An Efficient Approach to Optimization of Service Composition in Cloud Environment

YANG Yan, WANG Sai, LI Rong

Computer School, Central China Normal University, Wuhan, China, 430079

Email: ms_yangyan@163.com

Keywords: service composition; optimization; cloud computing

Abstract: In cloud computing environment, service composition provides an effective way to implement a composite service. This paper investigates the issue of optimization of cloud service selection and composition in cloud environment. First, a cloud service filtrating and composition optimization index system is proposed, which can be used to characterize and differentiate services with similar functions and narrow the selection scope of candidate cloud services according to user requirements for subsequent composition. Then an optimization method is introduced for cloud service composition, which contains a service filtrating process based on evidential reasoning approach before the mathematical model is established and resolved. After that, the case study in a travel planning scenario is presented, which shows the proposed method can quickly reduce the solution space and improve the efficiency of service composition in cloud environment.

1. Introduction

Cloud computing is a new service computing mode that enables users to use on-demand computing capabilities just like using public resources, such as electricity, water and gas. It can provide key software for enterprise management, effectively reduce the cost of hardware and software and maintenance costs of IT, and adapt to the needs of enterprises at different stages of development [1] [2]. The main abstraction level of cloud computing includes Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) [3]. Cloud computing can be regarded as a Service-Oriented Architecture (SOA) that encapsulates Internet resources including infrastructure, platform and application, as services [4]. The IaaS layer can be considered as the service encapsulation of the hardware / operating system level. The PaaS layer can be regarded as the service encapsulation of components and middleware. And the SaaS layer can be seen as the service encapsulation in application software level.

In cloud computing environment, the service composition provides a way to efficiently implement a composite service [5]. On the other hand, cloud computing provides a rich selection of atomic services for service compositions. However, the needs of users are complex, including the functional needs of composite services as well as non-functional requirements [6]. It is difficult to choose the most appropriate cloud services in cloud computing platform. Therefore, the problem of service selection is a challenging research topic considering multiple objectives, multiple constraints and even system load balancing. Due to the high arrival rate of cloud service composition requests in cloud computing environment, the service selection method needs to have higher real-time performance and more efficiency.

In this paper, we investigate the issue of optimization of cloud service selection and composition in cloud environment. First, we propose a cloud service filtrating and composition optimization index system, which is used here to characterize and differentiate services with similar functions and narrow the selection scope of candidate cloud services according to user requirements for subsequent composition. Then we introduce an optimization method of cloud service composition which contains a service filtrating process based on evidential reasoning. After that, we take travel planning as an example to expound our approach to cloud service selection and composition. The analysis shows the proposed method can quickly reduce the solution space and improve the efficiency of service composition in cloud environment.

The remainder of this paper is organized as follows. In section 2, a cloud service filtrating and composition optimization index system is proposed. In section 3, we introduce an optimization method of cloud service composition. In section 4, a case study is given. Section 5 concludes the paper.

2. Cloud Service Filtrating and Composition Optimization Index System

The optimization of cloud service composition refers to the process that users choose and combine cloud service resources to complete their required tasks with cloud computing service platform. First, the user submits the requirement to the cloud computing service platform and the platform decomposes the task into the sub-tasks that can be completed by the cloud service; then, the cloud computing service platform selects the candidate cloud service sets to complete the sub-tasks according to the requirements of users; finally, according to a certain task order, the services are combined and the optimal cloud service composition is determined to cooperate to complete the required task. In order to realize the composition optimization of cloud services, systematic and complete, scientific and flexible, objective and feasible cloud service filtrating and composition optimization index system is vital. In view of this, according to the process of cloud service composition, the cloud service filtrating and composition optimization index system is established. The following is a detailed description of the cloud service filtrating and composition optimization index system.

The cloud service filtrating indexes include:

(1) Time: Time (for short *T*) refers to the total duration of the execution process of a cloud service. It can be calculated by $T=T_{exe}+T_{que}+T_{tran}$, where T_{exe} represents the actual execution time of the service; T_{que} represents the queueing time, which means the duration between the arrival of cloud service to the queue of service instances and its beginning of execution; T_{tran} is the transmission delay.

(2) Cost: Cost (for short *C*) refers to the price payed by the requestor for a specific cloud service. When a cloud service is registered, its cost will be uploaded to the service catalog. Once the cost changes, an updating message is sent to the catalog by the service in order to keep the cost value in the catalog up to date.

