Geographic Location-Based Network-aware QoS Prediction for Service Composition

Yuanhong Shen, Jianke Zhu, Xinyu Wang*, Liang Cai, Xiaohu Yang, and Bo Zhou
College of Computer Science, Zhejiang University, Hangzhou, 310027, China
Email: {yhshen, jkzhu, wangxinyu, leoncai, yangxh, bzhou}@zju.edu.cn

Abstract—QoS-aware service composition intends to maximize the global QoS of a composite service while selecting candidate services from different providers with local and global QoS constraints. With more and more candidate services emerging from all over the world, the network delays often greatly impact the performance of the composite service, which are usually not easy to be collected before the composition. One remedy is to predict them for the composition. However, new issues occur in predicting network delay for the composition, including prediction accuracy and on-demand measures to new services, which affect the performance of network-aware composite services. To solve these critical challenges, in this paper, we take advantage of the geographic location information of candidate services. We propose a network-aware QoS (NQoS) model for the composite service. Based on that, we present a novel geographic location-based NQoS prediction approach before composition, and a NQoS re-prediction approach during the execution of the composite service. Extensive experiments are conducted on the real-world dataset collected from PlanetLab. Comparative experiment results reveal our approach facilitates to improve the prediction accuracy and predictability of the NQoS values, and increase global NQoS of the composite service while ensuring its reliability constraints.

Keywords-QoS-aware service composition; geographic location; network; performance prediction; re-selection

I. INTRODUCTION

In Service-Oriented Computing (SOC), service composition aims to select and integrate the different existing candidate services (CSs) from the distributed environment into a composite service [1]. As the number of CSs with equivalent functionalities from different service providers grows, Quality of Service (QoS) [2] is employed to distinguish these CSs. QoS-aware service composition aims to find the best combination of services that maximize the global QoS of the composite service while satisfying local and global QoS constraints defined by users.

With the rapid development of cloud computing, more and more services are published in open environments across Internet [3,4], where the composite service is often invoked as a decentralized model [5,6]. During the execution of a composite service, messages or large amount of data like videos often need to be routed between adjacent services across intercontinental networks, instead of between the user and each service. Therefore, network delays (reflected by the network latency [4] and bandwidth [5]) greatly affect the performances of the composite services and should be considered in the QoS of CSs. Usually, it is uneasy to collect these delays for CSs before the composition. One remedy is to predict them [4,6].

To increase the performance of the network-aware composite service, the following issues need to be considered in predicting network delay for the composition.

1. Prediction Accuracy. Accurate prediction of network delays of CSs influences the optimal selection of CSs for the composite service, and then affect the performance of the composite service. Most existing prediction methods are based on a great amount of historical network delay information. For example, to predict network latency between a pair of CSs, most of the current prediction methods, e.g., landmark-based approaches [7] or network structured approaches [8] require many existing network latency values between many other pairs of network nodes, and apply coordinate computing or other methods to calculate the latency value between them. With the increasing number and distribution of services in cloud computing environment, more such information is required. Thus, the prediction accuracy may be very low or the predicting procedure cannot be carried out successfully.

2. On-demand Measures to New Services. Most existing prediction methods require multiple on-demand measures to each new CS. For example, to predict network latency and bandwidth between a pair of new CSs, most of the current prediction methods, like [7,8,10] require multiple on-demand measures for latency values from landmarks or other network nodes to each of them to determine its accurate network position. With the increasing number and distribution of newly discovered CSs, the overhead of measuring may increase dramatically. In addition, performing many measures to new services increases implementation complexity and may trigger intrusion-detection alerts.

Geographic location information can be naturally utilized to indicate the network delay between CSs. This idea is confirmed by our experiments [3] to some extent in at least two aspects. In one aspect, it is obvious to see a geographically long route between CSs are more likely to have longer network latency than a geographically short route. In another aspect, we have confirmed that two geographically similar routes between CSs are more likely to have similar network latency values, since geographically similar routes

*http://www.iamresearcher.com/blog/post/66/
*Xinyu Wang is the Corresponding Author
are more likely to share the same IT infrastructures (network workloads, routers and so on). Thus, geographic location information may be utilized to increase the prediction accuracy. In addition, compared with other network position-based network latency prediction approaches, the geographic location of each CS can be easily determined, since no additional on-demand measures are required.

