A Cloud Service Composition Model Based on Virtual Chord Ring and QoS Constraint

Xing Wang, Jing Chen, Qingtao Wu, Mingchuan Zhang, Ruijuan Zheng
Information Engineering College, Henan University of Science and Technology,
Luoyang 471023, Henan Province, China

Abstract – The single service always difficult to meet the users personalized requirements in cloud service environments, which needs by means of combining multiple services to provide services for users. This paper proposes a cloud service composition model based on virtual chord ring, and the virtual organization chord rings with QoS constraints as the basis of service composition. The genetic algorithm as the basic algorithm to obtain the global optimal composite service that meets the constraint conditions. The simulation results show that the service composition model not only has the advantage of global search to provide services of satisfying user needs, but also has certain advantages of shortening combination time, speeding up the convergence speed and improving the precision. The service composition model improves the efficiency of service composition.

Keywords – Service Composition, Virtual Organization, QoS Constraints, Genetic Algorithm.

I. INTRODUCTION

Cloud computing[1] is a new computing pattern and rises in recent years which based on the Internet. It provides dynamic scalable services to meet computing requires of users. The requirements of cloud users have personalization features. There are large differences between cloud service requirements and traditional service requirements such as requirement, description and the degree of abstract about functions. It is the key technology that how to retrieve the personalized services of users and proceed dynamic combination of service[2,3] in cloud computing field.

In cloud computing environment, the single service always cannot meet the needs of users, which needs to select several services from cloud services to organic combine to provide services for the users[4]. At present, Local and global optimization are two main strategies method based on QoS service composition[5]. Local optimization just focuses on selecting corresponding services for each service node, while it has no strictly requirement on overall quality after service composition. There is no intersection among service instance corresponding with each service node, so, local optimization always just weighted and sorts for service QoS index during selecting every service instance, while the logical relationship between service nodes usually not considered. Global optimization strategy considered overall quality after service composition. In the process of service composition, QoS constraints are considered as constraint condition of service composition overall quality, and it selects the optimal service composition scheme under the constraint conditions. Therefore, global optimization strategy based on the QoS service composition has higher performance.

Reference [6] proposed multi-objective genetic algorithm to solve the problem of service composition global optimization based on QoS, and it could optimize multiple targets parameters at the same time. It solves the problem of Web service composition to some extent, however, these algorithms are easily fall into local optimal state. Reference [7] adopted functional clustering method to realize the QoS automatic service composition, and evaluated QoS of different clusters by clustering services with similar function, but this method cannot handle the results with more complex service combination. Reference [8] proposed a new type of semantic Web Service discovery model base on chord network and ontology. It introduced the concept of ontology concept and service feature vector. This model adopted routing algorithm of incompatible hash to retrieve registration services, which improves the efficiency of the service discovery, while it cannot be mapped to multi-ontology concept group at the same time. Reference [9] proposed a genetic algorithm coded by two-dimensional matrix, and used a mutation strategy to calculate fitness, while it has no combined with the research of services field. At present, for the service composition problem, there are the methods of service composition based on QoS constraints[10] and service composition[11] of meeting global search. These methods have no clearly involve virtualization organization of service resources, and the combination speed and efficiency need to be improved in further.

Traditional service composition technology combined the specific services of different functions, which tries to meet the complex needs of users. It is a centralized service retrieval matching model. The algorithm has a higher complexity. In the face of a large number of service requests, the resource servers have a load imbalance problem, which slows service composition speed. In cloud computing environment, it uses a resource virtualization technology[12] and manages all kinds of service resources to form a single or multiple independent logical resources available to the user. It reduces the service cost and execution time, balances the load of the resource server, and improves the speed of the service composition.

Therefore, based on the previous research result[13], this paper proposes a cloud service composition model based on virtual chord ring. It virtual organizes cloud services to form the chord rings[14] and QoS constraint, which speeds up the retrieval service. Genetic algorithm is combined into this method to complete service composition[15]. It reduces the average running time of service composition and improves the efficiency of the service composition.
II. SYSTEM MODEL

A. QoS Description

Service composition is a method, process and technology of logic entirety combined by several sever instances according to the given application logic. Each service node corresponding a candidate service set, and contains several specific services. These specific services have the same functional description and interface information while have different QoS attributes. In order to understand the process of service composition more intuitively, we assume that the service composition model contains 9 service nodes; each node $S_i$ has $m_i$ candidate services, the process of service composition as shown in Fig. 1.

