

## Performance Prediction Based EX-QoS Driven Approach for Adaptive Service Composition<sup>\*</sup>

LEI YANG, YU DAI<sup>†</sup> AND BIN ZHANG  
*College of Information Science and Engineering*  
*<sup>\*</sup>College of Software*  
*Northeastern University*  
*Shenyang, Liaoning, P.R. China*  
*E-mail: yanglei@mail.neu.edu.cn*

Web services run in a highly dynamic environment, as a result of which the QoS will change relatively frequently. In order to make the composite service adapt to such dynamic property of web services, we propose an adaptive approach for web service composition. This approach uses an EX-QoS model which makes the approach applicable in the decentralized service composition to achieve global optimization. Besides this, in order to make the composite service adjust itself quickly when execution problems occur, this paper uses replacement composite services backed up before the execution. Then, when a service fails, the composite service can easily switch to the replacement without affecting the runtime performance. In order to preserve the availability of the replacement, a re-selection for updating the replacement is used which is on the basis of a performance prediction to make the re-selection complete as early before the invocation of the failed service as possible. Experiments show that the proposed solutions have better performance in supporting the adaptive decentralized service composition.

**Keywords:** decentralized service composition, failure prediction, QoS, re-selection, adaptive approach

### 1. INTRODUCTION

With the popularity of web service, the creation of valued-added services by composing available services is gaining a significant momentum [1]. However, as the number of web services providing overlapping or identical functionality, albeit with different non-functional properties (*e.g.* QoS) on the Web is huge, a choice needs to be made to determine which services are to participate in a given composite service.

The web service selection is to select the best set of services, taking into consideration of QoS and user's constraints in order to make the composite one adapt to the dynamic nature of the service [2]. Currently, many people focus their study on this selection problem and the solutions can be classified into 2 categories: selection with the local optimization and selection with the global optimization.

Selection with the local optimization [2, 3] is to select a service for the task when it is required to execute. The advantage of this kind of approach is that it considers the QoS of the services at runtime. However, such local approach can only guarantee local QoS constraints, *i.e.*, candidate web services are selected according to a desired characteristic, *e.g.*, the price of a single web service invocation is lower than a given threshold.

---

Received March 16, 2008; accepted October 24, 2008.

Communicated by Chi-Sheng Shih.

<sup>\*</sup> This work was supported by the National Natural Science Foundation of China (No. 60773218).

Selection with the global optimization [2, 4, 5] aims at satisfying the user's constraints for the whole composite service. This solution turns the selection problem into a multiple constraints satisfaction problem (MCSP) and some classical algorithm (such as integer programming algorithm) as well as other artificial intelligence algorithm (such as genetic algorithm [6] and simulated annealing algorithm [7]) can be used.

The main issue for the fulfillment of the end-to-end constraint is the variability of the QoS. Indeed, the QoS of a web service may evolve relatively frequently, either because of internal changes or because of workload fluctuations [2, 8]. If a composite process has a long duration, services selected by the global optimization may change their QoS properties or some services can become unavailable. For such reason, it needs a re-selection in the execution of the composite service to adapt to this variable environment. However, since the time complexity of the re-selection is high [8], the performance of the composite service will be influenced. Thus, how to make the composite service more adaptive and preserve the runtime performance is still a problem needed to solve.

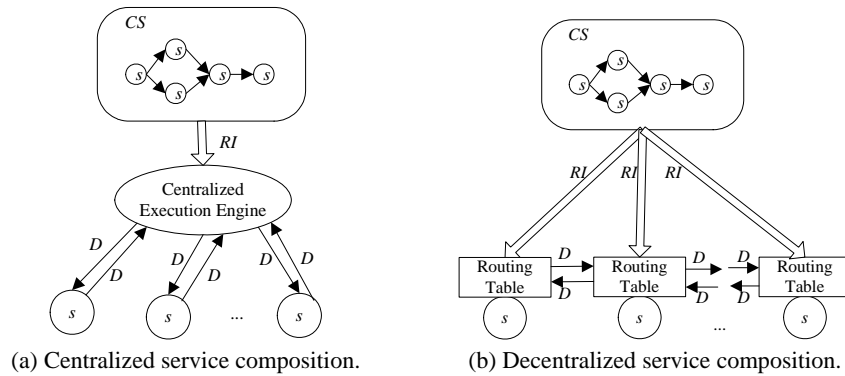


Fig. 1. Two service composition models.

Another drawback of the current global solution is that current approach for finding composite service with the global optimization is only suitable for centralized service composition. There exist two models for invoking a composite service: centralized model [2, 9] and decentralized model [10]. For such two models, QoS for evaluating the quality of services are surely to be computed differently. In centralized model (Fig. 1 (a), in this figure, *CS* signifies composite service, *D* signifies data transmitted, *RI* signifies routing information and *s* signifies service), there exists a centralized engine. According to the routing information of the composite service, the engine will route data among services. In this model, data transmission time which influences the QoS will be the time of routing data between the service and the centralized engine. The selection of one service will not influence the selection of others. However, as web service composition developed in the open environment (the Web), using the centralized model will result in some problems. Firstly, in centralized model, services must route data via a centralized engine which will add additional execution duration time and result in the bottleneck problem. Secondly, certain applications (*e.g.* B2B applications) will not allow routing data via a third party because of the security and business aware problems. Thus, a de-

centralized model for composite service is proposed. Compared with the centralized model, in decentralized model (Fig. 1 (b)), on each service, there exists a routing table generated from the routing information of the composite service. Such routing table is used for routing data among services. In this model, data transmission time will be the time of routing data between the services. Then, the selection of one service will influence the selection of others. Thus, how to select service with global optimization in distributed service composition is still another problem needed to solve.