Based on the above analysis, we utilize the geographic location information to predict the network delays for CSs, and integrate our prediction approach to the service composition, consisting of the following three contributions:

1) We propose a network-aware QoS (NQoS) model for network-aware composite service (Section III). The NQoS has included both traditional QoS and network-aware quality attributes (i.e. network latency and bandwidth). In addition, QoS constraints for the composite service are also taken into account.

2) We present a novel geographic location-based NQoS prediction approach for new CSs based on NQoS values of executed services and geographic location information of routes between executed services (Section IV). Using our prediction approach, more accurate NQoS values can be forecasted for CSs before composition.

3) We propose a NQoS re-prediction approach to adjust the predicted NQoS values with the change of context information during the execution of the composite service (Section V). Therefore, more accurate NQoS values of CSs can be forecasted before execution, and the performance of the composite service can be optimized continuously.

Our evaluations are conducted based on our real-world network performance dataset collected from PlanetLab. In our evaluation, we show that our prediction and re-prediction approach outperform other existing approaches especially with few historical NQoS values and no on-demand measures, and therefore optimize the performance of the composite service.

II. RELATED WORK

In SOC, the problem of QoS-aware service composition was often modeled as constraint optimizing problems and solved by (mixed) linear inter-programming [2], genetic algorithms [11-13](GAs) or hybrid approaches [14]. In recent approaches, the service execution contexts, including bandwidth [15], policy context [16], device context [13], etc, are also considered in the QoS models for the composition problems. In cloud computing environments, point-to-point network delay [17], including network latency between adjacent services [18] is also considered in service compositions. In their models, the QoS of a service is input-dependent, and the selection or execution of one service will influence the QoS of another service in the composite service. Compared with these works, our QoS model focuses on the network-aware services in SOC and takes both network latency and bandwidth into account, and also considers the reliability constraints of tasks in the composite service.

Based on the QoS model, many composition mechanisms have been proposed to increase the global QoS or availability of composite services in undetermined or dynamic environments, including service backup [18, 19], service re-selection [12, 20], network delay prediction [4, 6], etc. Our approach also adopts these composition mechanisms in order to increase the efficiency, availability and scalability of network-aware service composition.

Current network delay prediction methods can be divided into coordinate computing methods and non-coordinate methods. In coordinate computing methods, the network position for each network node is iteratively computed based on multiple on-demand measures or historical latency information. Then the network latency between any network nodes can be predicted using the Euclidean distance between the network locations of the nodes. Among the coordinate computing method, triangulated heuristic [7], global network positioning (GNP) [7], and Vivaldi [9] are three widely employed approaches. The approach in [4] adopts Vivaldi to measure network distance between different nodes for service composition, and choose the services that can fulfill the current task and are close to the network location of their preceding service. However, it has some weaknesses: 1) as shown in our experiment below, determining accuracy network locations requires many measures to new services; 2) it focuses on reducing network latency between services, and neglect some other QoS and QoS constraints (e.g., bandwidth and reliability constraints). The commonly used non-coordinate methods includes Internet Iso-bar [21], iPlane [8], OASIS [22], etc. Iso-bar [21] and iPlane [8] require additional hosts or a large number of on-demand measures for local latency information to calculate required global latency values. The approach in [10] is a non-coordinate method for bandwidth prediction, where the bandwidth from each network node to each landmark is used to determine each node’s bandwidth vector. Then the bandwidth vectors of two nodes are used to estimate the bandwidth between them in both ways. OASIS [22] maps different portions of the Internet in advance (based on IP prefixes) to the geographic coordinates, and stores the geographic coordinates of the replica closest to each prefix it maps. Then, the geographic location information can be used to predict the network latency between users and replicas. We also used geographic location information for latency prediction. In our approach, we predict the latency value for a new pair of CSs by composing the latency values of historical geographically similar routes between CSs, instead of mapping portions of Internet in advance. In service QoS prediction areas, the geographic or network locations of services are incorporated into collaborated filtering methods [23-25]. However, it seems that these methods can only be utilized to the Web service response time prediction between one service to each personalized service user, not to the QoS prediction between
service pairs.

III. NQoS Model for Network-aware Composite Service

In QoS-aware service composition, one or more CSs are selected to bind each task in the workflow in order to implement a composite service defined by users. The object of the service composite is to maximize the global QoS of the composite service. Therefore, the QoS model of the composite service needs to be defined. Since some quality attributes (i.e., network latency and bandwidth) between adjacent services have not been considered in previous QoS models, we introduce a network-aware QoS (NQoS) model for network-aware composite services.