The process of service composition actually is a workflow based on cloud service. Each service composition contains four kinds of basic structures according to the logical relationship between different service nodes, such as sequence, parallel, selection and circulation. Most service compositions are composed of these four basic structures. Thus, the service composition QoS is obtained by polymerizing basic structure QoS. This paper selects 4 common QoS parameters, i.e. the service execution time $T$, the execution cost $P$, the service reliability $R$, the service reputation $Rep$; $cs$ is set as service composition, and $s_i$ is the selected service instance of the service node $Si$. The QoS of $s_i$ and $cs$ can be described as (1) and (2).

$$Q(s_i) = (T, P, R, Rep)$$
$$Q(cs) = (T_{cs}, P_{cs}, R_{cs}, Rep_{cs})$$

The QoS calculation formulas of four structures as shown in table 1.

<table>
<thead>
<tr>
<th>structure</th>
<th>$T$</th>
<th>$P$</th>
<th>$R$</th>
<th>$Rep$</th>
</tr>
</thead>
<tbody>
<tr>
<td>sequence</td>
<td>$\sum_{i=1}^{m} T_i$</td>
<td>$\sum_{i=1}^{m} P_i$</td>
<td>$\prod_{i=1}^{m} R_i$</td>
<td>$\sum_{i=1}^{m} Rep_{p_i}/m$</td>
</tr>
<tr>
<td>parallel</td>
<td>max {$T_i$}</td>
<td>$\sum_{i=1}^{m} P_i$</td>
<td>min {$R_i$}</td>
<td>$\sum_{i=1}^{m} Rep_{p_i}/m$</td>
</tr>
<tr>
<td>selection</td>
<td>$\sum_{i=1}^{m} T_i$</td>
<td>$\sum_{i=1}^{m} P_i$</td>
<td>$\prod_{i=1}^{m} R \partial_i$</td>
<td>$\sum_{i=1}^{m} \partial_i Rep_{p_i}$</td>
</tr>
<tr>
<td>circulation</td>
<td>$k \sum_{i=1}^{m} T_i$</td>
<td>$k \sum_{i=1}^{m} P_i$</td>
<td>$\prod_{i=1}^{m} R_i$</td>
<td>$\sum_{i=1}^{m} Rep_{p_i}/m$</td>
</tr>
</tbody>
</table>

Service combination is the process of dynamic autonomously selection, and it follows certain rules and constraints. It selects the service instance which meets user demands to compose global optimal logic system. It needs to coordinate multiple QoS parameters in this process. Setting combination constraints is the key of success in service composition. This paper selects the four QoS attributes mentioned above as the parameters, and makes execution time and execution cost as two objective criterions. It hopes to get the minimum execution time and implementation costs, and makes the service reliability and the service reputation as two constraints, which indicates minimum reliability and reputation required by the service composition. A constrained multi-objective service composition optimization model can be described as equations (3) and (4).

$$\min T(cs), \min P(cs)$$
$$R(cs) \geq R_{min}, Rep(cs) \geq Rep_{min}$$

In (3) and (4), $T(cs), P(cs), R(cs)$ and $Rep(cs)$ are the time, cost, dependability and reputation of the service composition respectively. In fact, there are many kinds of QoS parameters, which could transform them into objective function and constraints of multi-objective optimization problems according to the proposed method. Therefore, this model can be extended to any more objective functions and constraint conditions.

B. Service Resources Organization

According to the interface functions and QoS attributes of cloud services, the corresponding expanded name of interface function is generated by hash algorithm, so, the expanded name could be called functional label of service. Cloud services get an m-bit identification number by their own function for SHA-1 operation and called it as ID, the related definitions are given as follows:

Definition 1 (sub-chord ring) According to the function label of services, the service with same function concatenation into a logical ring according to ID orders, namely sub-chord ring.

Definition 2 (main chord ring) In function labels, main chord ring is composed by services with largest ID in all sub-chord rings.

Fig. 2. Cloud Service Resource Organization Model
III. THE IMPLEMENTATION OF CLOUD SERVICES COMPOSITION MODEL

The genetic algorithm is introduced into cloud services composition field, and a set of QoS functional attributes in virtual chord rings are mapped to the chromosomes of genetic algorithm. It uses the natural advantages of genetic algorithm for sequence, parallel, selection and circulation operations to solve the problem of the efficiency of the cloud service composition. The detailed algorithm process is given as follows.

A. The Basic Principle of Genetic Algorithm

Genetic algorithm is a self-adaption global optimization method formed by the process of simulation biology genetic and evolutionary in the natural environment. It has the characteristics of global and parallelism, and it could search the solution space well to generate useful solutions and achieve the goal of optimization and search problems. For complex optimization problems, this algorithm depends on three operators to obtain optimal solutions, which needs no modeling or complex operation.

