Provision of Data-intensive Services through Energy- and QoS-aware Virtual Machine Placement in National Cloud Data Centers

S. Wang, A. Zhou, C. Hsu, *Senior Member*, *IEEE*, X. Xiao, F. Yang, *Senior Member*, *IEEE*

Abstract—Many data-intensive services (e.g., planet analysis, gene analysis, etc.) are becoming increasingly reliant on national cloud data centers because of growing scientific collaboration among countries. In national cloud data centers, tens of thousands of virtual machines are assigned to physical servers to provide data-intensive services with a quality-of-service (QoS) guarantee, and consume a massive amount of energy in the process. Although many virtual machine placement schemes have been proposed to solve this problem of energy consumption, most of these assume that all the physical servers are homogeneous. However, the physical server configurations of national cloud data centers often differ significantly, which leads to varying energy consumption characteristics. In this paper, we explore an alternative virtual machine placement approach to minimize energy consumption during the provision of data-intensive services with a global QoS guarantee in national cloud data centers. We use an improved particle swarm optimization (PSO) algorithm to develop an optimal virtual machine placement approach involving a tradeoff between energy consumption and global QoS guarantee for data-intensive services. Experimental results based on an extended version of the CloudSim framework show that our approach significantly outperforms other approaches to energy optimization and global QoS guarantee in national cloud data centers.

Index Terms—cloud computing; data-intensive service; national cloud data center; virtual machine placement; energy consumption; QoS

1 Introduction

With the increasing popularity of cloud computing in recent times, many countries and organizations have begun building national cloud data centers (NCDCs) to support collaboration in scientific research [1,2]. These are exemplified by the Galileo project[1], A Toroidal LHC Apparatus experiment[2], and the Coordination Group for Meteorological Satellites[3]. NCDCs are different from public cloud data centers, which focus on processing public and non-profit service requirements. In general, these service requirements are related to big data and the processing of large tasks, i.e., data-intensive services (e.g., large scale traffic data analysis). For example, five NCDCs in China are primarily dedicated to providing data-intensive services for public service departments (e.g., Meteorology, Resources, Health, and Traffic) and international collaboration in research.

NCDCs are different from public cloud data centers, which do not provide services to consumers in business scenarios. NCDCs provide scalable storage and computing resources for public and non-profit service requirements based on non-profit modes of operation. These scalable resources can be dynamically organized as virtual machines (VMs) to run data-intensive services / applications. The required resources of a VM are sliced from a physical server in the NCDC. A physical server (called "server," for short) may contain one or more VMs. When a NCDC needs to create a large number of VMs to satisfy data-intensive service requirements, a primary concern is the VM placement problem [3]. Furthermore, with an increasing amount of large-scale, international collaborative research incorporating data-intensive services for its purposes, energy consumption has become a crucial factor for the VM placement problem in NCDCs.

With the increasing number and size of physical servers in data centers, energy consumption imposes a significant operational cost. Currently, datacenters that power Internet-scale applications consume approximately 1.3% of the worldwide electricity supply, and this fraction is expected to grow to 8% by 2020 [4,5]. Datacenter carbon emissions were 0.6% of the global total, nearly equal to that of the Netherlands, and the fraction is expected to reach 2.6%, which exceeds the carbon emission of Germany by 2020 [5]. Hence, in order to make good use of NCDCs, energy conservation and carbon emission reduction form a major part of the strategies of governments the world over. Building a NCDC management mechanism for energy conservation and emissions reduction

- *S. Wang, A. Zhou, and F. Yang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. E-mail: {sgwang; aozhou; fcyang@bupt.edu.cn}.*
- *C. Hsu is with the Department of Computer Science and Information Engineering, Chung Hua University, Hsinch, Taiwan. E-mail: chh@chu.edu.tw*
- *X. Xiao is with the School of Electronic Engineering, Beijing University of Posts and Telecommunications. E-mail: ptfairyxxy@gmail.com*

[1] *http://galileo.rice.edu/*
[2] *http://www.atlas.ch/*
[3] *www.cgms-info.org/*

has thus become a vital task.

