Web Services Composition Optimization Based on Semi-Markov Decision Process

Yan Hai, Wangzhijian

1College of Computer and Information Engineering, Hohai University, China
2College of Information Engineering, North China University of Water Resources and Electric Power, China

Haiyan@ncwu.edu.cn

Abstract

Web service which is a kind of component on the Internet shows good encapsulation, loosely coupled and cross-platform. However, the single Web service has inevitable limitations in functionality; it can’t provide people with more function and complex service. In order to reduce the cost and time of developing, and achieve services value-added and reuse, we need to combine existing services to form a combination of services which can satisfy user’s demand. There are a lot of services with the same functions in the process of service composition; one of the key issues is to select service by using Qos (Quality of Service). The response time is a significant factor in the process of Web service composition; it is affected by network load and service itself. It is modeled by using finite state continuous time Semi-Markov Decision Process (SMDP), and then Q learning algorithm is used to obtain the optimal service composition policy.

Keywords: Web Service Composition, Quality Of Service (Qos), Semi-Markov Decision Process(Mdp), Q Learning Algorithm

1. Introduction

Currently, Web service composition has become a hot issue of the international services research [1]. In the Web service composition, there may exist many combination programs, so it is necessary to optimize the services combination. In addition, in the Web service composition process, many factors need considered. The key consideration is how to build the Web service composition model, and establish Web service selection algorithm based on QoS of the model to select a candidate for a higher quality of service. The optimal service combination strategy is combined. In the QoS model, [2] simply defines the performance of Web services used to measure quality attributes, there are service costs, response time, reliability and availability, etc..

In solving methods in combinatorial optimization service, the combination of services is divided into the local optimization and global optimization [3, 4]. Then use the integer programming method for solving the service composition plan, but the method requires that the objective function and constraints are linear, and the time complexity of the algorithm increases with the number of tasks from exponential growth. Use multi-attribute decision to make the candidate under each task scoring services to choose the best for each task node in the Web services. In general, this method is suitable for service candidates under each task number, which is not a lot of cases[4]. The crossover and mutation operations of the genetic algorithm [5, 6] based on particle swarm optimization (IPSOA) are introduced, and the further improvement of particle swarm optimization algorithm can be globally optimal combination of QoS services. In the complex service structure, [7] proposed an efficient service selection method. In this method, the service combination is in order to select the parallel structure of the recursion, and then use a recursive branch and bound method to solve the final experimental results show that using this method does not necessarily make the QoS global optimal, but it works well with the complex service structure. The combination of genetic algorithms and the services based on the genetic algorithm build a model for service composition, a certain extent, which solved the service selection problem, but which did not notice the defect of the genetic algorithm itself [9-11]. The application using the K-arm technology solved the gambling machines to meet the Markov nature of the portfolio of services, and the experimental results show that the method has a high hit rate.
A combination of service MDP model was proposed based on a reliable QoS-aware Web service composition algorithm, which used the numerical iterative solution of the optimal portfolio strategy services, but the way had "Modeling hard" problem.

In the service composition system, the response time is an important factor, which will be the network load and the impact of the service itself. If the response time considers the impact on system costs, it will have more practical significance. First, this paper describes the SMDP mathematical model. Second, the response time for performance is using SMDP model and gives a poor phase of discount Q-learning algorithm. With this algorithm, each task node selects specific services so that the entire service composition has the optimal performance. Finally, the simulation experiments validate the effectiveness of the algorithm.

2. SMDP model

SMDP is a continuous-time model, and it is the expansion based on Markov decision process (MDP) [12]. For the modeling of CTMDP model system, it requires the system to stay in each state time exponentially distributed. In real life, the system stays in each state that is subject to the general distribution of the time, so CTMDP does not solve all the problems in reality. SMDP requires the state to obey the general distribution of the staying time, which is more in line with reality, so researching SMDP is more practical significance. As the SMDP is based on the MDP, so the MDP can be applied to the relevant theoretical results in SMDP.