We first model the NQoS for each service in the composite service. Compared with traditional QoS models, the execution time of a service in our NQoS model includes its inner execution time and network delay between it and its next service. The network delay between two services: $S_a$ and $S_b$ ($Q^{dt}(S_a, S_b)$) can be computed as:

$$Q^{dt}(S_a, S_b) = \frac{A(S_a, S_b)}{B(S_a, S_b)} + l(S_a, S_b) \quad (1)$$

where $A(S_a, S_b)$ is the amounts of data to be transmitted from $S_a$ to $S_b$, and $B(S_a, S_b)$ is the average bandwidth between $S_a$ and $S_b$, $l$ is the network latency value. If $S_a$ equals $S_b$, $\frac{A(S_a, S_b)}{B(S_a, S_b)}$ is zero. As in [4, 17], QoS attributes in the network-aware composite service mainly include the execution time, cost, reliability, etc. We also model the cost for each service.

Then we model the NQoS constraints for each task (local constraint) in the composite service. Since other QoS constraints can be easily considered, we focus on modeling the reliability constraints. Since the service resource is rich and network disconnections often occur between service pairs, we employ a service backup mechanism: active strategy [26] for each task in the workflow to ensure the reliability constraints while reducing the overall execution time. As shown in Figure 1, each task in the workflow will associate several services as replicas, which is selected from CSs. Note that all the associated replicas for a task $T_i$ are executed in parallel. Moreover, their execution results will be sent to the master replica of $T_i$’s next task: $T_j$ that takes the first received required result as the final result for $T_i$ and re-direct $T_i$’s result to $T_j$’s other replicas. Once receiving the required result, the replica will be invoked and executed.

After that, we model the execution time and cost for each task in the workflow [26, 27] while binding their replicas. Except the network delays, we assume all the other QoS values can be collected directly.

Finally, we model the NQoS for the composite service by aggregating the NQoS of each task, which can be calculated according to the workflow structures and user defined weights over the different NQoS attributes [27].
a grid based on the geographical map, and divide the map into multiple buckets. The length and width of each bucket are represented by the differences of longitude and latitude, which are fixed in our empirical evaluation. Then, each node is mapped into a bucket in the map represented by node.bucket. We summarize the process of acquiring the similar measured paths for \( p_a \) into Algorithm 1, also as shown in Figure 2.

Algorithm 1 Acquire the similar measured paths for \( p_a \).

Input: \( p_a \).

Output: \( CP(p_a) \).

1: if the nodes in \( p_a \) are unknown then
2: add a set of virtual nodes (nodes) with different geographic locations in \( p_a \), the geographic distance of each adjacent nodes is \( maxDis \);
3: end if
4: get all the nodes: \( nodes \) in \( p_a \);
5: for each node do
6: fetch a set of measured paths: \( Set(node.bucket) \), such that for each \( p \) in the set, \( \exists \ p.node.bucket \ = \ node.bucket \);
7: \( CP(p_a) \) \( \leftarrow \ CP(p_a) + Set(node.bucket) \);
8: end for
9: return \( CP(p_a) \);

The geographic similarity \( sim \) between two paths \( p_a, p_b \) is defined as below:

\[
sim(p_a, p_b) = \left( \min \left( d(p_a, s, p_b, s) + d(p_a, d, p_b, d), d(p_a, s, p_b, d) \right) + \sum_{n_a \in p_a, n_b \in p_b} \min(d(n_a, n_b)) ||n_a||, ||n_b|| \geq |n_b| \right)\]

\[
\quad n_a \neq p_a.s, n_b \neq p_b.s, n_a \neq p_b.s, n_b \neq p_a.s, n_a \neq p_b.d, n_b \neq p_a.d
\]

(2)

where \( n_a, n_b \) refers to the node in the path, and \( d(n_a, n_b) \) represents the geographic distance between \( n_a \) and \( n_b \) on the surface of the earth. \( |n_a| \) and \( |n_b| \) denote the number of nodes with different IPs in \( p_a \) and \( p_b \).

