Towards Green Service Composition Approach in the Cloud

Shangguang Wang, *Senior Member, IEEE*, Ao Zhou, Ruo Bao, Wu Chou, *Fellow, IEEE*, and Stephen S. Yau, *Life Fellow, IEEE*

Abstract—With the increasing popularity of cloud computing, many notable quality of service (QoS)-aware service composition approaches have been incorporated in service-oriented cloud computing systems. However, these approaches are implemented without considering the energy and network resource consumption of the composite services. The increases in energy and network resource consumption resulting from these compositions can incur a high cost in data centers. In this paper, the trade-off among QoS performance, energy consumption, and network resource consumption in a service composition process is first analyzed. Then, a green service composition approach is proposed. It gives priority to those composite services that are hosted on the same virtual machine, physical server, or edge switch with end-to-end QoS guarantee. It fulfills the green service composition optimization by minimizing the energy and network resource consumption on physical servers and switches in cloud data centers. Experimental results indicate that, with comparisons to other approaches, our approach saves 20-50 percent of energy consumption and 10-50 percent of network resource consumption.

Index Terms—Service composition, cloud computing, energy consumption, network resource consumption

1 Introduction

Service composition is the core technology of service-oriented cloud systems [1]. It provides a solid foundation for service reuse and integration. In the service composition process, a *composite service* is combined with a series of abstract tasks and each task invokes a concrete service to satisfy users' requirements. In the process of service composition, in addition to considering the services' functions to match users' requirements, non-functional attributes, e.g., quality of service (QoS) also should be considered. In many cases, QoS such as response time, throughput, and reliability, is critical to the usability and success of the service applications. As such, most existing service composition schemes [2], [3], [4] in the cloud or other service systems have a focus on the QoS aspect of the composite services [1].

However, with an increasing number of services being deployed on virtual machines, the energy consumption has become one of the major causes of the increased cost in cloud data centers. Moreover, research has shown that network devices consume 20–30 percent of the overall energy consumed in the data centers [5]. Unfortunately, the network devices waste significant amounts of energy in the cloud data centers today [6]. Therefore, in addition to performance related to QoS requirements [3], [4], the energy efficiency of the services has become a critical consideration in the cloud [7], [8], [9].

Although energy efficiency has become an active focus of research [8], [10], [11], the energy consumption of composite services is yet to be considered. Those services are typically distributed across various physical servers with varying energy constraints depending on the types of switches utilized, the types of computing servers, etc. Google released information that the average power consumption for a Google search is about 0.0003 KWh[1]. With service composition, the energy consumption can be even much worse for large-scale computing infrastructures, especially for cloud systems.

We consider the problem in a fat-tree data center [13], [14], [15]. As shown in Fig. 1, tasking the specific service on the specific physical server, composing these services onto a subset of links, and switching off unnecessary network elements would be the most green (i.e., energy-efficient and network-resource-efficient) [16]. For example, services $s_{1,3}$ and $s_{2,4}$($s_{i,j}$ denotes the jth candidate service for the ith service classes) as one composite service would consume more network resource than a composite service comprising $s_{2,4}$ and $s_{1,4}$, because the latter's data packet transmission is in a subnet. Unfortunately, in a fat-tree cloud data center (FCDC), services are randomly distributed in different physical servers. Therefore, cloud companies urgently need an effective way to promote green computing by making use of distributed services deployed on different servers in the same subnet or the same pod.

In this paper, we first examine the trade-off among the QoS performance, energy consumption, and network resource consumption of composite services in a cloud fat-tree based

- *S. Wang, A. Zhou, and R. Bao are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: {sgwang, aozhou, br}@bupt.edu.cn.*
- *C. Wu is with the Huawei Technologies, Shenzhen, Guangdong 518129, China. E-mail: Wu.Chou@huawei.com.*
- *S. Yau is with the School of Computer Science and Engineering, Arizona State University (ASU), Tempe, AZ 85287 USA. E-mail: yau@asu.edu.*

Manuscript received 14 Sept. 2017; revised 25 June 2018; accepted 28 Aug. 2018. Date of publication 3 Sept. 2018; date of current version 5 Aug. 2021. (Corresponding author: Shangguang Wang.)
Digital Object Identifier no. 10.1109/TSC.2018.2868356

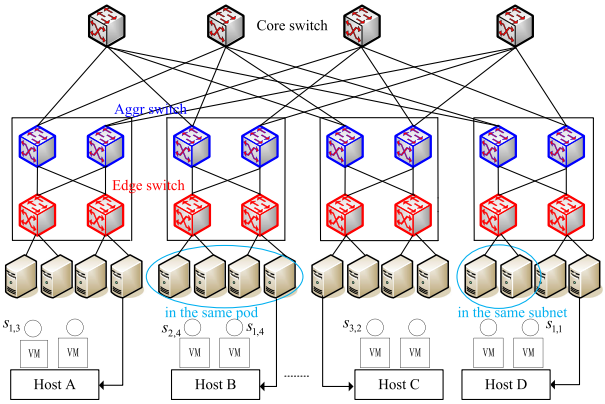


Fig. 1. Fat-tree topology structure in a cloud data center. The switches at the top (black), middle (blue), and bottom (red) layers are core, aggregation, and edge switches, respectively [12]. S represents services running on one virtual machine (VM) from one physical server such as Host A. All host servers that connect to the same edge switch are referred to as "in the same subnet". All host servers that connect to the same aggregation switch are referred to as "in the same pod".

data center. Then, we propose a green service composition approach that gives priority to those services hosted on the same virtual machine, physical server, or edge switch with end-to-end QoS guarantee. Finally, our approach fulfills the green service composition optimization by maximizing the QoS performance and minimizing the energy and network resource consumption from physical servers and switches in FCDCs. Our contributions in this paper are as follows:

1) In contrast to existing service composition approaches, we focus not only on efficient service composition but also on energy and network resource consumption as another essential consideration. To the best of our knowledge, this is the earliest effort to consider energy and network resource consumption of composite services in fat-tree based cloud data centers.
2) We propose a green service composition approach. It contains the design of an energy consumption model and network resource consumption model of composite services. It also formulates green service composition as a multi-objective optimization problem, and finds the optimal service composition solution.
3) To evaluate the performance of our approach, we implement our approach and compare the results to those from traditional approaches. The experimental results show that our approach achieves higher energy efficiency and network-resource-efficiency as well as end-to-end QoS guarantee than traditional approaches.

The remainder of this paper is organized as follows. Section 2 discusses the related work in this area. Section 3 describes our approach in detail, including the definition of the energy consumption, the network resource consumption model, the proposed service composition approach, and the determination of the optimal solution. Section 4 demonstrates the benefits of our approach via experimental evaluation. Finally, Section 5 concludes this paper and discusses future work.

2 Current State of the Art

A number of schemes have been proposed for service composition in the cloud. However, most of those schemes mainly focus on QoS performance as the continuation of the traditional QoS-aware web service composition approach [3], [17], [18], [19], [20], [21], [22]. In this section, we briefly review the relevant research efforts, explore the basic ideas behind these schemes, and highlight how our work extends the current research and makes advances in cloud service composition systems.

From the perspective of QoS-aware service composition in the cloud, Gutierrez-Garcia and Sim [23], [24] presented an agent-based approach to compose services in multi-cloud environments for different types of cloud services. In their approach, each cloud participant is represented and instantiated by an agent. The agent can autonomously and successfully deal with changing service requirements through self-organization and collaboration. The agent-based cooperative problem solving technique is used to dynamically select the most appropriate services to create a composite solution. Unfortunately, this approach [24] and other approaches [25], [26], [27], [28], including our previous approach [29], [30], still focus on traditional performance measures such as percentage of successful service compositions, average service composition time, and average number of messages exchanged.

Traditional service composition in the cloud is a problem in which there are many potential solutions, among which one or a limited number of solutions are optimal. Hence, many classical algorithms, such as 0-1 linear programming [31], particle swarm optimization [29], [32], heuristic algorithm [33], genetic algorithm [26], can be used to solve optimization problems. These algorithms can guarantee that an optimal solution will be found solely by taking exponential time complexity [1]. Thus, using classical algorithms to solve these optimization problems is possible with some improvements. These algorithms can decrease the time complexity and reduce the amount of services for service composition that simplifies the search space.

In contrast to QoS-aware service composition, green service composition focuses on reducing the energy consumption. For example, Bartalos and Blak [16] proposed an instantaneous power estimation approach for web services. This approach aggregates a linear instantaneous power model created from readily available hardware performance counters. Jiwei and Chuang [34] proposed stochastic models to describe web service systems, and mathematically analyzed them to evaluate the energy efficiency. They used Markov Decision Process to solve both the service selection problem and dynamic speed scaling problem. However, the studies cited above [16], [34] and other similar studies [10], [35], [36] only optimize the service request scheduling to achieve optimal energy efficiency in web service systems. They cannot reduce the energy consumption in the cloud because no consideration is given to the location of physical servers and the network topology of the cloud data centers.