To reduce energy consumption, server consolidation technology using virtualization is introduced in data centers. This technology can consolidate multiple applications on the same physical server, with each application typically running on its own virtual machines (VMs) [6]. In return, these VMs are mapped to physical servers. In the context of virtualized data centers (e.g., cloud data centers), it is a critical concern to design energy-efficient VM placement approaches that reduce energy consumption while satisfying quality of services (QoS) (e.g., response time, reliability, and throughput) of services / applications. Then, using the above technology as a foundation, some notable work [7-9] has been devoted to reduce the energy consumption of cloud datacenters. Although these energy-aware VM placement approaches can significantly achieve energy conservation and emissions reduction of cloud datacenters, they are low efficient for NCDCs by some of the following factors:

- Most current studies have assumed that the physical servers of a cloud data center are homogeneous. While this assumption appears reasonable at first blush, it is unreasonable in environments involving NCDCs because a variety of new servers are typically added to a NCDC to run new data-intensive services or satisfy novel demands in processing NCDC operations. Thus, they form a heterogeneous cloud data center environment. In a NCDC, server configurations often differ (the hardware configuration of NCDCs can differ significantly in terms of the CPU core count, memory, hard disk, and other components), which leads to varying server energy consumption characteristics. This implies that the minimum number of active servers may not consume the least amount of energy. Thus, approaches to energy conservation focusing on operating a minimum number of servers, which constitute most research on the issue, may not be able to achieve the best energy-saving effect. Therefore, these approaches are not applicable to NCDC environments involving a large number of heterogeneous servers.
- A data-intensive service consists of functional as well as non-functional attributes (e.g., QoS). The functional attributes of services are typically fixed through workflow management, but the QoS often changes because of dynamic network environments. Although the arrival rate of data-intensive service requirements is stable (in contrast to cloud data centers) due to the absence of commercial operations, the QoS of NCDC services often fluctuates. Hence, excellent data-intensive service provision systems not only satisfy the functional attribute requirements of the service, but also guarantee its QoS. Hence, QoS also plays an important role in determining the success or failure of service provision. However, traditional energy-aware VM placement schemes mainly focus on energy conservation to minimize the number of servers, and rarely consider the global QoS guarantee of data-intensive services. For example, some data-intensive services demand quick response times whereas others need high throughput. The global QoS guarantee thus relies on the aggregation of QoS requirements of all data-intensive service. From the perspective of NCDCs, if the VMs cannot provide a computing environments that satisfies the global QoS guarantee in order to fulfill a given service level agreement (SLA), this poses a serious hindrance to international collaboration in scientific research. In such a scenario, the relevant NCDC will not pay penalties for violating the relevant SLA, and the SLA violation of one or more data-intensive services is tolerable when satisfying the global QoS guarantee of all data-intensive services. Hence, finding the best VM placement solution with a global QoS guarantee becomes a crucial issue for NCDCs.

In this paper, based on our previous work [17], we propose an energy- and QoS-aware VM placement approach that NCDCs can use to support data-intensive services for international collaborative scientific research. Our contributions in this paper are as follows:

1) In contrast to past research in the area, our study eliminates the assumption of server homogeneity, adds a global QoS guarantee, and considers the VM placement optimization problem as a tradeoff between energy consumption and global QoS guarantee in NCDCs.
2) We present an energy-and QoS-aware VM placement optimization approach based on particle swarm optimization (PSO). To effectively solve the VM placement optimization problem, we improve PSO by redefining its parameters and operators. We then propose a local fitness-first strategy to update particle position. Moreover, we design a novel two-dimensional (2D) particle encoding scheme. Finally, we use the improved PSO to find the optimal virtual machine placement.
3) To evaluate our approach, we extend CloudSim[4], a well-known cloud simulator, to a new simulator called FTCloudSim by adding fat-tree data center network construction module, a QoS module,

[4]http://www.cloudbus.org/cloudsim/

and so on. We implement all approaches in FTCloudSim, and compare our approach with others in terms of energy consumption and global QoS guarantee. Experimental results show that our proposed approach can reduce energy consumption while still satisfying the global QoS guarantee.

The remainder of this paper is organized as follows: Section 2 introduces related work in the area. In Section 3, we develop our energy consumption model and design QoS utility functions. Our proposed energy- and QoS-aware virtual machine placement optimization method is detailed in Section 4. Experiments to compare our proposal against prevalent methods are described in Section 5. We offer our conclusions as well as an outlook on future work in the area in Section 6.

2. Related Work

A number of schemes have been proposed for efficient VM placement in cloud data centers.

