Modeling and Algorithms for QoS-aware Service Composition in Virtualization-Based Cloud Computing

Jun HUANG, Yanbing LIU, Members, Ruozhou YU, Qiang DUAN, Nonmembers, and Yoshiaki TANAKA, Fellow

SUMMARY Cloud computing is an emerging computing paradigm that may have a significant impact on various aspects of the development of information infrastructure. In a Cloud environment, different types of network resources need to be virtualized as a series of service components by network virtualization, and these service components should be further composed into Cloud services provided to end users. Therefore Quality of Service (QoS) aware service composition plays a crucial role in Cloud service provisioning. This paper addresses the problem on how to compose a sequence of service components for QoS guaranteed service provisioning in a virtualization-based Cloud computing environment. The contributions of this paper include a system model for Cloud service provisioning and two approximation algorithms for QoS-aware service composition. Specifically, a system model is first developed to characterize service provisioning behavior in virtualization-based Cloud computing, then a novel approximation algorithm and a variant of a well-known QoS routing procedure are presented to resolve QoS-aware service composition. Theoretical analysis shows that these two algorithms have the same level of time complexity. Comparison study conducted based on simulation experiments indicates that the proposed novel algorithm achieves better performance in terms of efficiency and scalability without compromising quality of service. The modeling technique and algorithms developed in this paper are general and effective; thus they are applicable to practical Cloud computing systems.

key words: cloud service provisioning, network virtualization, quality of service, service composition, approximation algorithm

1. Introduction

The prevailing of Internet brings severe challenges to traditional computing paradigms. The main task of computing science is no longer to enhance the ability of individual devices to deal with separated information on limited storage medium, but to develop mechanisms for more devices to collaborate on handling much larger scales of information from worldwide through the Internet [1]. To face the challenge of massive data and relevant computing scales, Cloud Computing is propounded as a new computing platform for service provisioning. Cloud computing can be defined as a large scale distributed computing paradigm that is driven by economics of scale, in which a pool of abstracted, virtualized, dynamically-scalable computing functions and services are delivered on demand to external customers [2]. As an X-as-a-Service (XaaS) based computing model, the Cloud encapsulates all types of computing relevant resources as services. Users exploit these services allowing enterprises to save the expenses for purchasing computing/storage devices and hiring professionals to manage a computing center.

In a Cloud computing environment, different types of network resources need to be virtualized as services that can be provided to end users. Network virtualization plays an important role to offer such a simple way. It separates network service provisioning functions from data transportation mechanisms; thus de-coupling network services from underlying network infrastructures [3]. In addition, network virtualization can significantly enhance the flexibility, diversity, and manageability of network services so as to improve networking performance for meeting the requirements of Cloud computing. Thus, network virtualization enables the notion of virtualization, which forms the technical foundation of Cloud computing.

While virtualization technology establishes the basis of Cloud computing, the main challenges in Cloud computing lie in the modeling and efficient algorithms for Cloud service provisioning based on network virtualization. This is because users in Cloud computing always need composite Cloud services instead of a specific one [4]. On the other hand, the network performance has a significant impact on Cloud service provisioning, and in many cases networks become a bottleneck that limits Clouds from supporting high-performance applications [5], [6], for example, an user who wants to watch some movie clips from the video Cloud (e.g. Youtube), the network performance would be quite sensitive to the user’s experience. Therefore, the modeling and efficient algorithm design for service composition are the two significant issues in Cloud computing, which are the main focuses of this paper.

To provide customized Cloud service to end users, the Cloud computing resources (e.g. processors, storage etc.) as well as network infrastructures must be first abstracted as a set of services through virtualization technologies, then the service components from these virtualized services need to
be further composed as a Cloud service. However, composing appropriate service components into a Cloud service that meets multiple QoS requirements is a challenging optimization issue since such a problem can be basically addressed as Multi-Constraints Path problem (MCP), which is known to be NP-hard [8]. In addition to this, little research has addressed the QoS-aware service composition in Cloud computing. In this paper, we investigate the problem of QoS-aware service composition based on network virtualization in Cloud computing. The contributions of this paper are threefold.