Other studies [7], [33], [37] have proposed energy-aware service composition solutions for the cloud, but the solutions proposed are still traditional service composition schemes. They only take the energy consumption as one attribute of services, and cannot find the essential relation between green service composition and cloud data centers. Unlike the studies outlined above, Li et al. [38] presented a cost and energy-aware scheduling algorithm for cloud schedulers to minimize the execution cost and reduce the energy consumption while

TABLE 1
Notations

Symbol	Meaning
s	A composite service in the cloud
m	Number of service classes
n	Number of candidate services in each service class
s_i	The ith service class, $i = 1, 2, \ldots m$
$s_{i,j}$	The jth candidate service of the ith service class, $j = 1, 2, \ldots n$
r	Number of QoS attributes
$q_k(s)$	Aggregated service QoS values by the kth attribute values of s in the cloud
$nq_k(s)$	Aggregated network QoS values by the kth attribute values of s.
$U(s)$	QoS utility function of s
c	Number of core switches in the cloud
a	Number of aggregation switches in the cloud
e	Number of edge switches in the cloud
p	Number of physical machines linked to edge switches in the cloud
s_{core}	Total size of packets transferred by c core switches from composite service s
s_{agg}	Total size of packets transferred by α aggregation switches from composite service s
s_{edge}	Total size of packets transferred by e edge switches from composite service s
$NRC(s)$	Total network resource consumption of composite service S in the cloud
P_i^{switch}	Energy consumption of the ith switch
P_i^{pm}	Energy consumption of the ith physical machine
$EC(s)$	Total energy consumption of composite service s

meeting deadline constraints. Their aim is to devise an optimal scheme to execute scientific applications within a specified time constraint for cloud schedulers. Scientific applications that are becoming increasingly data-communication, and computation-intensive. Hence, reducing their network resource consumption is more important than low time complexity for scientific applications in the cloud.

In contrast to existing schemes that are not providing the green solution for composite services in the cloud, our approach focuses on service location in FCDCs and derivation of a low energy and network resource consumption service composition solution. Finally, based on our proposed approach, green service composition can be performed while satisfying end-to-end QoS constraint in the cloud.

3 Preliminaries and Motivation

3.1 Service Composition

Several atom services from different service classes are composited to achieve a more complex service when the functional requirements of a user cannot be fulfilled by an individual service. Each service class often consists of multiple candidate services. Related notations are explained in detail in Table 1.

In the service composition process, functional and non-functional requirements have to be considered when choosing such candidate services. The non-functional requirements are specified by QoS attributes (such as latency, throughput, or reliability), and are especially important when many functionally equivalent services are available. Hence, QoS plays an important role in traditional service composition environments.

We only consider the sequential composition model since other models (e.g., parallel, conditional, and loop models) can be transformed into the sequential model using the techniques in [39], [40].

3.2 QoS Utility Function

In order to achieve the largest QoS value in the service composition process, we need to calculate the aggregated QoS of the selected candidate services. Then the aggregated QoS of a composite service can be obtained using the QoS utility function [41], [42]. Different from the traditional scheme, in this paper, we adopt the QoS utility function consisting of QoS from service and network [43].

Definition 1 (QoS Utility Function). *The QoS utility function of one sequential composite service s is defined as follows:*

$$U(s) = \sum_{k=1}^{r} \left(\frac{Q_k^{\max} - q_k(s)}{Q_k^{\max} - Q_k^{\min}} + \frac{NQ_k^{max} - nq_k(s)}{NQ_k^{max} - NQ_k^{min}} \right) . w_k \quad (1)$$

with

$$\begin{cases} Q_k^{max} = \sum_{i=1}^{m} Q_{i,k}^{max}, Q_{i,k}^{max} = max_{\forall s_{i,j} \in s_i}\, q_k(s_{ij}) \\ Q_k^{\min} = \sum_{i=1}^{m} Q_{i,k}^{\min}, Q_{i,k}^{\min} = \min_{\forall s_{i,j} \in s_i}\, q_k(s_{ij}) \\ NQ_k^{max} = \max nq_k(s) \\ NQ_k^{\min} = \min nq_k(s), \end{cases} \quad (2)$$

where $w_k \in R^+ (\sum_{k=1}^{r} w_k = 1)$ represents the weight of each QoS attribute; Q_k^{max} is the maximum value of the k*th QoS attribute of service in the composition service s; similarly, Q_k^{min} is the minimum value; NQ_k^{max} is the maximum value of the* k*th QoS attribute of network in the composite service s; and NQ_k^{min} is the minimum value. More detailed explanation can be found in [43].*

3.3 Motivation and Challenges

As an example, one composite service requirement $s = \{ s_1, s_2\ s_3 \}$ from a service provider in which three service classes (s_1, s_2,s_3) are invoked. Each service class contains five candidate services such as $s_1 = \{s_{1,1}, s_{1,2}, s_{1,3}, s_{1,4}, s_{1,5}\}$, $s_2 = \{s_{2,1}, s_{2,2}, s_{2,3}, s_{2,4}, s_{2,5}\}$, and $s_3 = \{s_{3,1}, s_{3,2}, s_{3,3}, s_{3,4}, s_{3,5}\}$.

Then, when a service composition requirement $s = \{ s_1, s_2\ s_3 \}$ is determined, traditional service composition schemes often select these services with high QoS values in FCDCs. However, the obtained solution may have good QoS aggregation values, but high energy and network resource consumption. For example, for the traditional schemes, $s = \{s_{1,3}, s_{2,4}, s_{3,2}\}$ may be the best solution, because the aggregated QoS value of the three services is higher than that of other services. However, several services may exist, in which although their QoS values are slightly lower than those three services, their composition could reduce the energy consumption and network resource consumption.

For example, the candidate service $s_{1,3}$ has a marginally higher QoS value (e.g., 0.1 second response time) than $s_{1,4}$. Because $s_{1,4}$ and $s_{2,4}$ run on the same physical server (Host B), running the service composition $s = \{s_{1,4}, s_{2,4}, s_{3,2}\}$ will significantly reduce energy consumption (e.g., 0.1 Watt) and network resource consumption (e.g., 0.1 MB). Then,

when 1000 users invoke the service composition, the traditional schemes would only improve 0.1 second at the cost of 1000×0.1 MB network bandwidth, and 1000×0.1 Watt of power. Unfortunately, users are completely unaware of the 0.1 second improvement in QoS, but for a cloud data center, it means 100 MB network bandwidth and 100 Watt power. Hence, the fundamental challenge is the question of how to find a green service composition scheme that reduces energy consumption and network resource consumption.

The problem is non-trivial and poses a set of unique challenges. First, what is the energy consumption model of service composition in FCDCs? Second, what is the best way to design a network resource consumption model to perform service composition? Third, how can the green service composition problem be formalized and the best solution be found?

4 Green Service Composition Approach

In this section, we solve the above challenges using our proposed green service composition approach. We first construct energy consumption and network resource consumption models of service composition. Then, we formulate green service composition as a multi-objective optimization problem and prove that it is a NP-hard problem. Finally, we use 0-1 linear programming to find the best green composition service.

4.1 Energy Consumption Model

1) Energy consumption of physical machines. The energy consumption of physical machines in the cloud relies on the comprehensive utilization of CPU, memory, disk storage, network interfaces, etc. Among these factors, the CPU is the most important energy consumption component [44], [45], [46]. Hence, based on the above discussion, we define the energy consumption (i.e., P^{pm}) of one physical machine as a function of the CPU utilization as follows:

$$P^{pm} = \sum_{i=1}^{l} x_i \cdot U_i^{CPU} \times (P^{busy} - P^{idle}) + P^{idle}, \quad (3)$$

where l denotes the number of services running on each virtual machine; U_i^{CPU} represents the CPU utilization when the ith service in the physical machine;x_i is a binary decision variable that represents whether a service is invoked in the physical machine (If a service is invoked, the corresponding binary decision variable x_i is set to 1, or 0 if not used); P^{busy} and P^{idle} denotes the power consumed when the physical machine is fully utilized and is idle, respectively; their values can be obtained from [44].