If the nodes along the path \( p_a \) are unknown, then \( sim(p_a, p_b) \) is calculated as follows:

\[
sim(p_a, p_b) = \left( \min \left( d(p_a, s, p_b, s) + d(p_a, d, p_b, d), d(p_a, s, p_b, d) \right) + \sum_{n_a \in p_a} d_1(n_a, p_b) ||n_b||, n_a \neq p_a.s, n_b \neq p_a.d \right)
\]

(3)

where \( d_1 \) refers to the shortest distance from a node to a line segment on the surface of the earth.

Then, we use the bilateral filtering [28] to predict the latency of \( p_a (p(p_a, l)) \):

\[
p(p_a, l) = \frac{\sum_{p \in CP(p_a)} p.l \cdot w(p_a, p)d(p_a)/d(p)}{\sum_{p \in CP(p_a)} w(p_a, p)}, \quad sim(p_a, p) < threLat
\]

where \( w(p_a, p) = \exp(-\frac{\text{sim}(p_a, p)}{2\sigma^2}) \). Therefore, the latency value of the geographically more similar path to \( p_a \) is set a larger weight in prediction. \( d(p_a)/d(p) \) is an adjustment factor. If the route of \( p_a \) is known, \( d \) refers to the number of nodes along the path; otherwise, \( d \) refers to the geographic distance between end point CSs of the path. Note that \( threLat \) can be adjusted, which is initially set to \( minthre \). We increase it by \( threLat = threLat \times 2 \) until \( p(p_a, l) \) can be calculated or the maximal threshold \( threLat > maxThre \) reaches.

Usually, a geographically long path will be more likely to have large latency than the geographically short path. Therefore, we design another geographic location-based latency prediction method named as “Geonaive”. In this method, to predict the latency between a pair of CSs, we first discover the historical latency data of CS pairs with a geographic distance between them. Then, we calculate the average latency per distance based on the discovered data. Finally, the predicted latency can be estimated based on the latency per distance and the geographical distance between the two CSs for prediction.

B. Bandwidth Prediction

The bandwidth data are predicted between CS pairs. We try to improve the landmark-based approach [10] by removing the dependency of on-demand measures. As in [10], the bandwidths for a set of indirect paths joining the CSs via a landmark need to be acquired to predict the bandwidth between a pair of CSs. The indirect paths having more links common with the path to be predicted are assigned more weights.

However, it is difficult to predict the bandwidth between CSs with very few accumulated bandwidth data. Typically, such indirect paths cannot be detected. Since two geographically close paths are more likely to share the same IT infrastructures, we replace the indirect paths by \( p_a \)’s candidate similar measured paths \( CP(p_a) \) to predict its bandwidth. Therefore, the predictability can be improved. Instead of using on-demand measures, we choose \( p \)'s end
point CSs \((p.s \text{ and } p.d)\) to be the landmarks for each \(p\) in \(CP(p_a)\). Then, the predicted end-to-end bandwidth for \(p_a\) \((p(p_a,B))\) can be calculated as below:

\[
p(p_a,B) = \sum_{p \in CP(p_a)} w(p) \cdot p.B, \text{sim}(p_a, p) < \text{threBand}
\]

where \(p.B\) is the bandwidth of \(p\), and \(w(p)\) is a normalized weight assigned to \(p\). \(\text{threBand}\) is the threshold for \(p\). As in \([10]\), \(w(p)\) is dependent on the two factors: the delay closeness between the landmarks and \(p_a\)'s end point CSs, and the geographic similarity between \(p\) and \(p_a\). For each \(p\), we express these two factors \(F(p)\) as:

\[
F(p) = \begin{cases} 
\left(\frac{R_{\min} \cdot \text{sim}(p_a, p)}{R(p) \cdot \text{sim}(p,p_a)} \right)^{\alpha} & \text{if } \text{sim}(p,p_a) > 0, \\
\left(\frac{R_{\min}}{R(p)} \right)^{\alpha} & \text{if } \text{sim}(p,p_a) = 0.
\end{cases}
\]

where \(R(p) = \min\{l(p.s,p_a.s), l(p.s,p_d), l(p.d,p_a.s), l(p.d,p_a.d)\}\). \((x,y)\) represents the historical network latency between \(x\) and \(y\), and \(R_{\min} = \min_{p \in CP(p_a)} R(p)\). \(\text{sim}(p,p_a) < \text{threBand}\). Thus, \(w(p)\) can be calculated as follows:

\[
w(p) = \frac{F(p)}{\sum_{p \in CP(p_a)} F(p)}, \text{sim}(p,p_a) < \text{threBand}
\]

C. Service Selection Algorithm

The service selection is to maximize the global NQoS of the composite service while selecting CSs for each replica to meet the local NQoS constraints of each task and global NQoS constraints for the composite service. This is an NP problem with multiple constraints and huge search space. We adopt an existing repair genetic algorithm (GA) \([11]\) to solve this problem since the GA seems to be suitable to solve such problems.