(3) Availability: Availability (for short *A*) refers to the possibility that a cloud service can normally provide a specific service. Availability is expressed as $A=T_{avail}/T_{total}$, where T_{avail} represents the total running time available for cloud service in the time period T_{total}. The service node can monitor the availability of a cloud service and send it to the service catalog when the catalog inquiries.

(4) Reliability: Reliability (for short *R*) reflects the capability or possibility of a cloud service completing the assigned service within the specified time under certain conditions and is expressed by successful execution rate of the cloud service, namely $R=R_{exe}(t)/B(t)$, where $R_{exe}(t)$ represents the number of tasks executed by the cloud service successfully in the time period *t*; *B(t)* represents the times of the cloud service being awakened in the time period *t*.

(5) Reputation: Reputation (for short *Rep*) is mainly used to measure the degree to which a cloud service is trustworthy, namely, the user's satisfaction with the security and reliability of the cloud service. Different users may give different evaluations to the same cloud service; therefore, the reputation of a cloud service is usually determined by the evaluations of different users. Reputation is expressed as $Rep=\sum_{i=1}^{n} S_i/n$, where S_i is the rating score of service reputation by user; *n* represents the rating times.

(6) CPU occupation rate: CPU occupation rate (for short *Cor*) refers to the proportion of the computing power needed by a cloud service to the total available computing power of the service node. It can be expressed as $Cor=CPU_{req}/CPU_{avail}$, where CPU_{req} represents the computing power needed by a cloud service. When the cloud service instance is developed and deployed on the

corresponding service node, the developer can use tools such as the performance monitor to test. It can be obtained by changing the input data in the test case to measure the different CPU occupancy rate and obtain the mean value of the integral. CPU_{avail} represents the available computing power of the service node where the cloud service resides in, which can be detected by the node itself.

(7) Bandwidth occupation rate: CPU occupation rate (for short *Bor*), similar to *Cor*, refers to the bandwidth utilization of overlay connection between candidate cloud services. It can be expressed as $Bor=BW_{req}/BW_{avail}$, where BW_{req} represents the overlay connection bandwidth required between two candidate cloud services. It can be estimated based on the output data rate of the prodromal service instance or the input data rate of a subsequent service instance. BW_{avail} represents the available bandwidth.

3. Optimization Method of Cloud Service Composition

Cloud computing service platform contains a huge number of various cloud services and there is a great difference between different services. In this case, direct composition optimization of cloud services often leads to a large amount of computation and low efficiency, so it is difficult to get the best cloud service composition. Therefore, the cloud service filtrating is vital for the optimization of cloud service composition. By filtering the cloud services not in the user setting range, the feasible space for optimization of cloud service composition is significantly reduced, effectively improving the efficiency of cloud service composition so as to get the optimal cloud service composition quickly.

Based on the filtrating index above, the cloud services are filtered with the evidential reasoning approach. The following is a brief introduction to the evidential reasoning approach and the filtrating process with evidential reasoning approach is given.

3.1. Evidential reasoning

Assuming that uncertain multi attribute decision making problems include M schemes p_l $(l = 1, \ldots, M)$ and L decision making attributes e_i $(i = 1, \ldots, L)$, relative weight of L attributes is expressed by $w = (w_1, \ldots, w_L)$ and $0 \le w_i \le 1$, $\sum_{i=1}^{L} w_i = 1$.

Given $\Omega = \{D_1, \ldots, D_N\}$ is a set of estimation scale, known as recognition framework of belief function. The uncertainty evaluation of M schemes on L attributes is expressed as $E(e_i(p_l)) = \{(D_n, \beta_{n,i}(p_l)), n = 1, \ldots, N\}$ $(i = 1, \ldots, L;\ l = 1, \ldots, M)$, meeting $\beta_{n,i}(p_l) \ge 0$, $\beta_{\Omega,i}(p_l) \ge 0$, and $\sum_{n=1}^{N} \beta_{n,i}(p_l) + \beta_{\Omega,i}(p_l)=1$, where $\beta_{\Omega,i}(p_l)$ represents unknown.