2) Energy consumption of switches. Switches are network hardware devices that consist of port transceivers, line cards, and switch chassis. All of these components contribute to the energy consumption of switches. According to [47], [48], the energy consumption of the switch chassis and line cards remain constant over time, while the consumption of the network ports can scale with the volume of the forwarded traffic as follows [48]:

$$P^{switch} = P_{chassis} + P_{linecard} + \sum_{r=1}^{R} \sum_{i=1}^{m \times n} x_i \cdot n_{sp}^r \times P_{sp}^r \times u_{sp}^r, \quad (4)$$

where $P_{chassis}$ is the energy related to the switch chassis, $P_{linecard}$ is the energy consumed by a single line card; R is the number of line cards plugged into the switch; m denotes the number of service classes; n denotes the number of candidate services in each service class; x_i is a binary decision variable that signifies whether a service is invoked in the switch (If a service is invoked, the corresponding binary decision variable x_i is set to 1, or 0 if not used); P_{sp}^r is the energy drawn by a port running at line card *r;* n_{sp}^r is the number of ports operating at line card *r;* and $u_{sp}^r \in [0, 1]$ is a port utilization that can be defined as follows:

$$u_{sp} = \frac{1}{T \times C_{sp}} \int_t^{t+T} B_{sp}(t)dt, \quad (5)$$

where $B_{sp}(t)$ is the instantaneous throughput at the port's link at time *t;* C_{sp} is the link capacity; and T is a measurement interval [48].

3) Energy consumption of composite services. The energy consumption of each candidate service has certain differences owing to the differences in specific functions and implementation. Different switches often consume different amounts of power owing to the transfer of different candidate services. Then, we can calculate the energy consumption of the composite service using (3) and (4). We take $EC(s)$ to denote the total energy consumption of composite service s, which is defined as follows:

$$EC(s) = \sum_{i=1}^{e \times p} P_i^{pm} + \sum_{i=1}^{a+c+e} P_i^{switch}, \quad (6)$$

where c denotes the number of core switches; a denotes the number of aggregation switches; e denotes the number of edge switches; p denotes the number of physical machines linked to edge switches; P_i^{pm} denotes the energy consumption of candidate services from the ith physical machine; and P_i^{switch} denotes the energy consumption of transferred candidate services by the ith switches.

4.2 Network Resource Consumption Model

For one composite service, if the data transmission over an invoked candidate service needs to go through more switches, this increases the total number of packets delivered by these switches, i.e., the composite service consumes more network resources [43].

Let s_{core} denote the total size of the packet transferred by core switches for composite service s, which can be calculated as follows:

$$s_{core} = \sum_i x_{i,j} \times size(packeti), \quad (7)$$

where $x_{i,j} \in \{0, 1\}$, and $x_{i,j} = 1$ indicates that the jth link is selected to transfer the ith packet of the composite service via the core switches; otherwise, $x_{i,j} = 0$.

Let s_{agg} denote the total size of the packet transferred by the aggregation switches for composite service s, which can be calculated as follows:

$$s_{agg} = \sum_i y_{i,j} \times size(packeti), \quad (8)$$

where $y_{i,j} \in \{0,1\}$, and $y_{i,j} = 1$ indicates that the *j*th link is selected to transfer the *i*th packet of the composite service via the aggregation switches; otherwise, $y_{i,j} = 0$.

Let s_{edge} denote the total size of the packet transferred by the edge switches for composite service s, which can be calculated as follows:

$$s_{edge} = \sum_{i} z_{i,j} \times size(packeti), \tag{9}$$

where $z_{i,j} \in \{0,1\}$, and $z_{i,j} = 1$ indicates that the *j*th link is selected to transfer the *i*th packet of the composite service via the aggregation switches; otherwise $z_{i,j} = 0$.

Then, in order to calculate the network resource consumption of the composite service in the FCDC, we denote the network resource consumption of composite service s by using $NRC(s)$ as follows [43]:

$$NRC(s) = s_{core} + s_{agg} + s_{edge}. \tag{10}$$

4.3 Problem Statement

The problem of finding the greenest service composition by enumerating all possible combinations is considered as an optimization problem. Then, the energy consumption and network resource consumption of the composite service must be minimized while its QoS utility value is higher or lower than a global QoS constraint value $C(0 < C \leq r)$ (In this paper, we chose "lower."). Formally, the optimization problem that we address can be stated as follows:

- The energy consumption of composite service $EC(s)$ is minimized;
- The network resource consumption of composite service $NRC(s)$ is also minimized;
- The QoS utility value of the composite service satisfies $U(s) \leq C$.

4.4 Service Composition Approach

In this problem, we consider three objectives: minimizing overall energy consumption, minimizing the network resource consumption, and maximizing the QoS utility value. It is feasible to find an optimal composition as the number of candidate services, whereas service classes are limited in the cloud. Our green service composition approach in the cloud can be formulated as a multi-objective optimization problem, given by

$$\mathrm{Min}EC(s) = \sum_{i=1}^{e \times p} P_i^{pm} + \sum_{i=1}^{a+c+e} P_i^{switch} \tag{11}$$

$$\mathrm{Min}NRC(s) = s_{core} + s_{agg} + s_{edge} \tag{12}$$

$$\text{subject to} \sum_{k=1}^{r} \left(\frac{Q_k^{\max} - q_k(s)}{Q_k^{\max} - Q_k^{\min}} + \frac{NQ_k^{max} - nq_k(s)}{NQ_k^{max} - NQ_k^{min}} \right) . w_k \leq C, \tag{13}$$

where $w_k \in R^+ (\sum_{k=1}^{r} w_k = 1)$ represents the weight of each QoS attribute; r denotes the number of QoS attributes; Q_k^{max} and Q_k^{min} are the maximum and minimum values of the *k*th service QoS attribute; $q_k(s)$ denotes the aggregated service QoS value of the *k*th service QoS attribute; and C denotes the global QoS constraint value. More explanation about the parameters can be found in Table 1.

Lemma 1. *The green service composition problem is NP-hard.*

Proof. The green service composition problem is also NP-hard Problem. See Appendix B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TSC.2018.2868356, for details. □

In this section, we adopt the weighting method to transform the green service composition problem into a signal objective optimization problem with (Pareto) optimal solution.

As shown in Eqs. (11), (12), and (13), the green service composition problem is formulated as follows:

$$\mathbf{P1}: \begin{cases} find \quad \mathrm{s} = (s_1, s_2, \ldots, s_m)^T \\ which \quad \min EC(s), \min NRC(s) \\ subject \quad to \quad g_k(s) \leq 0; j = 1, 2, \ldots, r \end{cases} \tag{14}$$

With

$$g_k(s) = \sum_{k=1}^{r} \left(\frac{Q_k^{\max} - q_k(s)}{Q_k^{\max} - Q_k^{\min}} + \frac{NQ_k^{max} - nq_k(s)}{NQ_k^{max} - NQ_k^{min}} \right) . \omega_k - C, \tag{15}$$

where s is an m-dimensional vector of decision variables, $EC(s), NRC(s)$ are functions defined on S, and $S = \{s | g_k(s) \leq 0; k = 1, 2, \ldots, r\}$. Then we can transfer (11-13) into (14) as be the same multi-objective optimization problem, i.e., *P1*.

Definition 2. *$\hat{s} \in S$ is said to be a Pareto optimal solution of* P1*, if and only if there does not exsit another $s \in S$ such that $EC(s) \leq EC(\hat{s})$, $NRC(s) \leq NRC(\hat{s})$, with strict inequality holding for at least one.*

Definition 3. *$s^* \in S$ is said to be a weakly Pareto optimal solution of* P1*, if and only if there does not exsit another $s \in S$ such that $EC(s) < EC(\hat{s})$, $NRC(s) < NRC(\hat{s})$.*

In order to get Pareto optimal solution or weakly Pareto optimal solution of *P1*, we modify the multi-objective optimization problem into single objective optimization problem by using the weighting method.

In this way, we suppose that the weighting coefficients w_1 and w_2 are real numbers such that $w_1, w_2 \geq 0$. It is also usually supposed that the weights are normalized, that is $w_1 + w_2 = 1$. Obviously, *P1* is transformed into the following single objective optimization problem, i.e., *P2*:

$$\mathbf{P2}: \begin{cases} find \quad \mathrm{s} = (s_1, s_2, \ldots, s_m)^T \\ which \quad \min(w_1 \cdot EC(s) + w_2 \cdot NRC(s)) \\ subject \quad to \quad g_k(s) \leq 0; j = 1, 2, \ldots, r \end{cases} \tag{16}$$

Where $w_1, w_2 \geq 0$ and $w_1 + w_2 = 1$.

Remark. The exact value of w_1 and w_2 may be difficult to find in practice. Therefore, in practice, we can regard the weight as a tuning parameter, which can be tuned so that the resulting composite service s yields good performance.

Theorem 2. *The solution of* P2 *is a weakly Pareto optimal solution of* P1.

Proof. See Appendix B, available in the online supplemental material for details. □

Theorem 3. *The solution of* P2 *is Pareto optimal if the weighting coefficients are positive, this is* $w_1, w_2 > 0$.