The goal of this paper is to set the basis to overcome the limits of the previous approaches to web service selection. We introduce a new QoS model for selection with global optimization in decentralized service composition. This new QoS model (EX-QoS) is an extension of the traditional QoS model after considering the relation of data transmission between services. We introduce a new framework for composite service to adapt to a highly variable environment. Key to this framework is the performance prediction, through using which the runtime performance of the composite service can be improved. As will be discussed in the remainder of the paper, our approach is effective.

## 2. RELATED WORKS

To adapt to dynamics of web services and meet the global constraint from users, researchers [8] propose a solution which uses a re-selection. In [8], the re-selection will be triggered when the actual QoS deviates from the initial estimates. When the failure is found, the execution of composite service will be stopped until the re-selection is finished. Therefore, this approach can be only used for runtime-unaware application.

To make the composite service recover from the failure with the minimal extra delay of the execution of the composite service, researchers [11, 12] propose another solution which is on the basis of the replacement. In their works, a replacement composite service is backed up for each service of the composite one. Then, when a service is failed, the composite service can easily switch to a replacement and such self-adaptive process will not affect the execution performance of the composite service. In [11, 12], all the replacement composite services are backed up before the execution of the composite service. Such two approaches do not consider the QoS of services in the execution of the composite service. Therefore, the replacement will not be available sometimes.

Besides this, the above works are only used in a centralized service composition.

Compared with the above works, we aim at solving the problem of adaptation in the decentralized service composition and an EX-QoS model is proposed. Moreover, to effectively adapt to the dynamic environment, this paper combines the approaches of pre-backup before the execution and re-selection in the execution together. In our approach, the re-selection is only used for updating the replacement which is predicted to be unavailable. In this way, the composite service can recover from the failure quickly.

## 3. EX-QOS MODEL FOR DECENTRALIZED SERVICE COMPOSITION

### 3.1 Scenarios of Service Composition

In this section, we will introduce some basic concepts, including composite process,

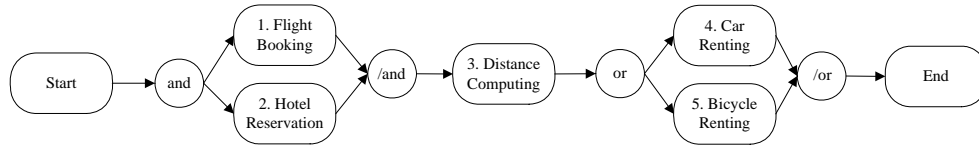


Fig. 2. Composite process of travel planning.

service class, service and composite service through a scenario of travel planning. Fig. 2 illustrates the main steps that are needed in the composition. In the *composite process*, a flight booking is done in parallel with a hotel reservation, after booking and reservation, the distance from the airport to the hotel is computed, and then either a car rental or a bike rental is invoked.

A composite process contains several *service classes*. For example, *flight booking* is a service class in the composite process of the travel planning. A service class is a set of services with similar functions. When it needs to achieve the function claimed by the service class, a service in the set will be assigned to perform this function. For example, when it needs to book a flight, a service  $s$  in the set will be assigned. In this situation, we call that service class *flight booking* is implemented by the service  $s$ . If for each service class in a composite process, it is implemented by a service and the services form a set, we call this set is a *plan* of the composite process. In reality, there exist several plans for a same composite process. Among these plans, the plan which has the best QoS and meets the constraints of the users is the *composite service*.

### 3.2 EX-QoS Model for Decentralized Service Composition

Due to dynamic nature of the web services, researchers propose approaches of QoS driven service selection with the aim of achieving better quality of the composite service. As illustrated in section 1, current approaches seldom focuses on the selection with the global optimization in the decentralized service composition. Besides this, the QoS in the context of the decentralized service composition is different from the one in the centralized service composition. In order to achieve global optimization in decentralized service composition, this paper introduces an EX-QoS model.

**Definition 1** EX-QoS of Atomic Service. For an atomic service  $s$  (which only contains one operation), the QoS of  $s$  can be defined as:  $QoS(s) = \langle Q^t(s), Q^p(s) \rangle$ , where:

- $Q^t(s)$  is the response time of  $s$  which is the interval of time elapsed from the invocation to the completion of service  $s$ .  $Q^t(s)$  can be defined as:  $Q^t(s) = \langle Q^{pt}(s), Q^{dt}(s) \rangle$ , where
  - $Q^{pt}(s)$  is the processing time for the request, which is provided by service providers;
  - $Q^{dt}(s)$  is the sum of the transmission time of request and result among the request sender, the result receiver and the service.  $Q^{dt}(s)$  can be computed as Eq. (1).

$$Q^{dt}(s) = \frac{A_{req}}{B_{req}} + \frac{A_{res}}{B_{res}} \quad (1)$$

where,  $A_{req}$  and  $A_{res}$  are amounts of the data transmitted among the service, the re-

quest sender and the result receiver respectively;  $B_{req}$  and  $B_{res}$  are the average bandwidths among the service, the request sender and the result receiver respectively.

–  $Q^t(s) = Q^{pt}(s) + Q^{dt}(s)$  means that the response time is a sum of request processing time and data transmission time.

- $Q^p(s)$  is the price of invoking  $s$ .

In the decentralized service composition, the data transmission time  $Q^{dt}(s)$  considers the time of data transmission among the request sender, the result receiver and the service, but not the time of data transmission between the service and the execution engine. Then, EX-QoS model can be used in a decentralized service composition.

The works by Cardoso [6] proposes a mathematical model for QoS computation of workflow, which involves aggregation functions for sequence, choice, parallel and iteration structure of the workflow. Based on these aggregation functions in the centralized service composition, this paper gives a set of functions (Table 1) for computing the QoS of composite service  $CS$  in decentralized service composition.  $\rho(s)$  is the probability to invoke the service  $s$  in the composite service with choice structure and  $n(CS)$  is the average number of iterations.