SMDP \( \{X_t, t \geq 0\} \), its model can be described as: At any time \( t \), the system state is the state set \( \Phi = \{1, 2, \cdots, M\} \). At any moment of decision, when the system is in the state \( i \in \Phi \), the action set \( A \) selects from the state of an action \( v(i) \), all the state action mapping forms a control strategy \( v \). \( \Omega = \{v \mid v = (v(1), v(2), \cdots, v(M)), v(i) \in A\} \) determines all deterministic stationary policy. For a fixed strategy \( v \), let \( T_0, T_1, \cdots, T_n \) express the transfer time of stochastic processes, where \( T_0 = 0 \). Let the process is right continuous, that is, \( X_{T_n} = X_{T_n+0} \), denoted by \( X_n = X_{T_n} \), \( n = 0, 1, 2, \cdots \), then \( \{X_n, n \geq 0\} \) are \( \{X_t, t \geq 0\} \) embedded Markov chain. In the state \( X_n = i \) corresponds to the operation decision-making \( v_i \), the system will be the probability of \( p_y(v(i)) \) transferred to the next state \( j \), note \( \pi = [p_y(v(i))] \) for the state transition matrix.

Infinite time SMDP discount performance criteria:

\[
\eta_\alpha = E\left[ \int_0^{\infty} e^{-\alpha t} f(X_t, Y_t, v(X_t))dt \mid X_0 = i \right], i \in \Phi, v \in \Omega, \quad (1)
\]

Which \( 0 < \alpha < 1 \) is the discount factor, \( Y_t \) is the next state of \( X_t \).

When \( \alpha \to 0 \), (1) says that the average performance of any state guidelines, which is defined as:

\[
\eta^* = \lim_{T \to \infty} \frac{1}{T} E\left[ \int_0^{T} f(X_t, Y_t, v(X_t))dt \right], v \in \Omega, \quad (2)
\]

This paper studies the problem of Web service composition, service composition process, the task node for each specific service selected, when the last task executed when the node, that is the end of execution. The following is the cost of finite-time discount criteria.

The price criterion of finite-time discount is defined as:

\[
\eta_{\alpha}^* = E[e^{-\alpha T_n} F(i_n)] + \sum_{n=0}^{N-1} \int_{T_n}^{T_{n+1}} e^{-\alpha t} f(X_t, Y_t, v(X_t))dt \mid X_0 = i], i \in \Phi, v \in \Omega, \quad (3)
\]
Which $F(i_N)$ is the cost of termination of the state, the meaning of $f(X, Y, v(X))$ is the same as the above.

The optimization goal of SMDP looks for optimal strategy $v^*$ in the policy set of $\Omega_s$ to meet:

$$v^* \in \arg \min_{\nu \in \Omega_s} \eta$$ or $$v^* \in \arg \min_{\nu \in \Omega_s} \eta^r$$

For the SMDP model, whether it is finite or infinite period of time performance optimization, the optimization goal is a certain criteria from a strategy focused on finding an optimal strategy. The optimal strategy meets long-term running from the system point of view to get paid to make the system maximum or minimum cost. That is, by comparing the size of remuneration or consideration of the merits determine strategies to achieve optimal control of the SMDP.

3. SMDP model for Web service composition

Web service composition can be divided into static and dynamic combination in accordance with its dynamic. In the modeling stage, the static combination is bound specific service for each abstract tasks. Services are provided to the user with no change. This method is suitable for fixed partners, but the flexibility is poor. If service fails, it can not provide services for the service requester. Dynamic portfolio is based on the user's request to run automatically when the combination of processes, and can automatically find and composite services. Dynamic combination in the portfolio of services designed for the abstract tasks without a specific Web service node binding shows only the functions and tasks necessary to achieve their respective types of services. The combination of services in the course of only the specific dynamic binding of Web services ensure that Web services can adapt to the dynamic combination of the higher of Internet applications. In a practice combination process, the combination of static or dynamic way is determined according to the characteristics of Web services and user needs in the end. If the service relationship between the relatively is fixed, then the combination is static methods. If the service often changes with the needs of users, the dynamic combination methods are used. This process is automatically generated based on the structure of service composition, and therefore falls within the dynamic service composition.