- We present a system model to characterize Cloud service provisioning behavior with network virtualization technology. Through this model, we address the problem of QoS-aware service composition in Cloud computing as a variant form of MCOP. To the best of our knowledge, we are the first to model QoS-aware service composition on the basis of network virtualization in Cloud computing.

- We devise two approximation algorithms based on the presented model to resolve the QoS-aware service composition problem. We also analyze the theoretical properties of the proposed algorithm. Both algorithms are shown to be extensible and resilient in composing Cloud services.

- We conduct thorough comparisons between two algorithms through numerical experiments. Our results show that the proposed modeling technique and presented algorithms are general and efficient; thus are applicable to practical Cloud computing systems.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work of Cloud service provisioning. Section 3 models the service provisioning behavior in virtualization-based Cloud computing and formulates the QoS-aware service composition problem on the basis of the presented model. Section 4 presents two approximation algorithms for such a problem and compares the theoretical performance of the algorithms. In Sect. 5, numerical simulations are conducted for evaluating and comparing these two algorithms. Section 6 concludes this paper.

2. Related Work

Recently service provisioning has attracted much research interest. Kirschnik and co-authors investigated the problem of automated Cloud service provisioning in [9], and presented a practical Cloud service provisioning architecture. Tursunova studied QoS-aware service provisioning based on existing protocols for both immobile and mobile service provisioning in [10]. Yet researches above are failed to offer abstract and general model for Cloud service provisioning. Other works like [11], [12] and [13] provided general approaches for Cloud service provisioning from different perspective, but they did not detail the service composition in Cloud computing. Also, the service composition is deeply investigated by many works [14], [15], but none of them is developed for Cloud environment in particular.

The aforementioned works fail in considering the network performance in Cloud service provisioning. Differing from above works, we present a model based on the virtualization-based Network-Cloud model developed by our previous work [4] to better serve the need of Cloud computing. The detailed description about the presented model is given in Sect. 3.

On the other hand, research efforts on service composition have been made in the region of web service [16]-[19]. It seems that these studies tend to inherit the algorithms from QoS routing research area to compose appropriate web service. Since QoS routing and service composition can be formulated as a same problem in theory, we consider the Cloud service composition from QoS routing perspective by following the way of web service composition. Up to now, much progress have been made toward designing efficient algorithms for QoS routing, i.e. Multi-Constrained Path (MCP). Ergun et al. [20] gave an approximation algorithm via adopting an approximation test procedure to generate a set of upper bounds and lower bounds for squeezing the optimal. This algorithm is known to be one of the fastest algorithms for Delay Constrained Least Cost (DCLC). Xue et al. [21], [22] defined a different version of MCP and presented a set of approximation algorithms for each problem, which are known as the best among the aforementioned results. Huang and coauthors in [23] gave a heuristic for Multi-Constrained Optimal Path (MCOP) selection, in which a nonlinear combination technique and a geographical version of a well-known process in solving DCLC were introduced. The results obtained in [23] show that the proposed heuristic can achieve a good trade-off between execution time and quality of the path, and it needs less time to compute a path compared with Xue’s algorithm in [21]. More recently, Huang and Tanaka [24] adopted a graph-extending technique and proposed a Fully Polynomial Time Approximation Scheme (FPTAS), which has been proved to be currently one of the fastest algorithms for MCOP. Other progress toward efficient QoS routing algorithm can be found from the surveys given in [26], [27], and [28].

The above researches mainly address the problem from QoS routing perspective, which cannot be applied directly in virtualization-based Cloud computing for QoS-aware service composition, we present two approximation algorithms in resolving the problem of QoS-aware service composition based on our previous work [23], [24], and [25]. The detailed description about the presented algorithms can be found in Sect. 4.

