Availability	Comp.Rate	Penalty Rate
0.95346254	1.0	1.0	0.8660254	0.0	0.0
1.0	1.0	1.0	1.0	0.57735026	0.6324555
0.0	0.70710677	0.70710677	0.0	1.0	1.0
0.522233	1.0	0.70710677	0.70710665	0.81649655	0.7745967
0.0	0.0	0.0	0.70710665	0.57735026	1.0

Figure 5. Quality Criteria Matrix

<b>Rank 1 southernrailways Points = 0.903</b>
<b>Rank 2 kpntravels Points = 0.694</b>
<b>Rank 3 rajcars Points = 0.678</b>
<b>Rank 4 smcars Points = 0.420</b>
<b>Rank 5 airindia Points = 0.321</b>

Figure 6. Ranking

Based on the input criteria, the appropriate service is selected and then ranked. Then we apply particle swarm optimization to further enhance the accuracy of the results.

## 9. Conclusion

In this paper, we have discussed an approach on efficiently selecting web services with similar functionalities. We first, gave an overall framework for service selection and ranking based on web service description model *WSDM-Q*. Then we use particle swarm optimization to further enhance the accuracy of optimized results. Finally, we discussed a efficient quality updating mechanism in terms of service requesters' feedback information. Extensible experimental results demonstrate that the proposed framework *WSSR-Q* can fulfill service requesters' non-functional requirements and achieve better web service selection effectiveness.

## References

- [1] K. Yue, X. Wang and A. Zhou, "Underlying techniques for web services: A Survey," *Journal of Software*, vol. 15, no. 3, 2004, pp.428-442.
- [2] M. Paolucci, T. Kawamura, T. Payne and K. Sycara, "Semantic matching of web services capabilities," *In Proc. of the 1st Intl. Semantic Web Conference*, 2002, pp.333-347.

- [3] Z. Xu, P. Martin, W. Powley and F. Zulkernine, "Reputation-enhanced QoS-based web services discovery," *In Proc. of the IEEE Intl. Conf. on Web services*, 2007, pp.249-256.
- [4] Y. Liu, A. Ngu and L. Zeng, "QoS computation and policing in dynamic web service selection," *In Proc. of the 13th Intl. Conf. on World Wide Web*, New York: ACM Press, 2004, pp.66-73.
- [5] X. Wang, T. Vitvar, M. Kerrigan and I. Toma, "A QoSaware selection model for semantic web services," *In Proc. of the 4th Intl. Conf. on Service-Oriented Computing*, 2006, pp.390-401.
- [6] D. Roman, U. Keller, H. Lausen, et al, "Web service modeling ontology," *Applied Ontology*, vol. 1, no. 1, 2005, pp.77-106.
- [7] V. Diamadopoulou, C. Makris, Y. Panagis and E. Sakkopoulos, "Techniques to support web service selection and consumption with QoS characteristics". *Journal of Network and Computer Applications*, vol. 31, 2008, pp.108-130.
- [8] S. Deng, J. Yin, Y. Li, J. Wu and Z. Wu, "A method of semantic web service discovery based on bipartite graph matching," *Chinese Journal of Computers*, vol. 31, no. 8, 2008, pp.1364-1375.
- [9] Guobing Zou, Yang Xiang, Yanglan Gan, "An agent-based web service selection and ranking framework with QoS"

### Authors



**T.A. Deepak Kumar** obtained his B.Tech degree from Madras Institute of Technology, Anna university. He is presently doing Masters degree in Sri Venkateshwara College of Engineering, Anna university.



**Dr. G. Sumathi** obtained her B.E. degree in Electronics and Communication from Bharathidasan University, M.E. degree in Computer Science and Engineering from Regional Engineering College, Tiruchirappalli and Ph.D in Computer Science and Engineering from National Institute of Technology, Tiruchirappalli. She had been trained at Carnegie Mellon University, Pittsburgh, U.S.A. Presently, she is working as a Professor in the Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur, Tamil Nadu, India. She is the life member of ISTE. Her research interest includes grid & cloud computing and clusters.