The basic process of genetic algorithm is as follows:

1. Initialization: initial chromosomes data are generated randomly. Every chromosome data is an individual, and m individuals generate a group. The maximum iterations set as T, and counter is initialized to 0, so population is initialized to P(0);

2. Fitness Evaluation Test: Individual fitness is calculated by fitness calculation standard. The result as the evidence of continuing to reproduce or dying out;

3. Selecting Operation: Based on calculating fitness of step (2), the individuals with high fitness value are selected from the current group and put them into the next iteration, while the individuals with low fitness value are abandoned;

4. Crossover Operation: The crossover operator is the core of genetic algorithm. The parent individuals generate new offspring individuals according to specific cross rules, which reflects the principle of information exchange;

5. Mutation Operation: An individual is selected from the group randomly, and some gene values in the individual is changed by a certain probability to make the genes variation. Group P(t) gets the next generation group P(t+1) through the three steps of genetic algorithm;

6. Determining Termination Condition: if the value of evolution algebra counter t>T, then the individual with maximum fitness is the optimal solution to output, and the calculation is terminated.

B. Service Composition Algorithm Based on Genetic Thought

This paper combines virtual chord rings with genetic algorithms for proceeding service composition. A feasible solution of service composition corresponds to a chromosome of genetic algorithm; each service of the feasible solution corresponds to each gene of chromosomes. Thus, this paper uses genetic algorithm with path coding, where a path is a chromosome and a feasible solution at the same time. Every path node is a gene and a service node at the same times, which corresponding with the sub-chord ring in the chord ring.

The coding forms are shown in Fig. 3.

1. Initializing Population: Each sub-chord ring corresponds to a set of service candidates. The specific service instance is selected from every sub-chord ring which matches condition to form an individual. It repeats many times until the size of the population reach M.

2. Fitness Function: Eq. 5 shows how to set fitness function of the chromosome $g$:

$$ F(g) = \frac{w_rR(g) + w_sReD(g)}{w_fP(g) + w_sT(g)} $$

Where $w_r$, $w_s$, $w_p$ and $w_f$ represent the weight of every kind of QoS attribute index respectively, $\{w_r+w_s+w_p+w_f=1\}$; $R(g)$, $ReD(g)$, $P(g)$ and $T(g)$ represent the value of dependability, reputation, cost and time respectively.

3. Selecting Operator: It adopts the method of proportional selecting, each individual forms a disk with whose area is 1, which according to the proportions of its fitness value and total fitness value. The pointing individual will be copied to the next generation after the pointer stopped. The higher value of the fitness is, the higher of selecting proportion will be. However, the individual with low fitness is may selected in the disk. This is helpful to maintain the diversity of population.

4. Crossover Operator: This algorithm adopts single-point crossover point for chromosome. A crossover point is selected randomly in chromosomes, and a new chromosome is composed by all of the genes before the first chromosome crossover point and all of the genes after the second chromosome crossover point (include the second chromosome crossover point), while the other new chromosome is composed by all of the genes before the second chromosome crossover point and all of the genes after the first chromosome crossover point (include the first chromosome crossover point).

5. Mutation Operator: A genes is selected from individual chromosomes by a certain probability, and it sees as a variation point. A new service instance is selected from its corresponding sub-chord ring, and it replaces the service instance on the current node.

6. End Conditions of Genetic Algorithm: The iterative time of the genetic algorithm is setted as N, where $N \leq N_{max}$, $N_{max}$ represent the maximum iterative time of the genetic algorithm. In practice, the $N$ is freely input according to requirements. If iterative time is more than $N_{max}$, the algorithm is over, and it returns the optimal service combination plan.
C. Algorithm Implementation

Firstly, cloud service resources for SHA-1 operate to obtain features label ID of every service resource. Then, the service after calculating for virtual organization to form chord ring. Finally, the specific services are selected from the corresponding sub-chord ring randomly to form an individual, which according to the number of service tasks and functions. The initialized population $P(0)$, which size is $M$, is formed after operating many times. Then, individual fitness value is calculated by the operation of selection, crossover and mutation. The detailed flow chart is shown in Fig. 4.

The pseudo-code is as follows:

```pseudo
while (t <= Nmax - the maximum iterative time;)
    for (i = 1 to M) do{
        Evaluating fitness of $P(t)$; //To calculate the fitness of each individual in $P(t)$
        end for
        for (i = 1 to M) do{
            Selecting operation to $P(t)$; //Selection operator acting on the group
            end for
            Crossover operation to $P(t)$; //Crossover operator acting on the group
            end for
            Mutation operation to $P(t)$; //Mutation operator acting on the group
            end for
            $P(t+1) = P(t)$; //Get the next generation $P(t+1)$
            end for
            $t = t + 1$; // Termination conditions determine $t$
        end while
    export K // Output the optimal composition K
end
```

IV. EXPERIMENTATION

Simulation program based on C# is designed to test the performance of the proposed service composition model, which simulates the interaction behavior between multiple services and users. The test running environment is Windows 7 operating system, hardware environment is Intel (R) Core(TM) i3-2130 CPU @ 3.40GHz, 2GB RAM memory. The algorithm of population size is pop size=150, the maximum genetic algebra is $N_{max}=100$, the crossover probability is $P_c=0.9$ and the mutation probability is $P_m=0.01$. By default, the number of service combination is 4, the candidate services of each task are 30, which have been processed by SHA-1 and generated corresponding sub-chord rings, the number of constraints is 2, and the number of QoS attribute is 4. The result returns the optimal solution. Every experiment runs ten times and gets the average as result.