**Table 1. Aggregation functions for computing the QoS of composite service.**

	Aggregation Function
Sequence	$Q^t(CS) = \sum Q^{pt}(s) + \frac{\sum Q^{dt}(s)}{2}$ $Q^p(CS) = \sum Q^p(s)$
Choice	$Q^t(CS) = \sum \rho(s) * Q^t(s)$ $Q^p(CS) = \sum \rho(s) * Q^p(s)$
Parallel	$Q^t(CS) = \max\{Q^t(s)\}$ $Q^p(CS) = \sum Q^p(s)$
Iteration	$Q^t(CS) = n(CS) * Q^t(s)$ $Q^p(CS) = n(CS) * Q^p(s)$

In order to provide composite service which satisfies global constraints and preferences defined by users, services in the service class should be selected according to the QoS. Such selection problem can be formulated as Eq. (2). The aim of the selection is to maximize the fitness function; and meet the constraints defined by the user.

$$\begin{aligned} & \max_v F(CS_v), \\ & \text{s.t. } Q^t(CS_v) \leq Q_c^t \end{aligned} \tag{2}$$

where  $CS_v = \{s_{1,k1}, \dots, s_{i,ki}, s_{n,kn}\}$ ;  $s_{v,kv}$  is the service selected for service class  $sc_v$ ;  $F$  is the fitness function which can be computed as Eq. (3).

$$F(CS_v) = w_t * \left( \frac{Q^t(CS_v) - u_t}{\sigma_t} \right) + w_p * \left( \frac{Q_j^p(CS_v) - u_p}{\sigma_p} \right) \quad (3)$$

Where,  $w_t$  and  $w_p$  are the weights ( $0 \leq w_t, w_p \leq 1, w_t + w_p = 1$ ).  $\sigma$  and  $\mu$  are the standard deviation and average of the QoS values for all plans of the composite process.

#### 4. FRAMEWORK OF THE APPROACH

From Eq. (2), the selection problem is a NP hard problem. Normally, it may take a long time to solve this problem. After the selection, the composite service will execute. Then, there is an interval between the service is selected and to be invoked. As web services operate in a highly changeable environment, the QoS will change during this interval. When the QoS of a service is changed, such changes will make the composite service difficult to satisfy global constraints. Thus, the composite service needs to adjust itself in order to meet the global constraints again. For such reason, we propose an approach (Fig. 3) for composite service adapt to the dynamic nature of the services.

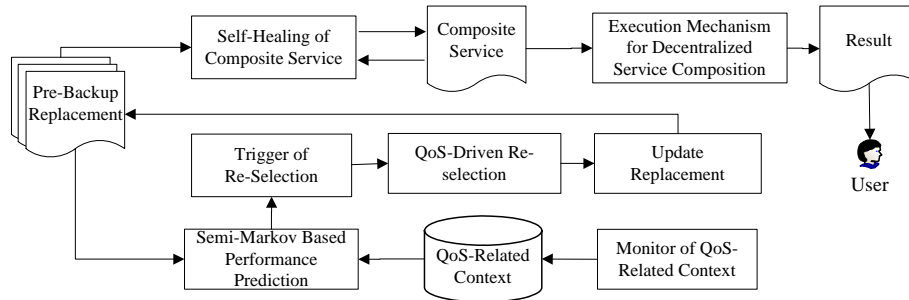


Fig. 3. Framework of the approach.

Firstly, the monitor of QoS-related context will collect QoS-related contexts for prediction.

Secondly, QoS of each service in the composite one will be predicted periodically. If the QoS at invoking time of certain service is predicted to be a large deviation, and the pre-backup replacement is predicted to be unavailable, the process of reselection will be triggered.

After that, taking the original replacement composite service as a reselected slice, a QoS-driven reselection will be done to get the new available replacement.

Finally, when the failed service is to be invoked, the composite service will automatically switch to the replacement to heal itself.

The key to our approach is performance prediction. If the performance of the service is predicted as a failure and the pre-backup replacement is not available, the process of reselection will be triggered. Compared with the works of triggering the reselection when to invoke the failed service, the approach in this paper can make the reselection complete before the invocation of the failed service. Thus, the composite service will not

need to wait for the completeness of the reselection and can adjust itself from the failure as quickly as possible.

## 5. SEMI-MARKOV MODEL FOR PERFORMANCE PREDICTION

The study of this paper is based on the following assumptions: (a) the speed of processing the request is constant; (b) the price of requesting for a service is constant; (c) the execution engine is never failed; (d) the failures by different service and communication links are separate and do not interfere with each other; and (e) during the data transmission process, data transmission speed does not change.

### 5.1 Semi-Markov Model for Data Transmission Speed

Markov Process model is one of the probabilistic models. It is useful in analyzing dynamic behaviors of the system [13]. A semi-Markov Process (SMP) is the extension of Markov's, which is to model time-dependent stochastic behaviors [14]. A SMP is similar to Markov process except that its transition probabilities depend on the amount of time that elapses since the last transition. Since web services operate with frequent changes, the change of data transmission speed can result either from the soft damage of the internet, *e.g.* network load, or from the hard damage of the internet like a communication link that is broken. In this sense, the stochastic behavior of the data transmission speed depends not only on the current state, but also on the duration of the state. Thus, a SMP can be used to analyze the behavior of data transmission speed.

In order to use SMP to analyze the stochastic behavior of data transmission speed, we firstly classify data transmission speed into 3 states: qualified state, soft damage state and hard damage state. The meanings of above 3 states are given in Definition 2. In Definition 2,  $V(t)$  is used to signify the data transmission speed at time  $t$  and  $ST(t)$  is used to signify the state of the data transmission speed at time  $t$ .

**Definition 2** States of Data Transmission Speed. We use  $th\_V_Q$  to signify the threshold of data transmission speed in the qualified state.

- If  $V(t) \geq th\_V_Q$ , then  $ST(t) =$  qualified state;
- If  $0 < V(t) < th\_V_Q$ , then  $ST(t) =$  soft damage state;
- If  $V(t) = 0$ , then  $ST(t) =$  hard damage state.