Proof. See Appendix B, available in the online supplemental material for details. □

Based on the above proofs, in this paper, we adopt 0-1 linear programming to solve the multi-objective optimization problem. In particular, we use the models stated above to collect and distribute data for the fat-tree data center in the execution process of the 0-1 linear programming, which is expected to improve efficiency. The 0-1 linear programming has been used to solve service composition problems by several researchers [42], [49]. In our study, binary deci-sion variables are used in the problem to represent the service candidates. A service candidate is selected in the optimal composition if its corresponding variable is set to 1 in the solution of the model and discarded otherwise. By solving using any 0-1 linear programming solver method (IBM CPLEX Optimizer[1] is used in this paper), a list of the best candidate services is obtained and returned to the service broker provided in the cloud. In our approach, particularly, we take energy consumption and network resource consumption into account in the process of execution of the 0-1 linear programming algorithm for the optimization problem. This is expected to make the service composition energy-aware and network-aware.

Remark. If the QoS requirements of a user are very high, our multi-objective optimization problem may not have a feasible solution. In other words, there is no candidate service that can satisfy the requirements of the user. We need to recommend the user to reduce the QoS requirements in this condition.

5 Performance Evaluation

To evaluate our approach, we implement it in WebCloudSim[2], and then compare the results obtained with those from traditional approaches. After extensive experiments, we compare the outcome in terms of energy and network resource consumption as well as QoS utility. The experimental results indicate that our approach is superior to traditional approaches. Moreover, we also analyze the parameters of our approach and present their optimal settings to IT engineers. A case study can be found in Appendix A, available in the online supplemental material.

5.1 Testbed

As stated above, to evaluate our approach, we construct a FCDC network environment consisting of core switches, aggregation switches, edge switches, physical servers, VMs, and services by using WebCloudSim.

1. ttps://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/
2. http://www.webcloudsim.org/

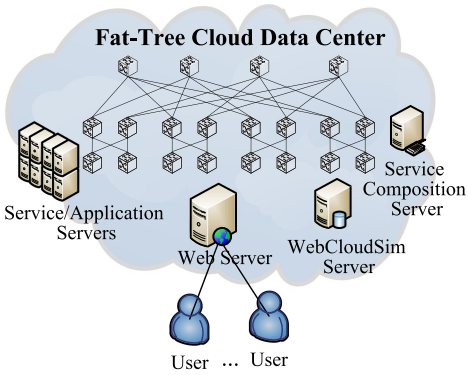


Fig. 2. Overview of the WebCloudSim system constructed for green service composition experiment in the cloud.

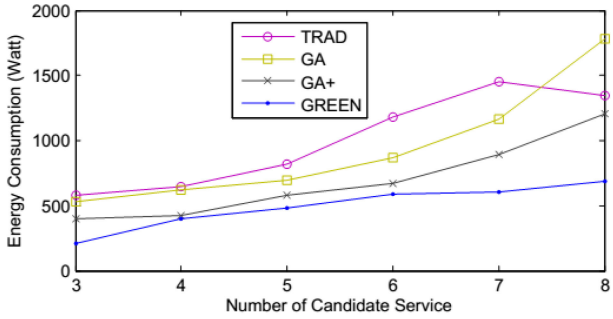
As in our previous work [43], on receiving service composition request from users, as shown in Fig. 2, the web servers in WebCloudSim assign the large number of requests to the service composition servers. The service composition servers then adopt suitable algorithms or methods to find the best services from physical servers. Traditional service composition approaches often select the best services based on service QoS values. In contrast, our green service composition approach pays more attention to energy consumption and network resource consumption. The relevant details about WebCloudSim can be found in our previous work [43], [50].

5.2 Experimental Setup

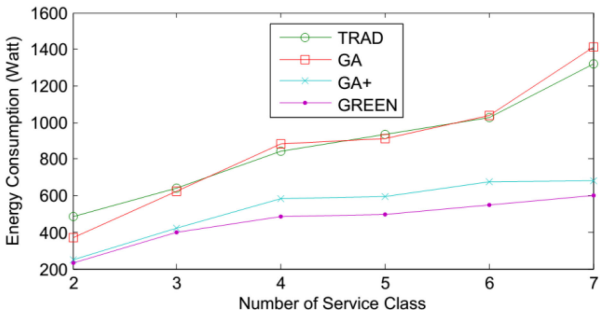
To evaluate our approach, we implement it in WebCloudSim, and the tool is available at http://www.webcloudsim.org/. In the WebCloudSim system, to evaluate our green service composition approach, we set up a 16-port FCDC consisting of 64 core switches and 16 pods. Each pod comprises eight aggregation switches and eight edge switches in a FCDC. We use Cisco Catalyst 6500 and Cisco Nexus 2224TP switches as the aggregation switch and the edge switch, respectively. The former switch contains two linecards and 48 Gigabit ports in each linecard. The power of the fixed part is 472 Watt while that of each port is 3 Watt [51]. Hence, the dynamic part constitutes 38 percent of the switch's power. The latter switch contains one linecard and 24 Gigabit ports. Its fixed part consumes 48 Watt, whereas each port consumes 2 Watt [52], which indicates that the dynamic part constitutes 50 percent of the switch's power. Hence, there are 128 aggregation switches and 128 edge switches.

The bandwidth of the core and aggregation switches is set as 10 Gps and the bandwidth of the edge switch as 1 Gps. Each edge switch could connect to eight physical servers that are divided into two categories: HP ProLiant G4 with a 3720 MIPS CPU and 4 GB memory, and HP ProLiant G5 with a 5320 MIPS CPU and 4 GB memory.

Each physical server hosts four VMs, and each VM hosts two services. Therefore, the data center contains 1024 host servers, 4096 VMs, and 8192 services. For the VM configuration, the base system is 769 MB; the RAM disk is 5.3 MB; the kernel is 1.6 MB; the memory size is 512 MB; and the user disk size is 1 GB. There are 8192 services from the data center and each service contains three QoS attributes (specifically, delay, throughput, and reliability). In order to ensure the repeatability of experiments, we use the real service QoS



(a) Energy consumption with varying number of candidate services for service composition in the cloud.



(b) Energy consumption with varying number of service classes for service composition in the cloud.

Fig. 3. Comparison of energy consumption of service composition in the cloud. Compared with TRAD, GA, and GA+, GREEN saves, respectively, more than 50 percent, approximately 40 percent, and approximately 20 percent on average in energy consumption regardless of the number of service classes or candidate services in use.

dataset [53], [54] as the QoS data of the three attributes for the 8192 services in the WebCloudSim system.

Remark. The created 1,000 service composition requests in the WebCloudSim system are randomly executed in one hour. It is worth nothing that the number of concrete services of a composite service is generally less than 10 in practice [55]. The values of the weighting coefficient are set to 0.5. Hence, in this experiment, the number of candidate services and service classes is less than 10. For example, when the number of service classes is only three and the number of candidate services only is three, at most 1000×3^3 services are used in the service composition process, i.e., many services are used multiple times. Other parameters settings are the same as in our previous work [43].

5.3 Compared Approaches

In order to evaluate the performance of our green service composition approach, we compare it with the following service composition approaches. All experiments are conducted on the same data center with identical configuration. The experiments are conducted 20 times repeatedly.

- *TRAD*. These traditional approaches (TRAD) often focus on traditional QoS utility, and adopt 0-1 linear programming to find the best solution based on traditional QoS utility function [41], [42]. This is a standard approach [40], [41], [42] to solve the service composition problem. The relative standard deviation of the results is lower than 7 percent.
- *GA*. Genetic algorithms [26], [56] belong to the larger class of evolutionary algorithms and are also often used to find optimal service compositions, via uniform crossover and uniform mutation. The initial population for the GA is set to 20, and the maximum number of iteration is set to 30 [57].

Remark. We supply this basic genetic algorithms (called GA) with the standard service composition problem, e.g., the traditional QoS fitness function. The mutation and crossover operator settings are from [54]. The relative standard deviation of the results is lower than 9 percent.

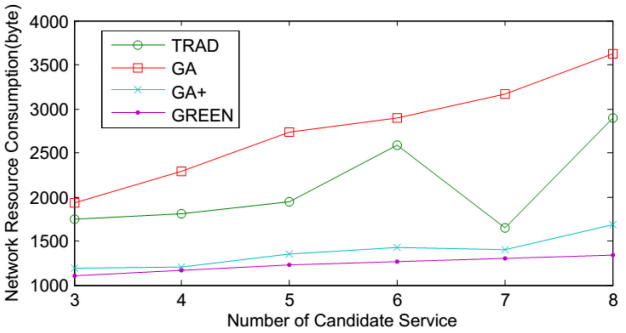
- *GA+*. It is a green-aware extension of basic genetic algorithms (called GA+). In addition to the settings used for GA+, we extend it with our proposed energy consumption model and network resource consumption model. The fitness function is the sum of (11) and (12). The parameters settings for GA+ and GA are same. The relative standard deviation of the results is lower than 9 percent.
- *GREEN*. Our proposed green service composition approach is described in Section 4. The relative standard deviation of the results is lower than 7 percent.