V. NQoS RE-PREDICTION

With the execution of a composite service, more and more NQoS data will be accumulated. Since the prediction accuracy increases according to the amount of accumulated NQoS information, it is necessary to study the re-prediction based on monitoring service execution contexts during the execution of the composite service. Given \(S_{i_1}, S_{i_2},\ldots, S_{i_n}\), for \(T_i\), the NQoS re-predictions for the network latency and bandwidth are performed between the following pairs of services: \((S_{i_k}, S_{j_1})\left[\begin{array}{c}n_i \text{ } k=1 \ldots \text{ } n_i \text{ } k=1 \ldots \text{ } n_i \end{array}\right]\) and \((S_{i_1}, S_{i_k})\left[\begin{array}{c}n_i \text{ } k=2 \ldots \text{ } n_i \text{ } k=2 \ldots \text{ } n_i \end{array}\right]\), where \(T_j\) is \(T_i\)'s subsequent task, and \(n_i\) is the replica number for each task.

A. Re-prediction Algorithm

To predict/re-predict NQoS between two services represented by the path \(p_a\), the most similar measured path \(\text{minDis}(p_a)\) to \(p_a\) are recorded in our proposed Geography approach. During the execution of a composite service, new service execution contexts are monitored continuously for \(p_a\) including discovering new NQoS of the measured paths similar to \(p_a\). If \(\text{minDis}(p_a)\) is updated, the NQoS values of \(p_a\) will be re-predicted, and the new predicted values will be recorded.

B. Service Re-selection Algorithm

To optimize the performance of the composite service continuously in dynamic environments, service re-selection is necessary. We propose an algorithm during the execution of the composite service to determine whether to invoke the service re-selection algorithm. The algorithm is based on monitoring variations of network latency and bandwidth values for the selected services as a result of performance re-predictions during the execution of the composite service. It is used to determine whether to invoke the service re-selection algorithm during the execution of the composite service. Once the performance re-prediction procedure is completed between a service pair in the most recent time, the algorithm is invoked. The algorithm employs \(p_c\) to denote the current predicted values between a service pair (e.g., \(p_c(l(S_{i_k}, S_{j_k}))\) and \(p_c(B(S_{i_k}, S_{j_k}))\)) and employs \(p_l\) to denote the predicted values between a service pair when the most recent service re-selection or selection algorithm is invoked (e.g., \(p_l(l(S_{i_k}, S_{j_k}))\) and \(p_l(B(S_{i_k}, S_{j_k}))\)). Once \(\text{thre}\%\) proportion of service pairs meet the condition that \(p_l(l(S_{i_k}, S_{j_k})) < \text{thre} \) or \(p_l(B(S_{i_k}, S_{j_k})) < \text{bThre}\), a service re-selection procedure will be invoked.

In this algorithm, \(\text{thre} \), \(\text{bThre} \) and \(\text{thre} \) are used to determine what magnitude of context change will lead to a service re-selection process. These parameters can be adjusted to balance the cost of service re-selections and increase of global NQoS for the composite service.

The service re-selection procedure is acted on all the tasks to be executed in the composite service. We also adopt the repair GA for the service re-selection.

VI. EVALUATION

A. Data Collection and Setup

To make fair comparison, the real-world cross-regional network performance datasets are required to evaluate the prediction accuracy. In this paper, we selected 622 distributed nodes around the world from PlanetLab. Moreover, we collected 403,916 network latency values by sending the ping messages and 18,519 bandwidth values by sending big data packages from one node to the other between the node pairs in our collection\(^3\). In addition, the IP addresses (nodes) along the path between each node pair are also recorded. As the standard deviation of bandwidths is usually high, we measure the bandwidth for each node pair several times and use the average bandwidth in our experiments.