From the perspective of energy-aware VM placement, C. Tang et al. [7] investigated the application workload placement optimization problem in the context of an enterprise data center. They presented an online application placement approach to minimize the number of application starts and stops. They maximized the total satisfied application demand, and balanced the load across machines. V. Petrucci et al. [8] modeled the energy optimization of cloud data centers as a mixed integer programming problem, and obtained an exact solution using the CPLEX solver optimization software package. D. Kusic et al. [9] modeled the energy consumption optimization of a virtualized data center as a sequential optimization problem, and proposed an energy optimization algorithm based on control theory. K. Le et al. [10] studied the possibility of lowering electricity costs for cloud providers operating multiple geographically distributed data centers, and designed policies that intelligently place and migrate load across the data centers to take advantage of time-based differences in electricity prices and temperatures. L. Wang et al. [11] developed scheduling heuristics to reduce energy consumption of a tasks execution and discusses the relationship between energy consumption and task execution time by increasing task execution time within an affordable limit. X. Jing and J. A. B. Fortes [12] modeled the VM placement problem as a multi-objective optimization problem of simultaneously minimizing total resource wastage, power consumption, and thermal dissipation costs, and used an improved genetic algorithm with fuzzy multi-objective evaluation to search through a large solution space. Although these energy-aware virtual machine placement approaches can significantly reduce the energy consumption of cloud data centers, they are inefficient in NCDCs because the physical servers are heterogeneous. Moreover, these schemes cannot provide the QoS guarantee for VMs, and distort the global QoS guarantee of data-intensive services in NCDCs.

From the perspective of energy- and QoS-aware VM placement, W. Shao-Heng et al. [13] investigated a method to integrate QoS awareness with energy saving in VM placement, i.e., in addition to fully exploiting the resources of servers, they considered the QoS requirements of user applications. This scheme combined three key techniques: (1) hop reduction, (2) energy saving, and (3) load balancing. Hop reduction was used to regroup VMs to lower the traffic load among them. Energy saving technique was adopted to choose the appropriate servers. The proposed load balancing was employed to periodically update VM placement. Goudarzi and Pedram [14] generated multiple copies of VMs without sacrificing QoS, proposed an algorithm based on dynamic programming and local search to determine the number of VM copies, and placed them on servers to minimize the total energy cost in cloud computing systems. Beloglazov and Buyya [15,16] defined the problem of minimizing energy consumption while meeting QoS requirements, stated requirements for VM allocation policies, and found the best solution in three stages: reallocation according to the utilization of multiple system resources at any given time, optimization of virtual network topologies established between VMs, and VM reallocation considering the thermal states of the resources. Their proposed energy-efficient resource allocation policies and scheduling algorithms consider QoS expectations and the power usage characteristics of devices. Although these energy- and QoS-aware VM placement approaches can reduce the energy consumption of cloud data centers with QoS guarantee, the QoS guarantees of these schemes are only local guarantees, and they cannot satisfy the global QoS guarantee of data-intensive services in NCDCs. Moreover, the QoS guarantee of these schemes involves avoiding SLA violations. However, For NCDCs, SLA violations of one or more data-intensive services is tolerable when satisfying the global QoS guarantee of all data-intensive services.

In contrast to existing schemes, which exhibit poor performance due to heterogeneous physical servers and the global QoS guarantee in NCDCs, our approach can minimize energy consumption while satisfying the global QoS guarantee by employing an improved PSO. Our approach can find the best VM placement scheme because it does not rely on enumerating all possible combinations of physical servers. Moreover, it can satisfy the global QoS guarantee by maximizing the overall QoS utility function, whereas existing schemes cannot support this crucial case.

3. Energy and QoS Computation

Note that the notations in Table I will be used throughout the paper.

TABLE I. NOTATIONS

Symbol	Meaning
ps_i	the *i-th* physical machine in the data center, *i*= 1, 2, …
vm_i	the *j-th* virtual machine in the data center, *j*= 1, 2, …
w_k	the weight of the *k*-th QoS attribute
S	a service
$Q_{j,k}^{max}$	the maximum value of the attribute in the *j*-th server
$Q_{j,k}^{min}$	the minimum value of the attribute in the *j*-th server
$q_j(S)$	the *j*-th attribute value in service
r_i^{cpu}	the maximum CPU and memory requirements of the *i*-th virtual machine
r_i^{mem}	the maximum CPU and memory requirements of the *i*-th virtual machine
c_j^{cpu}	the CPU and memory resource capacities of the *j*-th server
c_j^{mem}	the CPU and memory resource capacities of the *j*-th server
$u_{ij}(t)$	the CPU utilization of the i-th VM running on the *j*-th server
X_i^t	Anbit vector that represents a feasible VM placement solution
$u(t)$	the varying CPU utilization
$P(u(t))$	the energy consumption of the server at time *t*
P_{max}	the maximum energy consumed by a server that is fully utilized
f	local energy fitness
En	the overall energy consumption of the server in a period

3.1 Energy Consumption Model

It is well-known that the energy consumption of servers relies on the comprehensive utilization of a CPU, memory, disk, and network card. Of these factors, the CPU is the most important energy consumption component. Hence, the CPU utilization of a server usually represents its resource utilization [17,18]. CPU utilization can be modeled as a function of time according to workload variability, and the energy model of the server can then be established based on CPU utilization. Based on past work [16,18-20], we introduce an energy consumption model of a server in NCDCs as follows:

$$En = \int_{t_1}^{t_2} P(u(t)) \cdot dt, \quad (1)$$

with

$$P(u(t)) = c * P_{max} + (1-c) * P_{max} * u(t),$$

where En is the overall energy consumption of the server in the period $[t_1, t_2]$, $u(t)$ ($u(t) \in [0,1]$) is the varying CPU utilization, the CPU utilization can be obtained by monitoring the server, $P(u(t))$ is the energy consumption of the server at time *t*, P_{max} is the maximum energy consumed by a server that is fully utilized, and c is the fraction of energy consumed by the server when idle.