In this paper,  $th\_V_Q$  is the data transmission speed with a given probability of 70% of the past executions.

After defining the states of data transmission speed, a semi-Markov model for data transmission speed can be defined as follows:

**Definition 3** Semi-Markov Model for Data Transmission Speed. A semi-Markov model for data transmission speed can be defined as:  $SM = \langle Z, P, F \rangle$ , where

- $Z$  is the state space of data transmission speed,  $Z = \{1, 2, 3\}$ . When  $Z = i$ , it means that

the data transmission speed is in state  $i$ . State 1, 2 and 3 are qualified, soft damage and hard damage states respectively;

- $P$  is the matrix of state transition probabilities. If the current state is  $i$ , the next state enters in state  $j$  with probability  $P_{ij}$  and  $\sum_j P_{ij} = 1$ . Especially,  $P_{ii} = 0$ ;
- If the current state is  $i$  and the next state is  $j$ , the duration at state  $i$  until the completion of transition from  $i$  to  $j$  is  $d$ .  $d$  obeys the distribution  $F_{ij}(d)$ .

Let  $H_i(d)$  be the distribution of duration spent in state  $i$ ,  $H_i(d) = \sum_j F_{ij}(d) * P_{ij}$ . The

average duration in state  $i$  can be signified as  $\mu_i$ . According to lemmas [15] of semi-Markov model, there is stationary distribution  $\pi = [\pi_1, \pi_2, \pi_3]$  and for each  $\pi_j$ , it can be computed through getting the solution to Eq. (4). Also, let  $LimP_i$  the steady-state occupancy probability of state  $i$ , it can be computed as Eq. (5).

$$\begin{cases} \pi_j = \sum_{i=1}^5 \pi_i P_{ij} \\ \sum_{i=1}^5 \pi_i = 1 \end{cases} \quad (4)$$

$$LimP_i = \frac{\pi_i \mu_i}{\sum_j \pi_j \mu_j} \quad (5)$$

## 5.2 Failure Prediction Based on Semi-Markov Model

Aiming at improving the availability of the replacement composite service, this paper intends to predict the QoS performance.

### 5.2.1 Description of prediction

As to our prediction problem, we try to predict whether the data transmission speed of the service is a deviation from the estimated one during the invocation. Such a problem can be described as follows: if the current state is  $i$ , current time (predicted time) is  $t$  and the duration in the current state is  $d$ , we need to predict the probability of that the data transmission speed  $V_f$  at future time (the time to be predicted)  $t_f$  must not be below the maximum deviation  $V_e$ . Let  $j$  be the state of  $V_e$ . To solve this problem, we need to consider the following two situations:

**Situation 1:** State  $j$  is the same as  $i$ .

This situation is either because of lasting in state  $i$  from  $t$  to  $t_f$ , or as a result of switching to  $i$  again after multiple transitions. We will consider these two situations.

**Situation 1a:** State  $j$  is the same as  $i$  and during  $t_f - t$  no transition occurs.

Let  $D_i$  be the stochastic variable of duration in state  $i$ . If no transition occurs during  $t_f - t$ , it means that the duration in state  $i$  will be  $d + t_f - t$ . Then, the probability of  $V_f$  at

future time  $t_f$  that is not below the maximum speed  $V_e$  is computed as Eq. (6).

$$\begin{aligned}
 PR &= PR_{1a} = P((V \geq V_e) \wedge (D_i \geq t_f - t + d \mid D_i \geq d)) \\
 &= P(V \geq V_e) * P(D_i \geq t_f - t + d \mid D_i \geq d) \\
 &= (1 - FV_i(V_e)) * \frac{P(D_i \geq d \mid D_i \geq t_f - t + d) * P(D_i \geq t_f - t + d)}{P(D_i \geq d)} \quad (6) \\
 &= (1 - FV_i(V_e)) * \frac{1 - H_i(t_f - t + d)}{1 - H_i(d)}
 \end{aligned}$$

where  $FV_i(v)$  is the distribution of data transmission speed in state  $i$ ;  $P(D_i \geq d + t_f - t \mid D_i \geq d)$  is the probability of duration in state  $i$  bigger than  $d + t_f - t$ , under the condition that it has been already in state  $i$  for  $d$  since the last transition.

**Situation 1b:** State  $j$  is the same as  $i$  and during  $t_f - t$ , at least one transition occurs.

If at least one transition occurs, it means that the duration kept in state  $i$  before the first transition is shorter than  $d + t_f - t$ . Meantime, the probability of re-entering state  $i$  after several transitions during the interval from  $t_f$  to  $t$  is close to the steady-state occupancy probability  $LimP_i$  of state  $i$ . The probability can be computed as Eq. (7).

$$\begin{aligned}
 PR &= PR_{1b} = P((V \geq V_e) \wedge (Z_{t_f} = i) \wedge (d \leq D_i < d + t_f - t \mid D_i \geq d)) \\
 &= P(V \geq V_e) * P(Z_{t_f} = i) * P(d \leq D_i < d + t_f - t \mid D_i \geq d) \\
 &= (1 - FV_i(V_e)) * LimP_i * \frac{P(D_i \geq d \mid d \leq D_i < t_f - t + d) * P(d \leq D_i < t_f - t + d)}{P(D_i \geq d)} \quad (7) \\
 &= (1 - FV_i(V_e)) * LimP_i * \frac{H_i(t_f - t + d) - H_i(d)}{1 - H_i(d)}
 \end{aligned}$$

where  $P(Z_{t_f} = i)$  is the probability of the transition entering state  $i$  from the other state.