5.4 Comparison Results on Energy Consumption

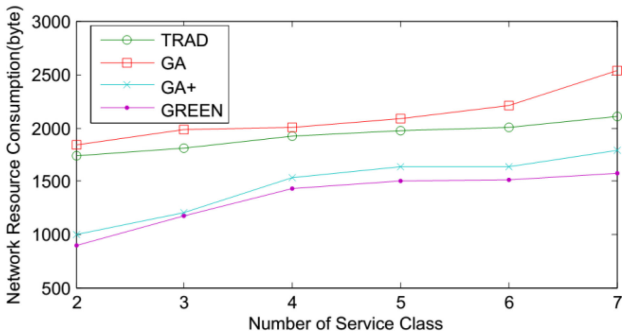
Fig. 3 shows that our GREEN enabled cloud service saves more energy than other approaches. For example, in Fig. 3a, the energy consumption of our GREEN approach is about 496 Watt on average, but those of the other approaches (e.g., TRAD, at about 1,005 Watt, and GA, at about 964 Watt) are much more expensive in this respect. TRAD, GA, GA+, and GREEN have energy consumption ratios of approximate 2.0, 1.9, 1.4, and 1.0, respectively. When there is only one service composition requirement, our GREEN approach can save more than 300 Watt of energy on average. When there is a large number of users (e.g., 1,000,000) who use service composition in the cloud, GREEN can save a small thermal power station's energy output. Similarly, in Fig. 3b, the energy consumption of our GREEN is approximately 460 Watt on average, but that of the other approaches (e.g., TRAD, at about 870 Watt) is much more on average. TRAD, GA, GA+, and GREEN have energy consumption ratios of approximate 1.9, 1.9, 1.2, and 1.0, respectively.

With reference to Fig. 3, compared with TRAD, our GREEN approach saves more than 50 percent on average in energy regardless of the number of service classes or candidate services in use. Compared with GA, our GREEN approach saves approximately 40 percent on average in energy. Compared with GA+, our GREEN approach saves approximately 20 percent on average in energy. This means that our GREEN approach can significantly reduce energy consumption, and is the best among all of the evaluated approaches.

In contrast to TRAD and GA, our GREEN energy consumption model for service composition in FCDCs gives adequate consideration to energy consumption of physical machines and switches in the cloud. In contrast to GA+, our GREEN approach uses 0-1 linear programming to

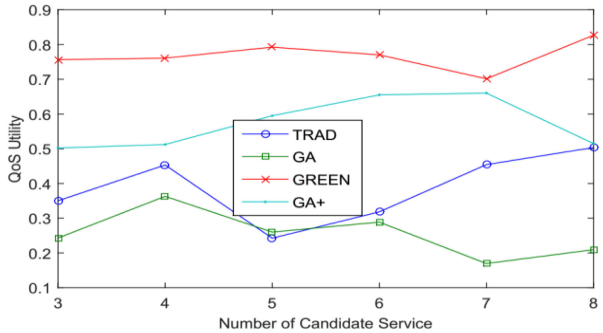


(a) Network resource consumption with varying number of candidate services for service composition in the cloud.

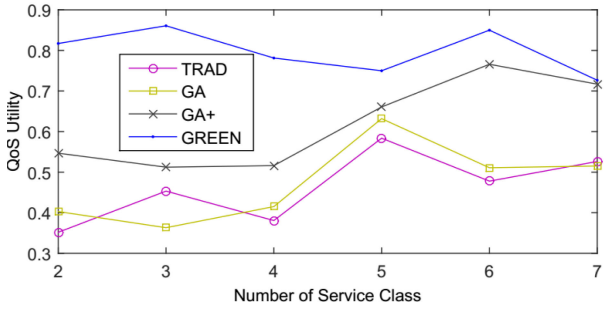


(b) Network resource consumption with varying number of service classes for service composition in the cloud.

Fig. 4. Comparison of network resource consumption for service composition in the cloud. Compared with TRAD and GA, our GREEN method saves approximately 50 percent on average in network resources regardless of the number of service classes or candidate services in use. Compared with GA+, our GREEN saves approximately 10 percent on average in network resources.



(a) QoS utility with varying number of candidate services for service composition in the cloud.



(b) QoS utility with varying number of service classes for service composition in the cloud.

Fig. 5. Comparison of QoS utility of service composition in the cloud. Compared with TRAD and GA, GREEN improves by approximately 50 percent on average in QoS utility regardless of the number of service classes or candidate services in use. Compared with GA+, GREEN improves by approximately 20 percent on average.

optimize the cloud service composition problem. Owing to the small search space of the cloud service composition problem, GREEN is more effective than GA+, which may be trapped in local optima. Moreover, because GA+ adopts our proposed energy model, it saves more energy than GA, which also confirms that our proposed model is very effective.

In summary, GREEN significantly reduces energy consumption in solving the service composition problem in the cloud.

5.5 Comparison Results on Network Resource Consumption

Fig. 4 illustrates the comparison of network resource consumption for the considered approaches, where the size of the network packet is set at 800 bytes. The figure clearly shows three significant facts.

First, our GREEN enabled cloud service saves more network resources than other approaches. It can be seen that its network resource consumption is less than 1300 bytes on average for the various candidate services and service classes. The results confirm that GREEN is an effective solution to cope with the service composition problem in the cloud.

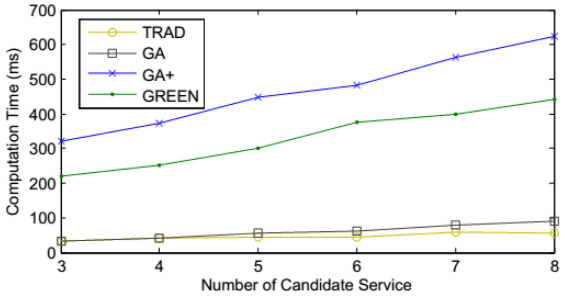
Second, compared with the other considered alternatives, the GREEN solution achieves a significantly higher energy saving (higher than 50 or 10 percent). For example, compared with TRAD and GA, our GREEN method saves approximately 50 percent on average in network resources regardless of the number of candidate services or service classes in use. Compared with GA+, GREEN saves approximately 10 percent on average in network resources. This means that our GREEN approach can significantly reduce network resource consumption, and is the best among all of the considered alternatives.

Third, as the number of candidate services or service classes increases, the energy saving is reduced. This last effect can be explained considering that, as the number of services grows, every evaluated approach must consider more physical machines and switches to configure the service composition system. This behavior clearly increases the energy consumption. However, even as the number of services increases, GREEN still outperforms the other considered approaches.

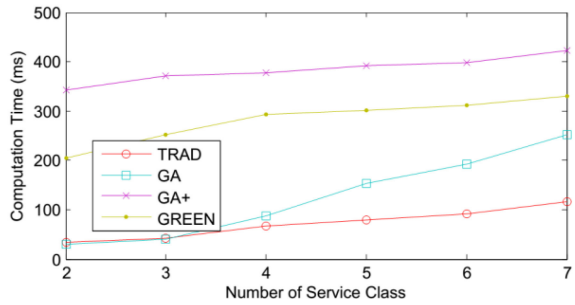
In summary, in contrast to TRAD and GA, our GREEN approach considers the topology of fat-tree cloud datacenter networks, making full use of the network resource consumption model to find the optimal solution in the cloud. In contrast to GA+, 0-1 linear programming is more effective for small search space optimization. Moreover, our previous work [43] also validates the conclusion of this experiment. Hence, GREEN significantly achieves higher network-resource-efficiency in solving the service composition problem in the cloud.

5.6 Comparison Results on QoS Utility

Fig. 5 illustrates the comparative evaluation results for QoS utility, including network QoS. As shown in Fig. 5, it can be seen that GREEN can significantly improve the QoS utility of service composition in the cloud. For example, in Fig. 5a, the QoS utility of GREEN is about 0.77 on average, whereas



(a) Computation time with varying number of candidate services for service composition in the cloud.



(b) Computation time with varying number of service classes for service composition in the cloud.

Fig. 6. Comparison of computation time for service composition in the cloud. Although the computation time of the GREEN approach is not the lowest, its computation time is very low regardless of the number of service classes or candidate services in use. This means that GREEN still can provide a fast composite service in the cloud.

other approaches attain lower values on average. TRAD, GA, GA+, and GREEN have QoS utility ratios of about 0.5, 0.3, 0.7, and 1.0, respectively. Similarly, Fig. 5b shows ratios of approximately 0.6, 0.6, 0.8, and 1.0, respectively.

With reference to Fig. 5, compared with TRAD and GA, GREEN improves by approximately 50 percent on average in QoS utility regardless of the number of service classes or candidate services in use. Compared with GA+, GREEN improves by approximately 20 percent on average in terms of QoS utility. This means that GREEN can effectively assure QoS utility, and is the best among all of the evaluated approaches.