Web services run in the highly challenging dynamic environment, so QoS will frequently change. These changes may be due to reduced service providers caused by the price, but also may be due to the network caused by too much network load, which will cause changes in the data transmission time. Changes caused by the network will affect the data transmission rate and further affecting the response time to the service portfolio. In the service portfolio, the response time is an important QoS attributes. Previous studies is the response time as a QoS attribute value to determine the release. In the actual portfolio of services system, the response time is not necessarily fixed, it will be the impact of network and service itself.

In this paper, considering the response time, network transmission and processing cost of service impact on system performance. Use finite state continuous time SMDP model of service composition. Continuous-time SMDP is a time characteristic of a model, which can solve the system of decision-making, so that the system performance is optimal. Finally, the use of model-independent Q-learning algorithm solves the optimal portfolio of services.

3.1. Web service composition description of the mathematical model

Web service composition can be seen from the initial state to terminate the state of the task flow generated by the process according to the structure. Systems are selected for each task node specific Web services. If the node for the current task successfully calls the service, it will move to the next task node or continue the task for the current section point of call service until it succeeds. Transfer state of the system is shown in Figue.1:
From the decision moment $T_n$ to the next decision moment $T_{n+1}$, experiences three stages: send service requests, processing services and return the results, it is $T_{n+1} = T_n + t_{pr1} + t_{pr2}$.

As the service portfolio contains a finite number of task nodes, so this continuous-time finite SMDP. $V$ is defined in the policy under consideration of the limited time discount criteria:

$$\eta(i) = E[e^{-\alpha T} F(i_N)] + \sum_{n=0}^{N-1} \int_{t_n}^{t_{n+1}} e^{-\alpha t} f(X_n, a_n, X_{n+1}) dt \mid X_0 = i] \quad (4)$$

Which, $F(i_N)$ is the cost of the final state, $f(X_n, a_n, X_{n+1})$ is defined as the current state of $X_n$ system to take action $a_n$ to move to the next state $X_{n+1}$ units of time before the expected costs. $\alpha$ is discount factor, when $\alpha > 0$. $\mu(i)$ represents the state under the discount price, the goal of studying optimize finds an optimal strategy in the strategy space to make the minimum consideration.

In the actual portfolio of services, there may be a combination of a variety of structures. Branching structure can be broken down into different execution paths, cycle structure in accordance with a specified number of cycles is executed. We discuss the services combination of sequent architecture and parallel architecture shown in Fig.2. In the service portfolio, task node $t_4$ and $t_5$ are the parallel structure, others are sequent structure. When a node performs two tasks in parallel, if a task calls the service node fails, it is necessary to re-selected for their service.

**3.2. Q learning optimization algorithm**

This use of Q learning algorithm solves the problem of Web service composition. In the moment of decision $T_n$, in the state $X_n$, the choice of action $a_n$, after a period of time, transfer to the state $X_{n+1}$, which can be divided into three time periods of sending service requests, processing services and returning the results. In these three time periods network resources and processing services need to pay a certain cost of the corresponding three parts, namely a transmission request for service costs, service costs and handling costs to return the results. In addition, the state $X_n$ call the service, which will be an immediate reward for taking action. $C_{a_n}$ takes actions $a_n$ corresponding to the immediate reward, the cumulative expense is calculated as:
\[ f'(X_n, a_n, X_{n+1}) = \int_{t_{\text{ral}}}^{t_{\text{ral+p+r+2}}} e^{-\alpha t} k_2 dt + \int_{t_{\text{ral}}}^{t_{\text{ral+p+r+2}}} e^{-\alpha k_1} dt + \int_{t_{\text{ral+p}}}^{t_{\text{ral+p+r}}} e^{-\alpha t} k_2 dt - C_{a_n} e^{-\alpha (t_{\text{ral+p+r+2})}} \] (5)