3. Modeling for Cloud Service Provisioning

3.1 System Overview

In virtualization-based Cloud computing, Cloud service provisioning consists of two key steps. First, the Cloud and network resources need to be abstracted as tandem services
that form a virtual network. In this step, network virtualization plays a crucial role in resource abstraction. Second, the service provider composes service components from these synthesized services as Cloud services and provides such composed Cloud services to end users.

Upon our previous work [4], a Cloud service can be described as composite Cloud-Network service. The Cloud resources such as processors, storage, etc. are abstracted as "pure" Cloud service, while the network infrastructures are also virtualized as network service through network virtualization. That is to say, Cloud service provisioning should be supported by network virtualization [7]. A network virtualization paradigm in Cloud computing can be represented as a model shown in Fig. 1. Such model includes two planes, namely the Infrastructure Provider (InP) plane and the Service Provider (SP) plane from bottom to top. Within the infrastructure provider plane, a collection of network resources are connected by physical links. The upper plane aims at forming a virtual network topology which consists of services abstracted from resources in the bottom plane. To enable SP to effectively utilize underlying infrastructure resources, interactions between InP and SP is becoming quite important. The InP is responsible for providing SP with information about its network infrastructure so that the networking resources in the infrastructure can be accessed by SP. On the other hand, due to the de-coupling of service provisioning and infrastructure management roles in virtualization-based Cloud computing, an InP may not want to expose every detail of its resources. An resource-as-a-service paradigm in Cloud computing allows an InP to provide information to SP with a certain level of abstraction through publishing an resource service description.

By abstracting the underlying resources as a set of services, the service components from these services can be synthesized to form Cloud services. From an user's perspective, when a request is posted to the Cloud, appropriate composed service should be delivered as fast as possible while meeting specific QoS requirements. Within the Cloud, the provisioning system receives a request, provides the composed services and delivers it back to users. Figure 2 shows a Cloud service provisioning system. In this system, when a user's service request arrives at the service entrance portal, the service composition process is invoked: a series of service components, which come from services abstracted by network virtualization, are combined appropriately to form a composed Cloud service (path). For example in Fig. 2, a path 0 → (1-1) → (2-3) → (3-1) → 4 in the service components formed virtual network represents a composed Cloud service. Through service composition, the Cloud service provisioning can be eventually realized for end users. The crucial concern in this process is the technique for service composition, which both influences the service experience of users and the system load of the Cloud. Therefore, the main focus in Cloud service provisioning is how to compose the optimal sequence of service components from virtualized services into the Cloud service and provide it to end users.

With above Cloud service provisioning system, QoS-aware service composition can be modeled as follows.

A virtual service network formed by interconnecting service components can be modeled as a directed graph \(G(V, E)\) with \(n\) vertices and \(m\) edges. Each vertex \(v\) is associated with a capacity weight \(c\) denoting its serving capability in service \(s\), and each edge \(e \in E\) is associated with \(K\) weights \(w = (w_1, w_2, \ldots, w_K)\) representing QoS parameters. Let \(\text{Req} = (C, W)\) be a user request, where \(C = (C_1, C_2, \ldots, C_K)\) denotes the requirements (constraints) for computing relevant capacities of \(H\) various services, which should be met by nodes. Denote \(W = (W_1, W_2, \ldots, W_K)\) as the \(K\) QoS constraints on edges, and a path \(p\) in the virtual service network as a series of tandem services. Then, the problem of QoS-aware service composition is formulated in the following.

### 3.2 Problem Formulation

**Definition 1:** Feasible Service Composition. A composed service, i.e., a path \(p\) in the service network, is said to be feasible if for \(\forall \nu \in p\) there is \(c_\nu \geq C_\nu\) where \(\nu \in S_h, 1 \leq h \leq H\), and \(w_k(p) \leq W_k\) for all \(1 \leq k \leq K\).