Considering the above two situations, the probability under the situation that state  $j$  is similar to  $i$ , can be computed as Eq. (8).

$$\begin{aligned}
 PR_i &= P((V \geq V_e) \wedge (D_i \geq d)) = PR_{1a} + PR_{1b} \\
 &= (1 - FV_i(V_e)) * \frac{1 - H_i(t_f - t + d)}{1 - H_i(d)} + (1 - FV_i(V_e)) * LimP_i * \frac{H_i(t_f - t + d) - H_i(d)}{1 - H_i(d)} \quad (8)
 \end{aligned}$$

**Situation 2:** State  $j$  is different from  $i$ .

If state  $j$  is different from  $i$ , it means that there exist at least one transition during the interval from  $t$  to  $t_f$ . Similar to *Situation 1b*, the duration kept in state  $i$  before the first transition is shorter than  $d + t_f - t$ . Meantime, the probability of entering state  $j$  after several transitions during the interval from  $t_f$  to  $t$  is close to the steady-state occupancy probability  $LimP_j$  of state  $j$ . Then, the probability can be computed as Eq. (9).

$$\begin{aligned}
PR &= PR_2 = P((V \geq V_e) \wedge (Z_{t_f} = j) \wedge (d \leq D_i < d + t_f - t \mid D_i \geq d)) \\
&= P(V \geq V_e) * P(Z_{t_f} = j) * P(d \leq D_i < d + t_f - t \mid D_i \geq d) \\
&= (1 - FV_j(V_e)) * \text{Lim}P_j * \frac{H_i(t_f - t + d) - H_i(d)}{1 - H_i(d)}
\end{aligned} \tag{9}$$

where,  $P(Z_{t_f} = j)$  is the probability of transition entering state  $j$  from the other state.

### 5.2.2 Process of prediction

To predict the data transmission speed, we need to get the distribution of data transmission speed  $F_i(V)$  in state  $i$ , the distribution of duration  $F_{ij}(d)$  and transition probability  $P_{ij}$ . To calculate the above distributions and probability, for a continuous-time semi-Markov process, a set of backward Kolmogorov integral equations [13] are developed. While the approaches to solve these equations are feasible and can achieve accurate results in some situations, they perform poorly in many situations, for instance, when the rate of transitions is exponential with time. In applications like [12], a discrete-time semi-Markov process is utilized to achieve simplification. In this paper, we develop a discrete-time semi-Markov process to calculate distributions and probability.

**Definition 4** QoS-Related Contexts. QoS-related contexts is the observed result at time  $t_{ob}$ , and it can be defined as  $qc = \langle t_{ob}, v, st_{ob} \rangle$ , where  $t_{ob}$  is the time of the observation;  $v$  is the observed data transmission speed at  $t_{ob}$ ;  $st_{ob}$  is the state of  $v$ .

In the following part, we will introduce how to compute needed probability and distributions based on QoS-related contexts.

- Establishing Transition Set

Let  $QCS = \{qc_1, qc_2, \dots, qc_n\}$  be set of QoS-related contexts observed in the past. For each  $qc_i \in QCS$ ,  $qc_i$  is the QoS-related context observed at time  $qc_i.t_{ob}$ .

Given  $QCS$ , all the transitions from state  $i$  to  $j$  form a set  $TR_{ij} = \{(qc_u, qc_v, duration) \mid qc_u, qc_v \in QCS, \wedge qc_u.st_{ob} = i \wedge qc_v.st_{ob} = j \wedge i \neq j\}$ . In this paper, we use  $TR_{ij}[k]$  to signify an element  $(qc_u, qc_v, duration)$  in  $TR_{ij}$  where  $duration$  is the time between  $qc_u.t_{ob}$  and  $qc_v.t_{ob}$ . And  $duration$  can be computed as Eq. (10). We use  $TR_{ij}[k].pre$  to signify  $qc_u$  and  $TR_{ij}[k].post$  to signify  $qc_v$ .

$$TR_{ij}[k].duration = TR_{ij}[k].post.t_{ob} - QCS[l].t_{ob} \tag{10}$$

where,  $QCS[l]$  is an element in  $QCS$  and it satisfies the following conditions:  $QCS[l-1].st_{ob} \neq TR_{ij}[k].pre.st_{ob}$  and for any  $qc$  in  $QCS$ , if  $QCS[l].t_{ob} \leq qc.t_{ob} < TR_{ij}[k].pre.t_{ob}$ ,  $qc.st_{ob} = TR_{ij}[l].pre.st_{ob}$ .

- Calculating Transition Probability  $P_{ij}$

Based on the transition sets  $TR$ , the probability of transition from state  $i$  to  $j$  can be computed as Eq. (11).

$$P_{ij} = \frac{|TR_{ij}|}{\sum_k |TR_{ik}|} \quad (11)$$

where  $TR_{ij}$  is the set of transitions from state  $i$  to  $j$  according to  $QCS$ ;  $|TR_{ij}|$  is the number of elements in  $TR_{ij}$ .

- Computing Steady-State Occupancy Probability  $LimP_i$

To compute the steady-state occupancy probability, we firstly get the solution to Eq. (4), to obtain  $\pi$ . According to Eq. (4),  $\pi$  can be computed as Eq. (12).

$$[\pi_1 \quad \pi_2 \quad \pi_3] = [0 \quad 0 \quad 1] \cdot \begin{bmatrix} 1 & -P_{12} & 1 \\ -P_{21} & 1 & 1 \\ -P_{31} & -P_{32} & 1 \end{bmatrix}^{-1} \quad (12)$$

Let  $u = [u_1, u_2, \dots, u_5]$ . For any  $u_i$ ,  $u_i$  is the average duration spent in state  $i$ , and it can be computed as Eq. (13).

$$u_i = \frac{\sum_j \sum_k TR_{ij}[k].duration}{\sum_j |\{TR_{ij}[k]\}|} \quad (13)$$

After getting  $\pi$  and  $u$ , by Eq. (5), the steady-state probability of each state can be computed.

- Getting Distribution of Duration  $H_i(d)$

To compute the probability as in Eqs. (8) and (9), what is needed is the duration spent in the current state since last transition. Let  $TR_{final}$  be the final transition according to  $QCS$ , and  $TR_{final}.post$  be  $qc$ . Then, the state of  $qc$  is the current state. Let the current time be  $t$ . Then, the duration is  $t - qc.t_{ob}$ .