\(^3\)http://www.iamresearcher.com/data/data/210/
We conducted our empirical evaluation on a PC with Intel Core2 Quad 2.19GHz CPU and 2GB memory. We adopt Java language to simulate that each selected node from PlanetLab contains one deployed CS. We assume the network latency and bandwidth values between pairs of nodes equals the collected values from PlanetLab. We use the workflow in Figure 1 for service composition and set the ranges of parameters in our experiment. We use the database file named “GeoLiteCity.dat” to map each CS and node along the paths between each CS pair to a geographical point.

To evaluate the performances of our NQoS prediction approaches, we employ three metrics: Mean Absolute Error (MAE) \[24\], Median Relative Error (MRE) \[24\], and predictability, which are defined as follows:

**MAE:** This metric is employed to average prediction accuracy. \[
MAE = \frac{1}{N} \sum_{a} |p(p_{a}, x) - p_{a}, x|,
\]
where \(p(p_{a}, x)\) and \(p_{a}, x\) denote the predicted and measured network performance values, respectively, for a path: \(p_{a}\), \(N\) is number of predicted values.

**MRE:** This metric is median value of all the relative error values. \[
MRE = \text{Median}_{a} \left(\frac{|p(p_{a}, x) - p_{a}, x|}{p_{a}, x}\right),
\]
which means 50% of the relative errors are below MRE.

**Predictability:** It is used to measure successful rate of predictions of a prediction approach. For predicting the bandwidth values of a path: \(p_{a}\), the set of geographically similar measured paths: \(CP(p_{a})\) need to be discovered. If such measured paths cannot be found, the prediction procedure cannot proceed, and the prediction is failed, otherwise, the prediction is successful.

**B. Performance on NQoS Prediction**

1) **Network Latency Prediction:** The pairwise network latency among CSs can be represented by a \(622 \times 622\) matrix. For each testing case, we randomly remove the data in the matrix and vary the density of matrix as 0.1%, 0.2%... The data remains in the matrix are assumed to be latency values, as long as the nodes along the routes between CS pairs, and the data removed from the matrix are assumed to be latency values between CSs that are need to be predicted. In each testing case, we randomly choose one latency value to be predicted, and compare our network latency prediction approach with other existing methods. We compare our proposed approaches with several state-of-the-art methods including Vivaldi \[9\], triangulated heuristic \[7\], iPlane \[8\], which can predict the network latency using only the historical observations between CSs without extra on-demand measures. Since we do not deploy landmarks in our experiment, in the Triangulated heuristic approach, we assume each end point CS of each measured path can be a landmark. For the iPlane method, we collect the extra latency values between adjacent nodes along the path between each CS pair. In our Geonaiive approach, historical latency values of CS pairs are discovered if one CS in each pair equals one CS in the CS pair for prediction. Thus, its predictability can be very high.

In all our experiments, we set the bucket length to 0.1175, bucket width to 0.1505, \(min\) \(\text{thre}\) to 1, \(max\) \(\text{thre}\) to 1000, \(max\) \(\text{Dis}\) to 20 and \(\sigma\) to 10. Table I shows the performance comparisons on the network latency prediction. When performing Geography approach, we set two situations where the routes between CSs for prediction are known (Geography(known)) or unknown (Geography(unknown)). All the prediction results are calculated from 200 different testing cases.

As depicted in Table II, Geography approach outperforms other approaches in increasing prediction accuracy with different matrix densities, and also performs better than other approaches in increasing the predictability (pred.) when the matrix density is very low. When the matrix density is very low, the Vivaldi approach faces the “cold start” problem and its prediction accuracy is based on the setting of initial values in its model to a large extent, thus, may be very low. When the matrix density is above 2%, the predictability of Vivaldi approach is 100%, which means the paths from other network nodes to each CS in all the CS pairs have been measured as least once, in that case, the prediction accuracy of Vivaldi approach is still far below the accuracy of Geography. It shows the effect of incorporating geographic location information of pairs of CSs into network latency predictions, which increases prediction accuracy and predictability. The prediction accuracy and predictability increases according to the increase of matrix density in our method. In addition, if the routes between pairs of CSs for prediction (more geographic location information) are known before composition, the prediction performance will be more improved.