Which, \( f'(X_n, a_n, X_{n+1}) \) indicates the obtained cumulative cost of the system from the moment of decision \( T_n \) to \( T_{n+1} \). \( k_1 \) is the cost of processing services per unit time. \( k_2 \) is the network transmission costs per unit time. As the uncertainty of a Web service, the calling process may succeed or fail. If we call the service successfully, \( C_{a_n} = \text{QoS}(S) \), otherwise \( C_{a_n} = 0 \), QoS (S) refers to the comprehensive evaluation of Web services.

In the consideration of the finite stage, the discount differential updates and status can not take action on the infinite time to update the formula. Web service composition based on the characteristics of Q learning algorithm for the general improvement gets a new updating formula. Service portfolio goal is to get an optimal strategy from the initial state to the final state. In the learning and optimization process, the Web services combination from any state proceeding to terminate status is looked as a simulation segment (episode) denoted by \( h \). After the simulation segment, discount \( d_h \) differential criterion is calculated as follows:

\[ d_h = \sum_{n=0}^{l_h} e^{-\alpha T_n} f'(X_n^h, a_n^h, X_{n+1}^h) - Q_\alpha(X_0^h, a_0^h) \] (6)

Which, \( T_0 = 0 \). \( X_n^h \) denotes a n-step state of the h fragment simulation, \( Q_\alpha(X_0^h, a_0^h) \) denotes the initial state of the simulation fragment and action on the Q value.

So get a h simulation and fragment of the initial state and actions \( Q_\alpha(X_0^h, a_0^h) \) value of the update formula:

\[ Q_\alpha(X_0^h, a_0^h) := Q_\alpha(X_0^h, a_0^h) + \gamma_h d_h \] (7)

Which, \( \gamma_h \) is a learning step.

The specific procedure of Q learning algorithm:

1. Initialize parameters. \( h=0 \), set the discount factor \( \alpha \), learning step \( \gamma_h \), learning fragment \( H \), initialize all states - actions on the Q value.

2. The moment of decision \( n = 0 \), arbitrarily chosen initial state \( X_n^h \) (not termination condition), and so the total cost is \( f = 0 \).

3. In the state \( X_n^h \), according to the \( \varepsilon \)-greedy method select action \( a_n^h \) and enforce action \( a_n^h \), observe the next state \( X_{n+1}^h \) and the response time of three time periods according to the formula (5) to calculate the cumulative cost of \( f'(X_n^h, a_n^h, X_{n+1}^h) \), and \( f := f + e^{-\alpha T_n} f'(X_n^h, a_n^h, X_{n+1}^h) \).

4. Determine whether the state \( X_{n+1}^h \) is the end state, if not, then \( n := n + 1 \), turn (3).

5. According to the formula (6) calculate the discount criteria under the differential \( d_h = f - Q_\alpha(X_0^h, a_0^h) \), and according to the formula (7) Update \( Q_\alpha(X_0^h, a_0^h) \).

6. \( h := h + 1 \), if \( h = H \), the study is over or turn to (2).

4. Experimental Results and Analysis

As the current service portfolio is not the relevant standard platform and test data sets, we randomly generate a large number of data to simulate. In the simulation we set \( k_1 = 40 \), \( k_2 = 20 \). \( t_{\text{ral}} \) and
\( t_{n_a 2} \) are subject to the exponential distribution of parameters \( \lambda_{n_a 1} = \lambda_{n_a 2} = 2 \), and \( t_{pr} \) obeys the exponential distribution of parameter \( \lambda_{pr} = 1 \).