Denote \(\{p'\}\) as all feasible service compositions in \(G(V, E)\), to each \(p' \in \{p'\}\), there exists a smallest value \(\eta_i \in (0, 1)\) such that \(w_k(p'_i) \leq \eta_i \cdot W_k, 1 \leq k \leq K\), and for \(\forall \nu \in p'_i\) such that \(c_\nu \geq C_\nu, 1 \leq h \leq H\) respectively.
Problem 1: QoS-aware Service Composition (QSC). QSC is to find an optimal composition of services $p^\text{opt}$ among feasible service compositions in $G(V, E)$ and the corresponding smallest value $\eta^\text{opt}$ among all $\eta_i$ such that $w_k(p^\text{opt}) \leq \eta^\text{opt} \cdot W_k$, $k \in [1, K]$ where $K \geq 2$.

Theorem 1: QSC problem is NP-hard.

Proof 1: Since the constraint on node $C = (C_1, C_2, \ldots, C_H)$ can be satisfied beforehand by pruning the topology in advance, QSC maps directly to the special case of Multi-Constrained Optimal Path (MCOP) [29], thus QSC is NP-hard.

Definition 2: $(1 + \epsilon)$-Approximation Algorithm. An algorithm is a $(1 + \epsilon)$-approximation algorithm (or simply, an approximation algorithm) for QSC if the algorithm generates a path $p$ such that $w_k(p) \leq (1 + \epsilon) \cdot \eta \cdot W_k$, $1 \leq k \leq K$. The running time of the algorithm is polynomially bounded by both the input size and $1/\epsilon$.

4. An Approximation Algorithm for QSC

4.1 Main Building-block of Approximation Algorithm

In order to solve QSC, this paper proposes an approximation algorithm depending on our prior work [24]. The main building-block of this proposed algorithm is given in Algorithm 1, Main Build-block Approximation Algorithm (MBAA). Essentially MBAA solves the special case of QSC where $K = 2$, and it includes four major steps. The first step (line 1 in Algorithm 1) prunes the topology according to the constraints on the vertex. This is a required step because $c$ denotes the serving capability of a node in a virtual service network and $c < C_s$ means such a node cannot meet the service requirement; therefore should be eliminated in the initial stage. In addition to this, the notation $G(V, E)$ is still used here to denote the pruned topology.

The second step of MBAA (lines 2 and 3 in Algorithm 1) transforms real-valued weights and constraints into integers. The integrated weights are denoted as $w'_k(e)$ and constraints as $W'_k$, where $1 \leq k \leq 2$. The parameter $\psi$ is designated to $(\frac{n-1}{\epsilon})$ to guarantee that the algorithm can find a $(1 + \epsilon)$-approximate for QSC (readers are referred to [24] for more detailed proof). The computing procedure of $w'_k = \lceil \frac{w_k}{W_k} \times \frac{n-1}{\epsilon} \rceil$ and $W'_k = \lceil \frac{n-1}{\epsilon} \rceil$ normalizes all the weights to an order of magnitude in $\lceil \frac{n-1}{\epsilon} \rceil$ for simplifying calculation.

The third step (line 4) of MBAA shows the procedure of graph extending from $G(V, E)$ to $G'(V', E')$, where $V'$ denotes the extended set of vertices and $E'$ denotes the extended set of edges. The extension is performed both on vertices and edges, and it works in two ways. First, original vertices are copied $\lceil \frac{n-1}{\epsilon} \rceil$ times, i.e., each of vertex is extended to be a vertex group. Meanwhile, original edges are also copied several times and each is associated with its first original weight $w'_k(e)$. By doing this, the second weight in the original graph is melted into the extended graph. Next, each vertex in the extended vertex group is lexicographically interconnected with the next one by a zero weight edge. Following this way, the original graph $G(V, E)$ is converted to a Directed Acyclic Graph (DAG) $G'(V', E')$. More detailed graph-extending illustrations can be referred to our work [24].

The fourth step (line 5) is to find optimal weighted path in $G'(V', E')$. As known, such a process could be terminated with time complexity of $O^{(\frac{n}{\epsilon})}$.

With the results calculated in line 5, the weights of the optimal path can also be obtained. Step 6 tests the results with the weight constraints computed in line 3 to check the feasibility of the optimal path.