Let's discuss how to compute  $H_i(d)$ . To compute  $H_i(d)$ , we will compute  $F_{ij}(d)$  firstly.  $F_{ij}(d)$  is the probability of entering the state  $j$  from state  $i$ , under the condition that the duration in state  $i$  is less than  $d$ . Then,  $F_{ij}(d)$  will be the ratio of the number of elements in  $TR_{ij}$  with the duration that is below  $d$  to the number of all elements in  $TR_{ij}$ . It can be computed as Eq. (14). Then,  $H_i(d)$  can be computed as Eq. (15).

$$F_{ij}(d) = P\{T < d\} = \frac{|\{TR_{ij}[u] | u \in [1, |TR_{ij}|] \wedge TR_{ij}[u].duration < d\}|}{|\{TR_{ij}[u]\}|} \quad (14)$$

$$H_i(d) = \sum_j F_{ij}(d) * P_{ij} \quad (15)$$

- Getting Distribution of Data Transmission Speed  $FV_i(v)$

It also needs to compute the probability  $FV_i(V)$  of data transmission speed that is less than the maximum deviation speed  $V$  in state  $i$ . It can be computed as Eq. (16).

$$P(v < V \wedge Z = i) = FV_i(V) = \frac{|\{qc \mid qc \in QCS \wedge qc.st_{ob} = i \wedge qc.v < V\}|}{|\{qc \mid qc \in QCS \wedge qc.st_{ob} = i\}|} \quad (16)$$

After computing all the above probabilities, we can use Eqs. (8) and (9) to compute the probability of data transmission speed not below the maximum deviation.

## 6. DETAILS OF THE APPROACH

### 6.1 Triggering Re-selection Based on Failure Prediction

The algorithm presented in Algorithm 1 describes the proposed re-selection triggering approach. The basic idea is to predict the data transmission speed of each service in the composite one, and when the predicted speed below the threshold, the confident degree is also low (which means a failure will be happened at invocation time) and no replacement is predicted to be available to rescue such failure, the re-selection will be triggered. The prediction will be based on the semi-Markov model as describe in the former section.

**Algorithm 1** Re-Selection Triggering Algorithm

*servcie* TriggerReselection(*CS*, *ReplaceCS*) // *CS* is a composite service; *ReplaceCS* is a replacement one.

```

1 begin
2   for each service s in CS do
3     if  $\exists s_u, \text{QoSPrediction}(s, s_u, T(s), Q(s)) < \text{ThresholdP}(s)$  and  $\text{QoSPrediction}(s, s_r, T(s), Q(s)) < \text{ThresholdP}(s)$  then
// T(s) and Q(s) are the invoking time and expected QoS of service s respectively;
// sr is the reliable service
4       if  $\neg \exists RCS \in \text{ReplaceCS}$ , all services in RCS is predicted good then
5         return s;
6       else return null;
7   end

```

### 6.2 Re-Selection for Finding Replacement

That the original replacement is not available may affect the adaptive ability of a composite service. Thus, a re-selection is needed to find a new replacement. The problem of such re-selection is described in Eq. (17).

$$\begin{aligned} & \max_v F(CS_v), \\ & \text{s.t. } Q^t(CS_v) \leq Q_c^t \end{aligned} \quad (17)$$

where,  $Q_c^t$  is the runtime of original replacement.

This paper proposes a heuristic algorithm for solving the above re-selection problem. And the approach can be used for pipeline [4] composite service. The basic idea of the algorithm is: firstly, for each service class, keep constant number of part plans in

order to control the time complexity; then, based on the heuristic function to select the part plans which is most possible to be the optimized replacement; finally, if the heuristic process cannot find the replacement composite service, a process of random searching will be done in order to find the needed composite service through adjustment.

**Definition 5** Part Plan and Plan. For composite process  $CP$ , if  $\exists P = \{X_{ij} \mid X_{ij} = 1 \wedge \sum_{k \in S_i, k \neq j} x_{ik} = 0\}$  and  $|P| = |\{S_i\}|$ , we call  $P$  a plan of  $CP$ . If  $|P| < |\{S_i\}|$ , we call  $P$  a part plan of  $CP$ . If  $X_{ij} = 1$ ,  $X_{ij}$  is an assign of  $i$ . Given a part plan  $P$ , if  $|P| = i$  and for each  $sc_u$  ( $1 \leq u \leq i$ ),  $\exists X_{ij} \in P$ , we call  $P$  an  $i$  level plan.

**Definition 6** QoS Govern.  $P_i^1$  and  $P_i^2$  are the  $i$  level plans of abstract composite service ACS. If  $q^{(z)}(P_i^1) > q^{(z)}(P_i^2)$ , we call under factor  $z$ ,  $P_i^1$  governs  $P_i^2$ .

$\overline{P_i^*}$  and  $\underline{P_i^*}$  are two virtual  $i$  level plans of abstract composite service ACS. If for each  $P_i^u \in i$  level plans of ACS, and under each factor  $z$ ,  $\overline{P_i^*}$  governs  $P_i^u$  and  $\underline{P_i^*}$  governs  $\overline{P_i^*}$ , we call  $\overline{P_i^*}$  and  $\underline{P_i^*}$  the best and worst  $i$  level plans respectively. Since that  $\overline{P_i^*}$  and  $\underline{P_i^*}$  defines the quality range of all plans, they can be used as the referenced point to evaluate certain level plan. For the  $i$  level plan  $p$ ,  $h(p)$  is used to give score to  $p$ .

$$h(P) = \frac{\sqrt{(Q^{(t)}(\overline{P_i^*}) - Q^{(t)}(P))^2 + (F(\overline{P_i^*}) - F(P))^2}}{\sqrt{(Q^{(t)}(\overline{P_i^*}) - Q^{(t)}(\underline{P_i^*}))^2 + (F(\overline{P_i^*}) - F(\underline{P_i^*}))^2}} \quad (18)$$

where  $0 \leq h(p) \leq 1$ , and bigger is  $h(p)$ , more closer is  $p$  to the optimal plan. This paper uses  $h(p)$  as heuristic function.