In the process of global searching, the gene is selected based on chord ring corresponding QoS attributes. The compare results between this algorithm and traditional genetic algorithms are shown as follows.

![Fig. 5. The Average Run-time](image)
As can be seen from Fig. 5, the proposed algorithm in this paper needs organize virtualization service into chord rings, which spends some times. Thus, the overall average running time is slightly higher than the traditional genetic algorithm when tasks are fewer. When the tasks are gradually increased over 150, this algorithm operates the ordered chord ring when selected service, which has a faster speed of service retrieval and higher efficiency. As a result, it reduces the average time about 10% than traditional genetic algorithms.

Fig. 6 shows the composition between the service composition algorithm proposed in this paper and traditional genetic service composition algorithm on the convergence speed. All of the operands in this paper are virtual organization chord rings, which has certain rules for selecting service. Therefore, under the condition of the same iterative, it has a faster speed in running speed than traditional genetic service composition algorithm. The convergence is speeding up about 2.8%.

Fig. 7 shows that the two algorithms have the same average precision basically when services are fewer. The functions and attributes of service have more diversity with the increasing number of services. QoS attributes virtual organize into chord rings, then these chord rings are classified according to function attributes of service, which makes more convenience for searching service, and doesn’t easy to fall into local optimum. When the number of services is gradually increased over 300, the average precision of the service composition improves about 3.7% than traditional genetic algorithm.

V. CONCLUSION

This paper composes service resource virtualization into chord rings, and adopts genetic algorithm to form the global optimal service which satisfied QoS constraints to solve service composition. The conclusions are shown as follows:

1) The service resources are constructed chord rings according to hash algorithm. It makes service resources have some certain retrieval rules to speed up the service retrieval, which provides the basis of resource retrieval;

2) This paper combines chromosomes of genetic algorithm with chord ring skillfully, and uses the advantages of the global search of genetic algorithm to obtain service composition which satisfied QoS constraints. It proves the effective feasibility of this algorithm;

3) The simulation results show that the proposed algorithm improves the speed and efficiency of the service composition than traditional genetic service composition algorithm. For example, with the increasing of tasks, it reduces the average time about 10%; with the increasing of iterations, the convergence is speeding up about 2.8%; and with the increasing of services, the average precision improves about 3.7%.

VI. ACKNOWLEDGMENT

This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grant no.U1404611, no. U1204614, and no. 61370221, in part by Program for Science & Technology Innovative Research Team in University of Henan Province under Grant no. 14IRTSTHN021, and in part by the Program for Science & Technology Innovation Talents in the University of Henan Province under Grant no.14HASTIT045 and no. 16HASTIT035, in part by Henan science and technology innovation outstanding talent under Grant no.164200510007.

REFERENCES


AUTHOR'S PROFILE

Xing Wang was born in Henan Province, PRC in April 1990. He studied in Henan University of Science and Technology (Henan, PRC) from Sept 2013 to now, majored in computer application technology. His research interests include cloud computing and information security.

Jing Chen was born in Henan Province, PRC in April 1990. She studied in Henan University of Science and Technology (Henan, PRC) from 2014 to now, majored in computer application technology. Her research interests include Mobile Internet and dependability management.

Qingtao Wu was born in Jiangsu Province, PRC in Mar 1975. He studied in East China University of Science and Technology (Shanghai, PRC) from Mar 2003 to Mar 2006, majored in computer application and earned a Doctor of Engineering Degree in three years time. He works as a Professor in Henan University of Science and Technology from Mar 2006 to now. His research interests include component technology, computer security and future Internet security. Mr. Wu holds a membership of China Computer Federation (CCF).

Mingchuan Zhang was born in Henan Province, PRC in May 1977. He studied in Beijing University of Posts and Telecommunications (Beijing, PRC) from Mar 2011 to Mar 2014, majored in computer application and earned a Doctor of Engineering Degree in three years time. He works as an Associate Professor in Henan University of Science and Technology from Mar. 2005 to now. His research interests include ad hoc network, Internet of Things, cognitive network and future Internet technology.