The following two lemmas give the analytical properties for MBAA, and the detailed proof can be found in [24].

Lemma 1: MBAA finds a solution within $O^{(\frac{n}{\epsilon})}$ time.

Lemma 2: MBAA finds a $(1 + \epsilon)$-approximation.

4.2 The Proposed Approximation Algorithm

On the basis of MBAA, the proposed approximation algorithm (PAA) is presented in Algorithm 2. The underlying principle of PAA is to first leverage nonlinear combination for weights aggregation and then enable MBAA to solve the transformed problem.

A nonlinear combination function for weights aggregation [23],[25] is

\[ w'(e) = \left( \sum_{k=1}^{K} w_k(e)^{W_k} \right)^{\frac{1}{q}} \]

where $w'(e)$ denotes the aggregated weight on edge $e$, $q$ is an integer constant.

It is interesting to notice that if $q = 1$, Eq. (1) reduces to the form of linear combination; while $q \to \infty$, Eq. (1) can
Algorithm 2 PAA

Input:  
Graph: \( G(\mathcal{V}, E, w, W, c, C) \)

Output:  
Path: \( p^\mathcal{M} \)

1: for \( \forall v \in [1, K] \) do
2: Construct a new graph \( G'(V, E, w', W', c, C) \) where \( w' = \left\lfloor \frac{w(v) \cdot (c(v))}{\sum_{i \in \mathcal{V}} \left( \frac{w_i(v)}{W_i} \right)} \right\rfloor \), \( W' = \{ W_i \cdot (K - 1)^\frac{1}{h} \} \);
3: Call MBAA using \( G'(V, E, w', W', c, C) \) as its input;
4: If the obtained path is better than previous best one \( p^\mathcal{M} \), replace \( p^\mathcal{M} \);
5: end for
6: return \( p^\mathcal{M} \);

be rewritten as

\[
w'(e) = \max_{1 \leq k \leq K} \left\lfloor \frac{w_k(e)}{W_k} \right\rfloor
\]  

(2)

Most of existing works in the field of web service composition, such as [16]–[19], incline to use Eq. (1) or Eq. (2) for establishing the overall quality of composed service. This is due to the fact that Eq. (1) or Eq. (2) can not only reflect the users’ preference on service components, but also is significant to the users’ experience from the Cloud. Essentially Eq. (1) provides an effective technique to convert the original graph to a simple one, which could be resolved by applying MBAA. On the other hand, since adopting nonlinear combination would introduce some errors to the final results, the path searching process in PAA (line 2 to line 4 in Algorithm 2) is allowed to iterate several times for progressively improving the quality of the finding path. In this sense, the final path is picked from the best one among all iterations of PAA.

By leveraging the concept of nonlinear combination, the time complexity of PAA can be easily determined as \( O(K \cdot \frac{\mathcal{M}}{\epsilon}) \). This is because the time complexity of PAA is dominated by that of MBAA and iteration times. However, the finding path can no longer achieve \((1 + \epsilon)\)-approximation for QSC. The following theorem gives a detailed explanation.

**Theorem 2:** PAA generates a \((K - 1) \cdot (1 + \epsilon)\)-approximation for QSC.

**Proof 2:** The Introduction of Eq. (2) leads to the \((K - 1)\)-approximation to the final result [21], and according to **Lemma 2**, it can be easily obtained that PAA generates a \((K - 1) \cdot (1 + \epsilon)\)-approximation for QSC.

4.3 A Variant of A Well-known Approximation Algorithm

Following the same way of PAA design, a well-known approximation algorithm ADAPT [20], which is proposed to solve delay-constrained least cost problem, can also be applied in solving QSC. Differing from path searching process of PAA, ADAPT adopts a technique of scaling-and-rounding to simplify the problem, and uses an approximation test procedure to generate a set of upper bounds and lower bounds for approximating the optimal. In order to enable ADAPT to solve QSC, some essential modifications should be made on that algorithm. Algorithm 3 shows the presented variant of ADAPT, named as VAA.