In contrast to TRAD and GA, GREEN considers the network QoS of service composition in the cloud. In contrast to GA+, because, in practice, the solution space of service composition problems in the cloud is not very large, our GREEN approach, which adopts 0-1 linear programming, is more effective than GA+ in the context of fat-tree cloud datacenter networks. Moreover, our previous work [43] also validates the conclusion of this experiment.

In summary, our GREEN approach significantly assures realistic QoS utility when solving service composition problems in the cloud.

5.7 Comparison Results on Computation Time

As shown in Fig. 6, we find that the computation time for GREEN is very low, and is about 300 ms on average regardless of the number of service classes or candidate services in use. Compared with GA+, GREEN reduces the computation time by approximately 30 percent on average. Although the computation time for GREEN is not the lowest among the approaches, it still provides a composite service in the cloud.

Finally, as Figs. 3, 4, 5, and 6 show that, our GREEN approach can achieve the best green solutions (low energy consumption and network resource consumption, and high QoS utility) with low computation time. This means that cloud service providers can adopt our approach to provide green cloud services (i.e., SaaS and IaaS) via a central control with all levels of information of services and network in cloud data centers [43].

5.8 Sensitivity Analysis of GREEN

In order to fully evaluate the proposed GREEN, we test our proposal under two operating scenarios (i.e., the bandwidth setting of edge switches, the bandwidth setting of core and aggregation switch) to understand how the FCDC characteristics affect its performance. We hope the results can help IT engineers to implement our approach better on their cloud systems.

5.8.1 Sensitivity to the Bandwidth of Edge Switches

The first sensitivity experiment focuses on the effect of the bandwidth of the edge switch on network resource consumption, QoS utility, and computation time for the different bandwidths of edge switches.

Fig. 7a shows that, as the bandwidth of edge switches grows, the energy consumption of GREEN gradually decreases. The intuitive effect can be explained that, 1) because the waiting time of service composition becomes shorter, GREEN shuts down (one or) several physical machines running services; or 2) (one or) several ports of edge switches are closed because a small number of ports can satisfy the bandwidth requirement of service composition in the cloud.

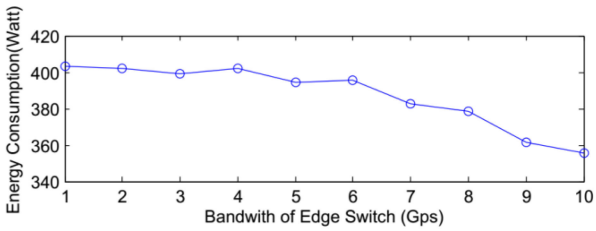
Fig. 7b shows that with increasing bandwidth of edge switches, the network resource consumption of GREEN also increases. The counterintuitive effect can be explained by the fact that, as the bandwidth of edge switches grows, GREEN tends to composite the services that are in the different subnets owing to the QoS constraint of (13) (as shown in Fig. 7c) in the cloud. Fig. 7c shows that, as the bandwidth of edge switches grows, the QoS utility of GREEN gradually increases. But when the QoS utility value of the composite service satisfies the constraint of (13), GREEN tends to reduce more energy and network resource consumption. This means that the QoS utility tends to decrease when the bandwidth of the edge switches is higher than 6 Gps.

Fig. 7d shows that with the increasing bandwidth of edge switches, the computation time of GREEN also decreases and tends to become stable. The intuitive effect can be explained that, because the waiting time of service composition becomes shorter, GREEN saves on computation time. Moreover, when the bandwidth of the edge switches is higher than 4 Gps, the computation time tends to be stable.

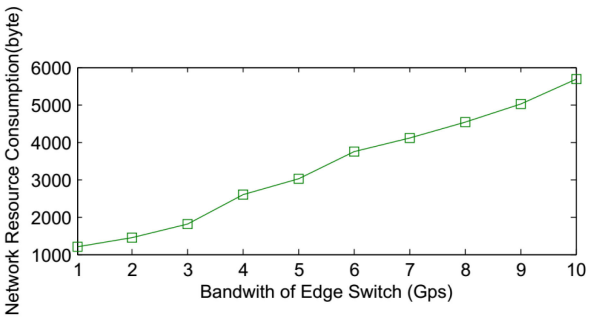
5.8.2 Sensitivity to the Bandwidth of Core and Aggregation Switches

The second sensitivity experiment focuses on the effect of the bandwidth of the edge switch on network resource consumption, QoS utility, and computation time for the different bandwidths of core and aggregation switches.

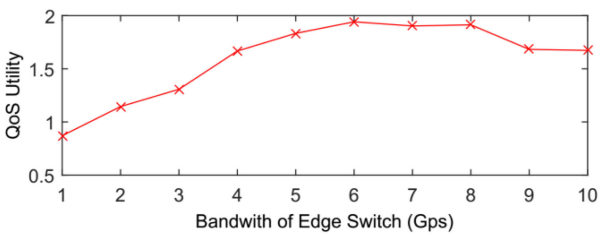
From Fig. 8, it is clear that, as the bandwidth of core and aggregation switches grows, the sensitivity analysis results



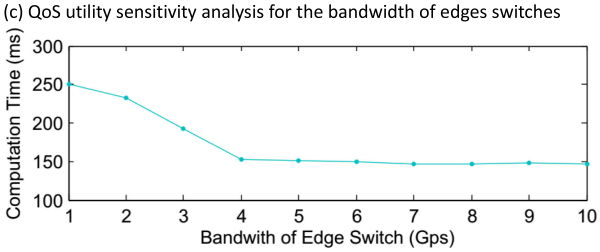
(a)Energy consumption sensitivity analysis for the bandwidth of edges switches



(b) Network resource consumption sensitivity analysis for the bandwidth of edges switches

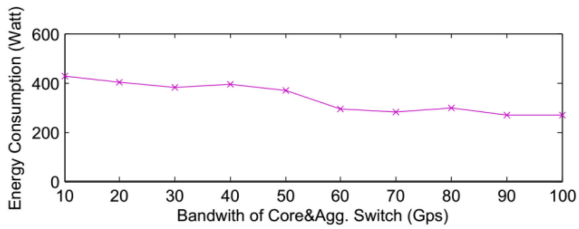


(c) QoS utility sensitivity analysis for the bandwidth of edges switches

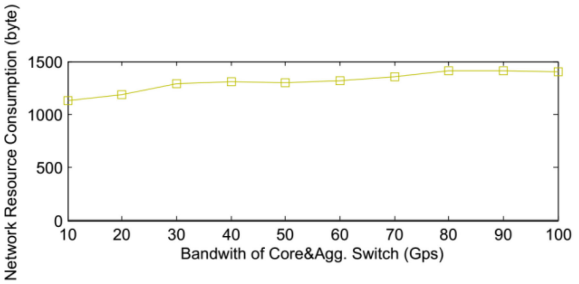


(d) Computation time sensitivity analysis for the bandwidth of edges switches

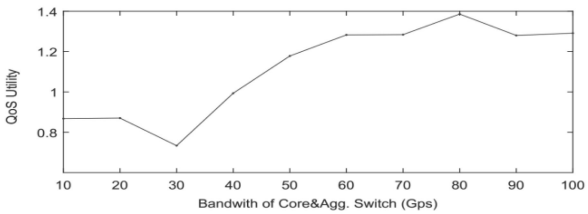
Fig. 7. Sensitivity to the bandwidth of edge switches for energy consumption, network resource consumption, QoS utility, and computation time.



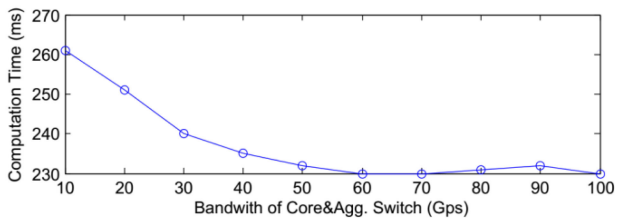
(a) Energy consumption sensitivity analysis for the bandwidth of core and aggregation switches



(b) Network resource consumption sensitivity analysis for the bandwidth of core and aggregation switches



(c) QoS utility sensitivity analysis for the bandwidth of core and aggregation switches



(d) Computation time sensitivity analysis for the bandwidth of core and aggregation switches

Fig. 8. Sensitivity to the bandwidth of core and aggregation switches for energy consumption, network resource consumption, QoS utility, and computation time.

become similar to Fig. 8. For detailed explanation of the sensitivity analysis in Fig. 8, refer to Section 5.8.1. Hence, in short, as the bandwidth of core and aggregation switches grows, Fig. 8 shows that: 1) the energy consumption of GREEN is reduced; 2) the network resource consumption of GREEN is increased; 3) the QoS utility of GREEN gradually increases, but tends to decrease when the bandwidth of the core and aggregation switches is higher than 80 Gps; and 4) the computation time of GREEN also decreases and tends to become stable when the bandwidth of the core and aggregation switches is higher than 60 Gps.