**Algorithm 2** Finding Replacement Composite Service  
 $PCS$  Find\_HEU( $FS$ ,  $cons$ ,  $W$ ) //  $FS$  is the re-selected slice  
1 **begin**  
2 **for each**  $P_{i-1}^u \in P_{i-1}$  **do**  
3 **begin**  
4 **for each** service  $j \in FS[i]$  **do**  
5 let  $X_{ij} = 1$  to form  $P_i^u$  and compute QoS of  $P_i^u$ ;  
6 Get  $\overline{P_i^*}$  and  $\underline{P_i^*}$ ;  
7 **for each**  $P_i^u \in P_i$  **do**  
8 calculate  $h(P_i^u)$ ;  
9 keep top  $k$  in  $P_i$ ;  
10 **if**  $\exists P_i^u \in P_i$  **and**  $P_i^u.q^{(t)} \leq cons^{(t)}$  **then**  
11 **return**  $P_i^u$ ;  
12 **else**  
13 **begin**  
14 select  $P_i^u \in P_i$  with maximum of QoS;

```

15     return RandomSearch( $CP, P_i^u, cons, w$ );
16 end
17 end
18 end

```

## 7. EXPERIMENTS

This part will verify the effectiveness of the proposed approach.

Experiment 1 is used to test the effectiveness of the proposed performance predicting approach based on the semi-Markov model. Simulate test set of data transmission speed according to the Gaussian distribution  $N(10, 1)$ . The threshold of reliability is 0.9. The size of test set is 100000. Compare the relations among the predicted results, predicted interval and the observed interval between two neighboring contexts. Table 2 gives the result ( $O_Q$  is the observed interval between two neighboring QoS-related contexts in the test set;  $N$  is the number of predictions;  $R$  is the average accurate rate of the predictions which can be computed as the ratio of the number of predictions that is right to the number of predictions;  $I$  is the predicted interval and the unit of  $I$  is second).

**Table 2. Semi-Markov based predicted result.**

$O_Q$	0.5s			0.1s			0.05s		
$I$	10s	60s	180s	10s	60s	180s	10s	60s	180s
$N$	300	300	300	200	200	200	150	150	150
$R\%$	95	86	80	97	93	83	98	95	92

Table 2 shows that if the observed interval between two neighboring contexts is smaller than 0.1s, although the predicting interval is relatively bigger (*e.g.* 60s), the predicted result (accuracy is 93%) is acceptable. If the observed interval is smaller (*e.g.* 0.05s), although the predicted interval is bigger (*e.g.* 180s), the accuracy of the predicted result (*e.g.* 92%) is also acceptable. Thus, through minimizing the observation interval, the accuracy of prediction result can be improved.

Experiment 2 is used to test the performance of the heuristic algorithm for finding the replacement. Randomly generate 200 scenarios, in each of which the number of service classes may be 5, 10, 20 and 40 respectively, the number of services for each service class is 10 and the number of part plans kept in each level is 20. Randomly generate 200 scenarios, in each of which the number of service classes is 20, the number of services for each service class are 10, 20, 50 or 100. Compare the runtime in Fig. 4.

From Fig. 4, the runtime of the proposed algorithm is less than that of the exhaustive searching algorithm.

Experimentation 3 is to test the effectiveness of the EX\_QoS model. We use satisfied degree of execution duration to evaluate the effectiveness of EX\_QoS. Eq. (19) gives how to compute the satisfied degree. Give 200 scenarios where the average amount of data transmission between services can be 0, 10, 50, 100 and 200 bit. Compare the satisfied degree of the composite services backed up according to the QoS model used in [2] and the EX\_QoS model. Fig. 5 shows the result.

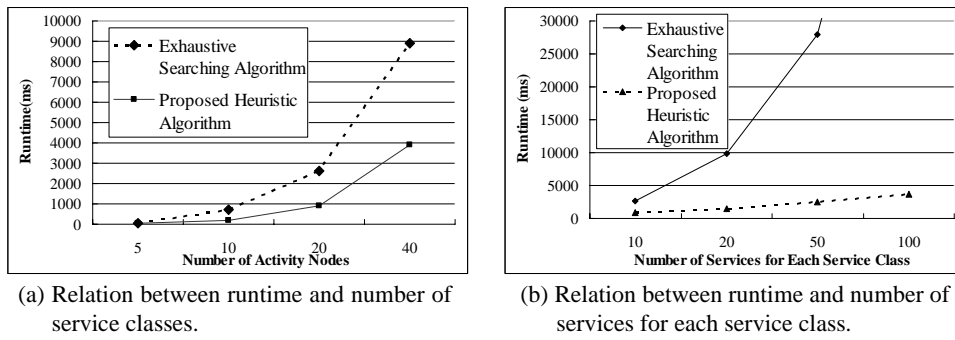


Fig. 4. Runtime comparison.

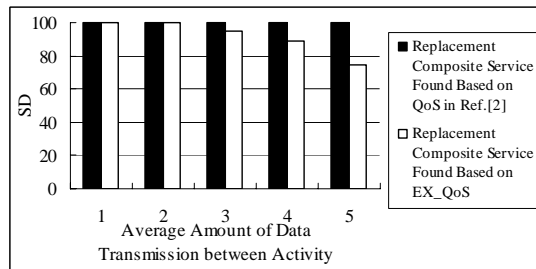


Fig. 5. Relation between the SD and average amount of data.

$$SD(CS, cons^{(t)}) = \begin{cases} \frac{Q^{(t)}(CS)}{cons^{(t)}}, & Q^{(t)}(CS) < cons^{(t)} \\ 1, & \text{else} \end{cases} \quad (19)$$

From Fig. 5, SD of the composite service backed up based on EX\_QoS is higher than the one based on QoS in [2]. EX\_QoS is effective in selection in decentralized service composition.