The analytical algorithm in the evidential reasoning framework is used to synthesize the uncertainty evaluation on each attribute to form an integrated evaluation on each scheme [7] [8], denoted by $E(y(p_l)) = \{(D_n, \beta_n(p_l)), n = 1, \ldots, N\}$ $(l = 1, \ldots, M)$. The utility value of estimation scale given by an expert, namely $u(D_n)$ $(n = 1, \ldots, N)$, meets $0 = u(D_1) < \ldots < u(D_N) = 1$ and $E(y(p_l))$, so the minimum and maximum expected utility values of the scheme p_l $(l = 1, \ldots, M)$ are

$$u_{\min}(p_l) = \sum_{n=2}^{M} \beta_n(p_l) u(D_n) + (\beta_1(p_l) + \beta_\Omega(p_l)) u(D_1),$$

$$u_{\max}(p_l) = \sum_{n=1}^{M-1} \beta_n(p_l) u(D_n) + (\beta_N(p_l) + \beta_\Omega(p_l)) u(D_M).$$

Based on the expected utility, the minimum satisfaction of the scheme p_l $(l = 1, \ldots, M)$ is $V(p_l) = u_{min}(p_l) - max_{m \neq l}\{u_{max}(p_m)\}$, meeting $V(p_l) \in [-1,1]$. The greater the minimum satisfaction of the scheme, the better it is.

3.2. Filtrating process

Based on the above analysis, the cloud service filtrating process based on the evidential reasoning approach is given.

Step 1 The opinions from the expert groups of cloud computing service platform and performance data of cloud services comprehensively give the evaluation of M alternative cloud services on 7 filtrating indexes $E(e_i(p_l))$ $(i = 1, \ldots, 7;\ l = 1, \ldots, M)$;

Step 2 Determine the weight of each evaluation index according to the demand preference of the users $w_i (i = 1,\ldots, 7)$;

Step 3 Use the analytical algorithm in the evidential reasoning approach to synthesize the evaluation of each cloud service on each filtrating attribute so as to calculate the minimum satisfaction of each cloud service $V(p_l)$ $(l = 1, \ldots, M)$;

Step 4 Determine the upper and lower limits V_{max} and V_{min} of V according to the user needs. Use the upper and lower limits to select the cloud services meeting $V_{min} \le V(p_l) \le V_{max}$ for service composition optimization.

Although the bigger the minimum satisfaction of cloud service, the better it is, the overlarge minimum satisfaction will affect the performance of the services on the composition optimization indexes like time and cost. Therefore, the cloud services with the minimum satisfaction of more than V_{max} are also filtered out.

3.3. Optimization of cloud service composition

We assume that the objective of the cloud service composition optimization is to select a collection of cloud services based on user requirements with minimized CPU and bandwidth occupation rate as well as maximized reliability of cloud service composition. Meanwhile, users specify the main constraints regarding the time, cost and availability of the composite service. Therefore, the optimization of cloud service composition can be modeled as a constrained multi-objective optimization problem. The mathematical model and its solution to the problem is detailed in another work [9].

4. Case Study

We take travel planning as an example to expound our approach to cloud service composition. A traveler intends to go to another city for sightseeing. Before starting, he submits a travel reservation request to the cloud computing service platform with a smartphone through a web portal. The cloud computing service platform implements a specific planning algorithm and can produce service composition schemes to complete the required task according to the given information [10]. In this case, a composition scheme with four sub-tasks is generated, i.e. $Task=\{ST_1, ST_2, ST_3, ST_4\}$, as shown in Figure 1.

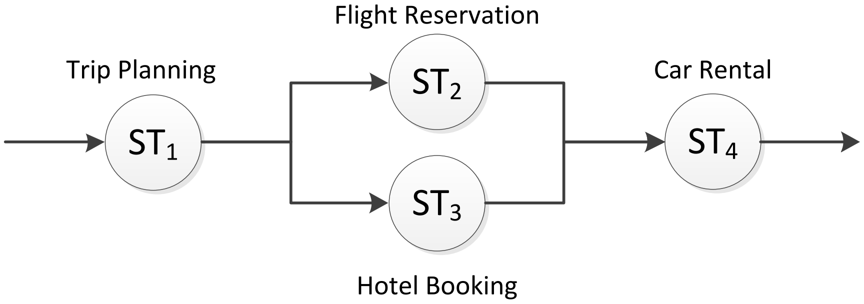

Figure 1. An example of composition scheme for travel planning

First, ST_1 selects the appropriate airline and hotel according to the time, destination and personal preferences given in the request. After that, ST_1 sends the information to the flight reservation service ST_2 and the hotel booking service ST_3. ST_2 and ST_3 are executed in parallel. When the execution is finished, they send the information of the booked hotel and the scenic spots to be visited to the next car rental service ST_4. ST_1 and ST_4 are sequential structure. Each of these four sub-tasks can be completed by a single cloud service independently and executed according to the structure in Figure 1. There is a cloud service collection for each sub-task to select from, i.e. CSS^i $(i=1,\ldots,4)$.