![Algorithm 3 VAA](image)

It is worthwhile to note that procedure ADAPT in the VAA is reconfigured to handle the directed graph, and the time complexity of VAA is \( O(K \cdot \frac{\mathcal{M}}{\epsilon}) \). This is because ADAPT is developed for the undirected graph, and it could be terminated within \( O(\frac{\mathcal{M}}{\epsilon}) \) time [20]. Therefore, the time complexity of VAA is \( O(K \cdot \frac{\mathcal{M}}{\epsilon}) \) after ADAPT iterates \( K \) times. For the quality of solution, Algorithm 3 finds a \((K - 1) \cdot (1 + \epsilon)\)-approximation path for QSC. The proof of this proposition is quite similar to that of Theorem 2. In other words, VAA has the same theoretical performance with PAA. In the following section, the performance of both PAA and VAA will be examined experimentally.

5. Numerical Simulation and Results Analysis

Though both PAA and VAA algorithms have the same time complexity of \( O(\frac{\mathcal{M}}{\epsilon}) \), they actually perform very differently in running time. In this section, we evaluate the performance of both algorithms with simulation experiments. First the algorithm performance on a specific virtual network topology is evaluated. Then both algorithms are tested on a set of randomly generated service network topologies, which is a more realistic case in Cloud computing environments.

5.1 Performance Metric

The metrics that are used in the performance evaluation are defined as follows.

**Definition 3:** Main Procedure Running Time (MPT). MPT of an algorithm denotes the running time of the main procedure of the algorithm. The main procedure of an algorithm is the process of the algorithm excluding operations for conduction, initialization, and destruction. This metric is for evaluating time cost performance of an algorithm, which reflects the time that a user has to wait before receiving the response for a service request.
**Definition 4:** Returned Path Weights ($RPW_k$). $RPW_k$ of an algorithm, where $k \in [1, K]$, denotes the $k$-th QoS attribute of the service composition path returned by the algorithm, that is

$$RPW_k = \sum_{e \in \rho} w_k(e).$$

This metric gives the QoS attributes of the service composition path returned by an algorithm, which reflects the QoS that the service composition may offer to its users. We assume that the smaller of the returned path weights is, the better performance of such path.

The performance metrics defined above are used in the rest of this section for evaluating PAA and VAA algorithms.

### 5.2 Experiments over Specific Virtual Network (DsVN)

A specific virtual network is given here to test the performance of the algorithms explicitly. The topology of DsVN is shown in Fig. 3, in which service 1 and service 4 have three service components while service 2 and service 3 each has four components. The components are interconnected with virtual links, each has a pair of weights. The detailed configuration of such topology can be found in [30], and this topology is constructed totally based on the model described above.

PAA and VAA are tested independently over the DsVN topology. To evaluate the performance of the proposed PAA, we prefer to test parameters $\psi$ instead of $\epsilon$ for simplicity. The value of $W_1$ is set to a much larger value of $W_1 = 1000$. In this way the variables in the experiments are reduced to only $W_2$ and $\psi$. The fact to be shown here is that with different values of $\psi$ relevant to the constraints $W_k$, the algorithm of PAA gains different paths.

The experimental results are shown in Table 1 and Table 2. As can be seen in the tables, when a strict value of $W_2$ is given, VAA algorithm can find the exact path that meets the constraint of $W_2$ while seeks an optimal value of $w_k(p^{opt})$, whereas PAA gains the path with different $\psi$ that has more deviation from the constrained optimal path. On the other hand, when the value of $\psi$ is exactly a multiple of $W_2$, PAA can still find the exact optimal path, as shown where $\psi = 10$ and $W_2 = 10, \psi = 11$ and $W_2 = 11, \psi = 12$ and $W_2 = 12$ and so on.

The results reported in Tables 1 and 2 show that PAA algorithm is less accurate than VAA for finding the optimal path in the small scale DsVN network. However we found that PAA gains superiority to VAA in large scale networks, which is shown in the following.