5.9 Open Issues and Limitations

5.9.1 Open Issues

The following four issues are open issues for our proposed approach.

1) In traditional service selection or composition approaches, including our many previous work, the number of candidate services is often at least more than 100, and the number of service classes is often at least more than 10. Interestingly, we can not find any service composition application based on traditional approaches in any real-world IT systems. It is worth noting that, although the number of services in the cloud still grows, the number of concrete services comprising a composite service is generally less than 10 in practical systems [55]. Hence, we believe that the computation time is not a critical issue at the current stage for service composition owing to the small search space and not a large number of services that need to be composited.
2) Traditional service selection or composition approaches, including our several previous studies, focus more on service composition algorithm optimization. They often try to adopt more complex algorithms (such as genetic algorithm, particle swarm optimization, and ant colony optimization) to find the best solution. However, in practical systems, finding the optimal composition requires enumerating all

possible combinations of candidate services, which is not expensive in terms of computation time owing to the small search space. Hence, a basic service composition algorithm such as 0-1 linear programming is good enough to find the best solution. Therefore, we believe that our service composition algorithm is suitable, and the basic algorithm is sufficient and effective.

3) Who will use our proposed approach and how to use it? Cloud service vendors, such as Amazon AWS, Microsoft Azure and Alibaba Cloud, deploy various services in their cloud data centers, including their own services and services from third-parties. Many of these services can be used for composition and many of them are functionally equivalent. As such, cloud infrastructure providers can adopt and benefit from our approach to reduce the energy and network resource consumption in their cloud data centers. The hypertext-driven REST API [80], which consists of connected REST resources that provide different services through uniform interfaces or Service Mesh, can support the implementation of our approach in real cloud data centers. It includes services from both data and control planes, such as switches, routers, subnets, networks, NAT devices, and controllers in SDN environments.
4) In the early stages of cloud computing, it could be difficult to dynamically co-host services that need to be composed frequently on the same (or close) physical location. However, with the advance of container and virtualization technology, we can adopt virtual machine migration and fast container deployment technology to host these services on the same or nearby physical servers to make green service composition practical.

5.9.2 Limitations of Our Approach

There are several limitations of our proposed approach as follows:

1) It is not easy to conduct a large scale evaluation within a cloud data center, but there are still possibilities to create a small replica of a cloud data center, which is running on commodity hardware. Although this setup cannot host the same set of services as those used for the simulation, it can be used to monitor actual data instead of just summing up predefined numbers during a simulation.
2) The fact that an actual evaluation would support the usefulness of our proposed approach, there are also flaws or missing information for the simulation. The simulation assumes that there are no packet drops during the evaluation, which is not the case in real cloud data centers. Hence, we hope to have a collaboration with top cloud service vendors to implement our approach in a real cloud data center and further evaluate the performance.

6 Conclusions

In this paper, we propose a green service composition approach for FCDCs. Our approach not only consists of an overall energy consumption model and network resource consumption model, but also has a service composition model to find the best green (such as low energy consumption, low network resource consumption) composition service in the cloud. In contrast to current approaches in this area, our approach links services, networks, physical machines, and switches and considers energy and network resource consumption in the service composition process. We implement the proposed approach in the WebCloudSim system, and find that it outperforms current approaches in terms of energy consumption, network resource consumption, QoS utility, and computation time.

Future work will explore a good means of enhancing our proposed approach in software defined network (SDN) environments. Moreover, the question of how to design a service-composition-aware virtual machine or container migration scheme will also be explored.

Acknowledgments

This work was supported in part by the National Science Foundation of China (No. 61472047), National Science Foundation of China(No. 61602054), and Beijing Natural Science Foundation (No. 4174100).

References

[1] A. Jula, E. Sundararajan, and Z. Othman, "Cloud computing service composition: A systematic literature review," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3809–3824, 2014.
[2] K. Kritikos and D. Plexousakis, "multi-cloud application design through cloud service composition," in *Proc. IEEE 8th Int. Conf. Cloud Comput.*, 2015, pp. 686–693.
[3] F. Alrebeish and R. Bahsoon, "Stabilising performance in cloud services composition using portfolio theory," in *Proc. IEEE 22th Int. Conf. Web Services*, 2015, pp. 1–8.
[4] V. Dastjerdi and R. Buyya, "Compatibility-aware cloud service composition under fuzzy preferences of users," *IEEE Trans. Cloud Comput.*, vol. 2, no. 1, pp. 1–13, Jan.-Mar. 2014.
[5] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, A. P. Sharma, A. S. Banerjee, and A. N. McKeown, "ElasticTree: Saving energy in data center networks," in *Proc. USENIX 7th Conf. Netw. Syst. Des. Implementation*, 2010, pp. 17–17.
[6] W. Lin, Z. Fa, J. Arjona Aroca, A. V. Vasilakos, Z. Kai, H. Chenying, L. Dan, and L. Zhiyong, "GreenDCN: A general framework for achieving energy efficiency in data center networks," *IEEE J. Selected Areas Commun.*, vol. 32, no. 1, pp. 4–15, Jan. 2014.
[7] F. Xiang, Y. Hu, Y. Yu, and H. Wu, "QoS and energy consumption aware service composition and optimal-selection based on Pareto group leader algorithm in cloud manufacturing system," *Central Eur. J. Operations Res.*, vol. 22, no. 4, pp. 663–685, 2014.
[8] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J.-M. Pierson, and A. V. Vasilakos, "Cloud computing: Survey on energy efficiency," *ACM Comput. Surveys*, vol. 47, no. 2, pp. 1–36, 2014.
[9] W. Lin, Z. Fa, Z. Kai, A. V. Vasilakos, R. Shaolei, and L. Zhiyong, "Energy-efficient flow scheduling and routing with hard deadlines in data center networks," in *Proc. IEEE 34th Int. Conf. Distrib. Comput. Syst.*, 2014, pp. 248–257.
[10] J. Liu, J. Jiang, X. Cui, W. Yang, and X. Liu, "Power consumption prediction of web services for energy-efficient service selection," *Personal Ubiquitous Comput.*, vol. 19, no. 7, pp. 1063–1073, 2015.
[11] C. Sandionigi, D. Ardagna, G. Cugola, and C. Ghezzi, "Optimizing service selection and allocation in situational computing applications," *IEEE Trans. Services Comput.*, vol. 6, no. 3, pp. 414–428, Jul.-Sep. 2013.
[12] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2008, pp. 63–74.
[13] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *Proc. IEEE 29th Conf. Inf. Commun.*, 2010, pp. 1–9.
[14] N. Limrungsi, J. Zhao, Y. Xiang, T. Lan, H. H. Huang, and S. Subramaniam, "Providing reliability as an elastic service in cloud computing," in *Proc. IEEE Int. Conf. Commun.*, 2012, pp. 2912–2917.