The cloud services to be selected are now filtered, with the process as follows:

(1) The cloud computing service platform evaluates its services in terms of time, cost, availability, reliability, reputation, CPU occupation rate and bandwidth occupation rate to form evaluation data and transforms the evaluation under different frameworks into the evaluation under

a common framework.

(2) According to the needs of users, determine the weight of seven indexes including time, cost, availability, reliability, reputation, CPU occupation rate and bandwidth occupation rate w_i *(i = 1,…, 7)= {0.1, 0.15, 0.2, 0.2, 0.15, 0.1, 0.1 }* and upper limit of the minimum satisfaction of cloud service $V_{max}=0$ and lower limit $V_{min}=-0.2$.

(3) Evidential reasoning approach introduced above is leveraged to calculate the minimum satisfaction of each candidate cloud service.

(4) Based on the value of V_{max} and V_{min}, those cloud services that meet the requirements are screened out.

In this case, the cloud service collections for sub-task ST_1, ST_2, ST_3 and ST_4 to select from contain 18, 20, 32 and 40 cloud services, respectively. So the number of the groups of cloud service composition required to be calculated is 18×20×32×40 = 460800. With the method proposed in this paper, the entire search space has not changed, and the space is reduced to a specific extent after filtering the cloud services. And in this situation, the number of the groups of cloud service composition required to be calculated is 5×6×12×17 = 6120, which has greatly reduced the calculation time. Therefore, under the condition of satisfying the same service requirement, the proposed method can quickly reduce the solution space and improve the efficiency of service composition.

5. Conclusion

Selecting the optimal cloud service composition is the key to complete the required task for users. To improve the efficiency of cloud computing service platform, this paper establishes the cloud service filtrating and composition optimization index system. Based on this index system, the composition optimization of cloud services is divided into two steps: the first step is to use evidential reasoning approach to filter the cloud services to obtain the candidate cloud services in the user setting range; the second step is to build a mathematical model for optimization of cloud service and then solve the model to determine the optimal cloud service composition. Taking the travel planning scenario as an example, the filtering and optimization process of cloud service composition is expounded. A comparison is made between the proposed method and the cloud service composition optimization without service filtering. The case study shows the proposed approach can greatly reduce the solution space and improve the efficiency of cloud service composition. The applicability, effectiveness and efficiency of the proposed method are verified.

Acknowledgement

National Natural Science Foundation of China (Grant No. 61370108).

Reference

[1] M.U. Bokhari, Q.M. Shallal, Y.K. Tamandani, Cloud computing service models: A comparative study [C]. International Conference on Computing for Sustainable Global Development, 2016.

[2] G.J. Mirobi, L. Arockiam, Service Level Agreement in cloud computing: An overview [C]. International Conference on Control, Instrumentation, Communication and Computational Technologies, 2016, pp.753-758.

[3] J. Gibson, R. Rondeau, D. Eveleigh, et al. Benefits and challenges of three cloud computing service models[C]. Fourth International Conference on Computational Aspects of Social Networks, 2013, pp.198-205.

[4] F. G. Mármol, M.Q. Kuhnen, Reputation-based web service orchestration in cloud computing: a survey [J]. Concurrency & Computation Practice & Experience, 2015, 27(9), pp.2390-2412.

[5] A. Jula, E. Sundararajan, Z. Othman, Cloud computing service composition: A systematic literature review [J], Expert Systems with Applications, 2014, 41 (8), pp.3809-3824.

[6] S. Wang, A. Zhou, F. Yang, R.N. Chang, Towards Network-Aware Service Composition in the Cloud [J]. IEEE Transactions on Cloud Computing, 2016 (99), pp.1-14.

[7] Y.M. Wang, J.B. Yanga, Environmental impact assessment using the evidential reasoning approach [J]. European Journal of Operational Research, 2006, 174(3), pp.1885-1913.

[8] C. Fu, M. Xiao, Optimization method of cloud service composition in cloud manufacturing environment [J]. Application Research of Computers, 2014, 31 (6), pp.1744-1747,1751.

[9] Y. Yang, H.X. Yao, S. Wang, An optimization method for reliable cloud service composition with low resource occupancy [C], Proceedings of ICCSPA 2017.

[10] Y. Yang, H.X. Yao, J.M. Ye, et al. Leveraging ontology-aided AI planning for automatic composition of semantic web services [C], Proceedings of ICIII 2010, pp. 110-115.