### 5.3 Experiments over Random Virtual Networks

Experiments are conducted using random resource networks to further test the scalability and robustness of the proposed algorithm. The virtual network topologies used in these experiments were generated randomly. The network scale varies from 100 nodes to 1000 nodes which are uniformly distributed in $H$ tandem services. Here $H$ was set to be 10 since the amount of service class is not large. Figure 3 shows an example of the random generated network, in which there are 14 nodes and the number of virtualized service $H$ is equal to 4. Each link in the random service networks has three QoS parameters. Each reported result was obtained by repeating the experiment for twenty times independently under the same configuration of the experiment environment.

The comparisons comprise of two sets of experiments. In the first set of experiments, two factors including variation of $q$ in Eq. (1) and $\epsilon$ that have impact on performance of PAA were tested. Three values, namely, 1, 2 and 3 were chosen for variable $q$, known as the linear combination, square combination and cubic combination, respectively. $\epsilon$ were set to be 0.1 and 0.5, and the constraints of the three weights were set as different magnitude of values $W_1 = 300.0, W_2 = 0.0015$ and $W_3 = 0.2$ to test the robustness of the algorithms.
The first set of experiment results are presented in Tables 3, 4, 5 and 6. Table 3 shows the RPW returned by the PAA. As it shown, the average running time of PAA using three combination methods is almost the same. This indicates that different combination method has little impact on the efficiency of PAA. While for the different value of \( \epsilon \), the performance of PAA varies significantly from one to another. With the \( \epsilon \) reduced from 0.5 to 0.1, the MPT's increase several times. This is because when \( \epsilon \) reduces, the graph extending step of PAA constructs a much larger graph, thus finding shortest paths in a larger graph needs more time.

Tables 4, 5 and 6 shows the RPWs of PAA using linear, square and cubic combination respectively. A similar implication can be seen from these tables that three combination methods also offer similar effectiveness, but different value of \( \epsilon \) shows little influence here, that is, the paths with the same number of nodes mostly share common weights value.

In the second set of experiments, the performance comparisons between PAA and VAA are conducted by configuring Eq. (2) for PAA and VAA instead of linear, square and cubic combination. A new value of \( \epsilon = 0.2 \) is added to examine its impact on the performance of both algorithms in more detail. The comparison results are shown in Tables 7, 8, 9 and 10.

The MPT results reported in Table 7 reflect the efficiency of two algorithms. As can be seen from this table, the proposed PAA algorithm runs much faster than VAA under the same testing environment (in the same network and with a fixed \( \epsilon \) value). As the number of nodes increases, the difference of MPT's increases significantly as well, and when the scale grows to as much as 1000 nodes, PAA can even run tens of times faster than VAA. This implies that PAA is more efficient than VAA for finding a service composition path, especially in large scale networks, which also indicates better scalability performance.

The RPW values given in Tables 8, 9 and 10 measure...
The experiment results obtained in randomly generated virtual networks show that PAA can achieve the same level of QoS performance for service composition as VAA but with shorter processing time. Also, increment of PAA time cost with network scale is much slower than that of VAA, which implies that PAA has better scalability than VAA. Therefore, PAA surpasses VAA in time efficiency and scalability. PAA can meet both QoS and real-time efficiency requirements for service composition in Cloud computing environments.

6. Conclusion

Cloud computing is proposed as a new computing paradigm for service provisioning, and network virtualization plays a crucial role in Cloud computing. In a Cloud environment, various resources need to be virtualized as services by network virtualization, and the service components from these services need to be then composed into Cloud services that are provided to end users. Therefore, composing services that satisfy Quality of Service (QoS) requirements becomes a significant technical issue in Cloud service provisioning. This paper investigated QoS-aware service composition problem in virtualization-based Cloud computing. The contributions made in this paper include a system model for Cloud service provisioning and two approximation algorithms developed based upon this model for resolving QoS-aware service composition. Particularly, the first approximation algorithm is a novel proposed one, while the second algorithm is a variant of a well-known procedure for QoS routing. Comparison between these two algorithms indicated that though both algorithms have the same theoretical properties, the proposed novel algorithm shows better experimental performance in both running time and quality of solution. The modeling technique and algorithms presented in this paper are general and agnostic to Cloud implementations; thus, they are applicable to practical Cloud computing systems.