To validate the feasibility and effectiveness of the proposed algorithm, we respectively verify the structure of the order portfolio of services and portfolio of services with a parallel structure. Structure of the simulation in order to set the service portfolio of 10 tasks, each task has 10 candidate services. Fig.3 and Fig.4 (The horizontal axis is the iteration steps, and the vertical axis is the price discount) are optimized under different discount factor curve. It can be seen from the figures that the use of Q learning algorithm converge to the optimal or sub-optimal strategy. In the early learning, learning is less effective, but with the increase of learning steps learns more and more knowledge, and the effect will be getting better and better and eventually converge.

![Figure 3. The optimal curve of the order structure when the discount factor \( \alpha = 0.01 \)](image)

![Figure 4. The optimal curve of the order structure when the discount factor \( \alpha = 0.1 \)](image)

In order to analyze the process in several different tasks and each task of the different candidates on the system, calculate the cost of services and the impact of this structure are included the order of 10, 20 and 80 up to the task flow simulation experiments, where each task has respectively 10, 20 and 30 candidate services, the experimental results are shown in Fig.5 (The horizontal axis is the number of tasks, and the vertical axis is the computational cost) . Fig.5 shows that the computational cost of the final strategy will increase with the tasks and the candidate services increasing. When the task is less than 30, the candidate services varies between 10 to 30, the computational cost of the impact is not
great, but when the task is greater than 50, the computational cost will increase as the number of candidate services added.

![Figure 5. The computational cost of different tasks candidate services](image)

Fig.6 (The horizontal axis is the number of tasks, and the vertical axis is the success rate of service portfolio) shows the success rate of the different tasks and candidate services. From Fig.6 we can see that the use of Q learning algorithm can be relatively high success rate of the service portfolio. And service portfolio, the combined success rate with the number of tasks and the number of candidate services have a relationship, the number of tasks is identified with the case of the corresponding number of candidate services, which varies between 10 to 30. The success rate of the portfolio is not very large, but the number of tasks increases, and the combined success rate will be reduced. The more are the candidate services, the more are the changes of the combined success rate. This shows that the more is the number of candidate services, the performance of the system is the more stable, but the system will increase the computational cost, which is consistent with the actual.

![Figure 6. The success rate of the service portfolio of the different tasks and the candidate services](image)

We do the service structure simulation, and the parameter setting is the same as the order structure. Fig.7 and Fig.8 (The horizontal axis is the iteration steps, and the vertical axis is the price discount) are the optimal curve that each task has 10 candidate services under different discount factors. It can be seen from the figures for the parallel architecture of a combination of services, the use of Q learning algorithm will eventually converge, which can be gotten the best or second beat strategies.
Figure 7. The optimal curve of the parallel structure when the discount factor $\alpha = 0.01$

Figure 8. The optimal curve of the parallel structure when the discount factor $\alpha = 0.1$

Fig. 9 (The horizontal axis is the number of candidate services, and the vertical axis is the computational cost) is the computational cost of the parallel structure with a combination of Web services. We can see from the figure that the number of tasks is identified in the case, and the computational cost with the increase in the number of candidate services. When the change of the number of candidate services is not large, the computational cost of the impact is not great, which is consistent with the simulation results of the order structure.

Figure 9. The computational cost of candidate services of the different parallel structure
5. Conclusion

In this paper, the randomness of Web services and the uncertainty of Internet environment mainly considerate the response time of the service portfolio performance. The first gives a description of Web service composition problem, and then is modeled as a finite state continuous time SMDP, because this is a finite period, the infinite time based on the research gives an improved finite time under the discount Q learning algorithm. Finally, consider the portfolio of services with a parallel structure with focusing on cost considerations. Experimental results show the feasibility and effectiveness of the algorithm and the algorithm has a high success rate of the service portfolio and adapts to a dynamic environment.

6. References