[15] J. Xu, J. Tang, K. Kwiat, W. Zhang, and G. Xue, "Survivable virtual infrastructure mapping in virtualized data centers," in *Proc. IEEE 5th Int. Conf. Cloud Comput.*, 2012, pp. 196–203.
[16] P. Bartalos and M. B. Blake, "Green web services: Modeling and estimating power consumption of web services," in *Proc. IEEE 19th Int. Conf. Web Services*, 2012, pp. 178–185.
[17] C. Zeng, X. Guo, W. Ou, and D. Han, "Cloud computing service composition and search based on semantic," in *Proc. IEEE Int. Conf. Cloud Comput.*, 2009, pp. 290–300.
[18] T. Wei-Tek, Z. Peide, J. Balasooriya, C. Yinong, B. Xiaoying, and J. Elston, "An approach for service composition and testing for cloud computing," in *Proc. IEEE 10th Int. Symp. Autonomous Decentralized Syst.*, 2011, pp. 631–636.
[19] H. Cai and L. Cui, "Cloud service composition based on multi-granularity clustering," *J. Algorithms Comput. Technol.*, vol. 8, no. 2, pp. 143–162, 2014.
[20] S. Sundareswaran, A. Squicciarini, and D. Lin, "A brokerage-based approach for cloud service selection," in *Proc. IEEE 5th Int. Conf. Cloud Comput.*, 2012, pp. 558–565.
[21] W. Zeng, Y. Zhao, and J. Zeng, "Cloud service and service selection algorithm research," in *Proc. ACM 1st World Summit Genetic Evol. Comput.*, 2009, pp. 1045–1048.
[22] H. Qiang, H. Jun, C. Feifei, W. Yanchun, R. Vasa, Y. Yun, and J. Hai, "QoS-aware service selection for customisable multi-tenant service-based systems: Maturity and approaches," in *Proc. IEEE 8th Int. Conf. Cloud Comput.*, 2015, pp. 237–244.
[23] J. O. Gutierrez-Garcia and K.-M. Sim, "Self-organizing agents for service composition in cloud computing," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci.*, 2010, pp. 59–66.
[24] J. O. Gutierrez-Garcia and K. Sim, "Agent-based cloud service composition," *Appl. Intell.*, vol. 38, no. 3, pp. 436–464, 2013.
[25] Z. Ye, A. Bouguettaya, and X. Zhou, "QoS-aware cloud service composition based on economic models," in *Proc. IEEE 5th Int. Conf. Service-Oriented Comput.*, 2012, pp. 111–126.
[26] A. Klein, F. Ishikawa, and S. Honiden, "Towards network-aware service composition in the cloud," in *Proc. ACM 21st Int. Conf. World Wide Web*, 2012, pp. 959–968.
[27] L. Qu, Y. Wang, M. A. Orgun, L. Liu, H. Liu, and A. Bouguettaya, "CCCloud: Context-aware and credible cloud service selection based on subjective assessment and objective assessment," *IEEE Trans. Services Comput.*, vol. 8, no. 3, pp. 369–383, May/Jun. 2015.
[28] Z. Zheng, X. Wu, Y. Zhang, M. R. Lyu, and J. Wang, "QoS ranking prediction for cloud services," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1213–1222, Jun. 2013.
[29] S. Wang, Q. Sun, H. Zou, and F. Yang, "Particle swarm optimization with skyline operator for fast cloud-based web service composition," *Mobile Netw. Appl*, vol. 18, no. 1, pp. 116–121, 2013.
[30] S. Wang, Z. Liu, Q. Sun, H. Zou, and F. Yang, "Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing," *J. Intell. Manuf.*, vol. 25, no. 2, pp. 283–291, 2014.
[31] W. Shangguang, Z. Zibin, S. Qibo, Z. Hua, and Y. Fangchun, "Cloud model for service selection," in *Proc. IEEE Int. Conf. Comput. Commun. Workshops*, 2011, pp. 666–671.
[32] S. Deng, L. Huang, J. Taheri, and A. Y. Zomaya, "Computation offloading for service workflow in mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 12, pp. 3317–3329, Dec. 2015.
[33] U. Wajid, C. A. Marin, and A. Karageorgos, "Optimizing energy efficiency in the cloud using service composition and runtime adaptation techniques," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2013, pp. 115–120.
[34] H. Jiwei and L. Chuang, "Agent-based green web service selection and dynamic speed scaling," in *Proc. IEEE 20th Int. Conf. Web Services*, 2013, pp. 91–98.
[35] S. Deng, H. Wu, D. Hu, and J. L. Zhao, "Service selection for composition with QoS correlations," *IEEE Trans. Serv. Comput.*, vol. 9, no. 2, pp. 291–303, Mar./Apr. 2016.
[36] C. Ying, H. Jiwei, X. Xudong, and L. Chuang, "Energy efficient dynamic service selection for large-scale web service systems," in *Proc. IEEE Int. Conf. Web Services*, 2014, pp. 558–565.
[37] J. Um, Y.-C. Choi, and I. Stroud, "Factory planning system considering energy-efficient process under cloud manufacturing," *Procedia CIRP*, vol. 17, pp. 553–558, 2014.
[38] Z. Li, J. Ge, H. Hu, W. Song, and B. Luo, "Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds," *IEEE Trans. Serv. Comput.*, vol. 11, no. 4, pp. 713–726, Jul./Aug. 2018.
[39] J. H. Jang, D. H. Shin, and K. H. Lee, "Fast quality driven selection of composite Web services," in *Proc. 4th Eur. Conf. Web Serv.*, 2006, pp. 87–96.
[40] D. Ardagna and B. Pernici, "Adaptive service composition in flexible processes," *IEEE Trans. Softw. Eng.*, vol. 33, no. 6, pp. 369–384, Jun. 2007.
[41] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for web services composition," *IEEE Trans. Softw. Eng.*, vol. 30, no. 5, pp. 311–327, May 2004.
[42] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for web services selection with end-to-end QoS constraints," *ACM Trans. Web*, vol. 1, no. 1, pp. 1–26, 2007.
[43] S. Wang, A. Zhou, F. Yang, and R. N. Chang, "Towards network-aware service composition in the cloud," *IEEE Trans. Cloud Comput.*, 2016, https://ieeexplore.ieee.org/document/7553440/
[44] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency Comput.: Practice Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.
[45] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Future Generation Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012.
[46] P. Reviriego, V. Sivaraman, Z. Zhao, J. A. Maestro, A. Vishwanath, A. Sanchez-Macian, and C. Russell, "An energy consumption model for energy efficient ethernet switches," in *Proc. IEEE 10th Int. Conf. High Perform. Comput. Simul.*, 2012, pp. 98–104.
[47] X. Jing and J. A. B. Fortes, "Multi-objective virtual machine placement in virtualized data center environments," in *Proc. IEEE/ACM Int. Conf. Green Comput. Commun.*, 2010, pp. 179–188.
[48] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Models for efficient data replication in cloud computing datacenters," in *Pro. IEEE Int. Conf. Commun.*, 2015, pp. 6056–6061.
[49] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient QoS-aware service composition," in *Proc. IEEE 18th Int. Conf. World Wide Web*, 2009, pp. 881–890.
[50] A. Zhou, S. Wang, B. Cheng, Z. Zheng, F. Yang, R. Chang, M. Lyu, and R. Buyya, "Cloud service reliability enhancement via virtual machine placement optimization," *IEEE Trans. Serv. Comput.*, vol. 10, no. 6, pp. 902–913, Nov./Dec. 2016.
[51] "Cisco Catalyst 6500 Series Switches, 2016. [Online]. Available: http://www.cisco.com/c/en/us/products/switches/catalyst-6500-series-switches/index.html
[52] Cisco Nexus 2000 series data sheet, 2016. [Online]. Available: http://www.cisco.com/c/en/us/products/collateral/switches/nexus-2000-series-fabric-extenders/data_sheet_c78-507093.html
[53] E. Al-Masri and Q. H. Mahmoud, "Discovering the best web service," in *Proc. IEEE 16th Int. Conf. World Wide Web Conf.*, 2007, pp. 1257–1258.
[54] Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed QoS evaluation for real-world Web services," in *Proc. IEEE 8th Int. Conf. Web Serv.*, 2010, pp. 83–90.
[55] J. Lin, B. Liu, N. Sadeh, and J. I. Hong, "Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings," in *Proc. IEEE Symp. Usable Privacy Security*, 2014, pp. 199–212.
[56] G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani, "An approach for QoS-aware service composition based on genetic algorithms," in *Proc. ACM 7th Annu. Conf. Genetic Evol. Comput.*, 2005, pp. 1069–1075.
[57] R. L. Haupt, "Optimum population size and mutation rate for a simple real genetic algorithm that optimizes array factors," in *Proc. IEEE Antennas Propagation Society Int. Symp.*, 2000, pp. 1034–1037.

Shangguang Wang received the PhD degree from the Beijing University of Posts and Telecommunications, in 2011. He is an associate professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (BUPT). His research interests include Service Computing, Cloud Computing and Edge Computing. He is a senior member of the IEEE.

Ao Zhou received the PhD degree in computer science from the Beijing University of Posts and Telecommunications of China, in 2015. She is an associate professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Her research interests include cloud computing, service reliability.

Ruo Bao received the bachelor's degree in computer science from the Beijing University of Posts and Telecommunications of China, in 2014. He is working toward the graduate degree at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He research interests include cloud computing and services computing.

Wu Chou received the PhD degree in electrical engineering from Stanford University, Stanford, CA. After graduating from Stanford with four advanced degrees in science and engineering, he continued his professional career from AT&T Bell Labs to Lucent Bell Labs and Avaya Labs before joining Huawei as the Head of Huawei Shannon (IT) Laboratory. Currently, he is VP and Chief Technology Officer of Network and Enterprise Communications at Huawei. His research interests include Cloud computing, data networking, SDN/NFV, Internet-of-Things (IoT), big data, machine learning, communication, Internet/Web, service computing, and Web services. He is a fellow of the IEEE.

Stephen S. Yau received the PhD degree from the University of Illinois, Urbana electrical engineering. He is professor of computer science and engineering at Arizona State University (ASU), Tempe, Arizona. Previously, he was on the faculties of Northwestern University, Evanston, Illinois, and University of Florida, Gainesville. He served as the president of the IEEE Computer Society and the editor-in-chief of IEEE computer magazine. He is the general chair of the 2018 IEEE World Congress on Services. His current research includes services and cloud computing systems, cyber security, software engineering, internet of things, and ubiquitous computing. He has received various awards and recognitions, including the Tsutomu Kanai Award and Richard E. Merwin Award of the IEEE Computer Society, and the Outstanding Contributions Award of the Chinese Computer Federation. He is a life fellow of the IEEE and a fellow of the American Association for the Advancement of Science.