As for future work, we plan to enhance the our algorithms' capability of handling invalid request, and implement algorithms into the Cloud simulation environment or deploy them in the real-world Cloud computing systems.

Acknowledgement

We would like to thank the anonymous reviewers for their insightful comments on the early submission of this paper.

This work was partially supported by program for NCET and the following grants: National Science Foundation of China (Grant No. 60973160), National 973 Foundation (Grant No. 2012CB315803, 2010CB334710), Natural Science Foundation of Chongqing (Grant No. CSTC.2012JB40008), R&D Foundation of Chongqing (Grant No. KJZH(2020) and Open Project of Key Lab of Information Network Security of Administration of Public Security (Grant No. C11609).
References


http://code.google.com/p/dsvntop/downloads/list
Yanbing Liu received M.S. degree in computer science and technology from Beijing University of Posts and Telecommunications, China, in 2001, and Ph.D. degree from University of Electronic Science and Technology of China, in 2007. He is currently a professor and vice-president of Chongqing University of Posts and Telecommunications (CQUPT). He is the recipient of National Science and Technology Award and several Chongqing Science and Technology Awards. He is an executive director of Chongqing Youth Federation of Science and Technology, a panelist of Chongqing Science and Technology Award, a member of the Chongqing General Bid Evaluation Committees, and an academic leader of CQUPT. He has published more than 60 refereed papers. His research interests include Cloud computing, network virtualization, Internet of things, and system performance evaluation.

Ruzhou Yu is an undergraduate at school of computer science, Beijing University of Posts and Telecommunications. His research interests are QoS routing, Cloud computing, and network virtualization.

Qiang Duan received the B.S. in Electrical and Computer Engineering and M.S. in Telecommunications and Electronic Systems in 1992 and 1995 respectively. He received his Ph.D. in Electrical Engineering from the University of Mississippi in 2003. Dr. Duan is an Assistant Professor of Information Science and Technology at the Pennsylvania State University Abington College. His research interests include data communications and computer networks, network virtualization and the next generation Internet, the service-oriented architecture, Cloud computing, and networking system hardware/software design and analysis. He has published over 50 journal articles, conference papers, and book chapters. Dr. Duan is an associate editor for International Journal of Network Protocols and Algorithms and International Journal of Internet and Distributed Computing Systems.

Yoshiaki Tanaka received the B.E., M.E., and D.E. degrees in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1974, 1976, and 1979, respectively. He became a staff at Department of Electrical Engineering, the University of Tokyo, in 1979, and has been engaged in teaching and researching in the fields of telecommunication networks, switching systems, and network security. He was a guest professor at Department of Communication Systems, Lund Institute of Technology, Sweden, from 1986 to 1987. He was also a visiting researcher at Institute for Posts and Telecommunications Policy, from 1988 to 1991, and at Institute for Monetary and Economic Studies, Bank of Japan, from 1994 to 1996. He is presently a professor at Global Information and Telecommunication Institute, Waseda University, and a visiting professor at National Institute of Informatics. He received the IEEE Outstanding Student Award in 1977, the Niwa Memorial Prize in 1980, the IEICE Achievements Award in 1980, the Okawa Publication Prize in 1994, the TAF Telecom System Technology Award in 1995 and in 2006, the IEICE Information Network Research Award in 1996, in 2001, in 2004, and in 2006, the IEICE Communications Society Activity Testimonial in 1997 and in 1998, the IEICE Switching System Research Award in 2001, the IEICE Best Paper Award in 2005, the IEICE Network System Research Award in 2006 and in 2008, the IEICE Communications Society Activity Award in 2008, the Commendation by Minister for Internal Affairs and Communications in 2009, and the APNOMS Best Paper Award in 2009.