2) **Bandwidth Prediction:** The bandwidth values are collected from pairs of CSs from 350 of the 622 CSs. We construct a similar matrix and perform the bandwidth prediction procedure. If the bandwidth value between a pair of CSs is known, then the latency value between them is also known. We let the routes between pairs of CSs for prediction be unknown, and test the performance of our prediction approach with different values of \(\text{thre}\) \(\text{Band}\) and get the results as shown in Table III. When \(\text{thre}\) \(\text{Band}\) equals zero, our approach resembles the approach in \[10\]. However, when \(\text{thre}\) \(\text{Band}\) equals zero, no indirect paths can be collected based on our little historical information according to our testing. Thus, our approach can improve predictability in most cases. Each result is also acquired from 200 different testing cases.

According to the results in Table III, the prediction accuracy of our approach increases according to the decrease of the value of \(\text{thre}\) \(\text{Band}\) in most cases, while the predictability in-
increases according to the increase of the values of threBand. It shows the effect of using geographic location information in increasing prediction accuracy. In addition, a user can choose different values of threBand based on its trade-off between accuracy and predictability, which makes our approach flexible.

3) Effects in the Performance of Composite Services before Execution: The accuracy of NQoS prediction may influence the optimal selection of CSs for the composite service, and then influence the performance of the composite service. Thus, this experiment intends to analyze our NQoS prediction approach in the performance of composite service before execution. Based on the prediction approaches, we compare the performances (actual global NQoS) of the composite service after performing five service selection procedures using the five comparing network latency prediction approaches for the same set of pairs of CSs in the networks. All the prediction approaches are based on the same records of bandwidth values in the matrix. We use our bandwidth prediction approach to acquire required bandwidth values between pairs of CSs in all the five service selection procedures. We assume all the routes between pairs of CSs for prediction be unknown and each result is averaged over five different testing cases in all the experiments below.

We compare the actual global NQoS of the composite service as the simulated number of tasks in the workflow and the number of simulated CSs for each task varies. To vary the number of tasks in the workflow, we construct different number of blocks of the workflow in Figure 1 with different tasks. In this way, we can construct different workflow whose task number ranges from 1×7 to 7×7. We set the matrix density to 1% in NQoS prediction in this experiment. We set the value of threBand to 200 in this and the following experiments. We let the routes between CSs for prediction be unknown and each result is averaged over five different testing cases in all the experiments below.

As depicted from Figure 3, Geography approach has more effect in increasing the actual global NQoS of the composite service than other approaches. With the increase of number of CSs for each task, the benefit of Geography approach is more obvious.

C. Effects of the NQoS Re-prediction in the Performance of Composite Services after Execution

This experiment intends to analyze our NQoS re-prediction approach in the performance of composite service after execution. Based on the re-prediction approaches, we compare the performances (actual global NQoS) of the composite service after execution after performing five service re-selection procedures using the five comparing network latency re-prediction approaches. With the execution of a composite service, the newly predicted NQoS values between newly selected services are recorded, thus, our service re-selection algorithm may be invoked at times after performing the re-prediction procedure and the service selection algorithm.

We also compare the actual global NQoS of the composite service as the simulated number of tasks in the workflow or the number of simulated CSs for each task varies. We set...
the matrix density to 1% in network latency and bandwidth prediction in this experiment.

As depicted from Figures 4 and 5, our NQoS re-prediction approach has more effect in increasing the actual global NQoS of the composite service than other approaches. With the increasing number of tasks in the workflow, the benefit of Geography approach is obvious. In addition, Geography approach requires less times of re-selections during the execution of the composite service.

VII. CONCLUSIONS AND FUTURE WORK

We have addressed the problem of network-aware service composition in SOC. To deal with the issues of prediction accuracy and on-demand measures in applying network delay predictions to the composition, we have firstly presented a network-aware QoS (NQoS) model for the composite service. Based on that, we have proposed a novel geographic location-based NQoS prediction approach before composition and a NQoS re-prediction approach during the execution of composite services. Comprehensive experiment results revealed our approach facilitates to improve the prediction accuracy and predictability of the NQoS values, and increase global NQoS of the composite service while ensuring its reliability constraints. Currently, we assume that all the CSs do not deploy on private networks and no bandwidth limitations are imposed between node pairs. In future, we will investigate other QoS attributes to improve our NQoS model and focus on predicting network delays at different times in one day for service composition.

VIII. ACKNOWLEDGMENTS

This research was sponsored in part by NSFC Programs (No.61103032, 61103105 and 91120302) and the National 973 Fundamental Research and Development Program of China under the Grant 2009CB320701.

REFERENCES