Experimentation 3 is to test the interrupting time caused by re-selection process. Randomly generate 10 scenarios with  $O_Q = 0.1s$ ,  $I = 60$ ,  $k = 20$  and the threshold of failure probability is 0.9. Compare the extra delay, the result of which is shown in Fig. 6.

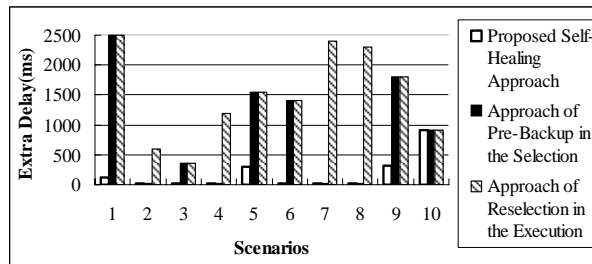


Fig. 6. Comparison of extra delay.

Fig. 6 shows the extra delay caused by the proposed approach is always the shortest among the three approaches. The reason is that the proposed approach is an improvement of the traditional pre-backup approach. By performance prediction, services which will incur a QoS violation can be predicted and if the corresponding replacement composite service is not available, the reselection process aiming at finding a replacement will be more likely to finish before the invocation of the service and thus the composite service can adjust itself from the failure as quickly as possible. Besides this, through the re-selection, the availability of the replacement can be preserved, and the composite service can use the replacement to rescue from the failure. Then, the extra cost will be the minimum.

## 8. CONCLUSION

In order to adapt to dynamic property of services, we propose an adaptive approach for web service composition. The contribution of this paper includes: (1) an adaptive approach for web service composition is proposed; (2) an EX-QoS model for achieving global optimization in decentralized service composition is presented and corresponding heuristic algorithm is given; (3) a method of performance prediction is introduced in order to make the re-selection for updating the replacement complete as early before the invocation of the failed service as possible.

In the future work, we will study on how to do the re-selection for a complex composite service. The prediction approach will be studied more and the proposed adapting approach will be put into practical applications of web service composition.

## REFERENCES

1. N. Milanovic and M. Malek, "Current solutions for web service composition," *IEEE Internet Computing*, Vol. 8, 2004, pp. 51-59.
2. L. Z. Zeng and B. Benatallah, "QoS-aware middleware for web services composition," *IEEE Transactions on Software Engineering*, Vol. 30, 2004, pp. 311-327.
3. D. Ardagna and B. Pernici, "Global and local QoS guarantee in web service selection," *International Journal of Business Performance Management*, Vol. 1, 2005, pp. 233-243.
4. T. Yu, Y. Zhang, and K. J. Lin, "Efficient algorithms for web services selection with end-to-end QoS constraints," *ACM Transactions on the Web*, Vol. 1, 2007, pp. Article 6.
5. M. C. Jaeger, G. Muhl, and S. Golze, "QoS-aware composition of web services: An evaluation of selection algorithms," *International Conference on Cooperative Information Systems*, 2005, pp. 646-661.
6. J. Cardoso, A. P. Sheth, J. A. Miller, and K. Kochut, "Modeling quality of service for workflows and web service processes," *Journal of Web Semantics*, Vol. 1, 2004, pp. 281-308.
7. H. Jin, H. H. Chen, J. Chen, P. Kuang, L. Qi, and D. Zou, "Real-time strategy and practice in service grid," in *Proceedings of the 28th Annual International Computer Software and Applications Conference*, 2004, pp. 161-166

8. G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani, "QoS-aware replanning of composite web services," in *Proceedings of the IEEE International Conference on Web Services*, 2005, pp. 121-129.
9. F. Casati, S. Ilnicki, and L. Z. Jin, "Adaptive and dynamic service composition in eFlow," *Advanced Information Systems Engineering*, 2000, pp. 13-31.
10. B. Benatallah, Q. Z. Sheng, and M. Dumas, "The self-serv environment for web services composition," *IEEE Internet Computing*, Vol. 7, 2003, pp. 40-48
11. T. Yu and K. J. Lin, "Adaptive algorithms for finding replacement services in autonomous distributed business processes," in *Proceedings of the 7th International Symposium on Autonomous Decentralized Systems*, 2005, pp. 427-434.
12. G. Chafle, K. Dasgupta, A. Kumar, S. Mittal, and B. Srivastava, "Adaptation in web service composition and execution," in *Proceedings of IEEE International Conference on Web Services*, 2006, pp. 549-557.
13. M. Malhotra and A. Reibman, "Selecting and implementing phase approximations for semi-Markov models," *Communication Statistics-Stochastic Models*, Vol. 9, 1993, pp. 473-506
14. Y. Altinok and D. Kolcak, "An application of the semi-Markov model for earthquake occurrences in north Anatolia turkey," *Journal of the Balkan Geophysical Society*, Vol. 2, 1999, pp. 90-99
15. Z. B. Fang and B. Q. Miao, *Stochastic Process*, University of Science and Technology of China Press, China, 2007.



**Lei Yang (楊雷)** is a lecture in the college of Information Science and Technology at Northeastern University, Shenyang, China. He received his Ph.D. degree from Northeastern University, China in 2007. His research interests include service composition and service oriented computing.



**Yu Dai (代鈺)** receives her Ph.D. degree from the College of Information Science and Technology at Northeastern University, Shenyang, China in 2008. She is a lecture in the College of Software at Northeastern University. Her main research interests include service composition and service oriented computing.



**Bin Zhang (張斌)** is a professor in the college of Information Science and Technology at Northeastern University, Shenyang, China. He is a member of CCF. He received his Ph.D. degree from Northeastern University, China in 1997. His current research interests are service oriented computing and information retrieval.