3.2 QoS Utility Function

It is well-known that the QoS requirement of a data-intensive service (called service, for short) contains many attributes, such as response time, reliability, throughput, delay, availability, and so on. In general, these attributes can be divided into two categories: positive and negative QoS attributes. Positive QoS attributes (e.g., reliability, availability, etc.) imply that the larger the attribute value, the better the performance of the server running relevant service. Conversely, negative QoS attributes (e.g., response time, delay, etc.) ought to be as low in value as possible. In this paper, we only consider negative QoS attributes (positive attribute values can be easily converted into negative attribute values, i.e., by multiplying by -1).

Consider a QoS requirement for service s with *r* attributes with attribute vector $qs = \{q_1(s), q_2(s), ..., q_r(s)\}$, where the value of $q_k(s)$ $(1 \le k \le r)$ represents the *k-th* attribute value in service s. Similarly, the attribute vectors of all *l* services can be expressed as $QS = \{q_1(S), q_2(S), ..., q_r(S)\}$ ($S = \{s_1, \cdots s_m\}$), where the value of $q_k(S)$ is aggregated by the k-th attribute values from all services, as shown in Table II. Table II lists the QoS aggregation functions of services.

TABLE II. QoS Aggregation Functions

QoS Attributes	Functions
Response time	$q(S) = \max_{i=1}^{l} q(s_i)$
Throughput	$q(S) = \sum_{i=1}^{l} q(s_i)$
Availability, Reliability	$q(S) = \min_{i=1}^{l} q(s_i)$

Each service involves multiple QoS attributes leading to different units or scope, which is not helpful in satisfying the global QoS guarantee. Therefore, we need to design a QoS utility function to map the vector of QoS values qs into a single real value. Moreover, in this paper, we consider a NCDC composed of n servers $PS = \{ps_1, ps_2, ..., ps_n\}$ hosting a set of m VMs $VM = \{vm_1, vm_2, ..., vm_m\}$. A service is often implemented as a VM deployed to a server while satisfying its specified resource (i.e., CPU and memory) and QoS constraints. Each VM runs one service as a time-varying workload ($m = l$). A service runs only on a VM. The QoS of the service is then usually associated with VM provision. Hence, our QoS utility function scales all attribute values to the domain [0, 1] for uniform computation on multi-dimensional QoS attributes depending on the servers, as shown in Definition 1.

Definition 1 (QoS Utility Function): Suppose there are r QoS attributions. The QoS utility functions for the *i-th* service $s_i \in S$ $(1 \le i \le l)$ running the VM of the *j-th* server

ps_j ($(1 \le j \le n)$) and all services S are defined as follows:

$$U(s_i)=\sum_{k=1}^{r}\frac{Q_{j,k}^{max}-q_k(s_i)}{Q_{j,k}^{max}-Q_{j,k}^{min}}.\omega_k, \tag{2}$$

$$U(S)=\sum_{k=1}^{r}\frac{Q_k^{max}-q_k(S)}{Q_k^{max}-Q_k^{min}}.\omega_k, \tag{3}$$

with

$$\begin{cases} Q_k^{max}=\sum_{k=1}^{r}Q_{j,k}^{max}\,(Q_{j,k}^{max}=\max\limits_{\forall s_i\in ps_j} q_k(s_i)) \\ Q_k^{\min}=\sum_{k=1}^{r}Q_{j,k}^{\min}\,(Q_{j,k}^{\min}=\min\limits_{\forall s_i\in ps_j} q_k(s_i)) \end{cases}, \tag{4}$$

where $w_k \in R^+(\sum_{k=1}^{r} w_k = 1)$ represents the weight of each QoS attribute, the users can adjust the weights based on their own needs, $Q_{j,k}^{max}$ is the maximum value of the k-th attribute in the *j-th* server and $Q_{j,k}^{min}$ is its minimum value, Q_k^{max} is the summation of each $Q_{j,k}^{max}$ in all servers and, similarly, Q_k^{min} is the summation of each $Q_{j,k}^{min}$.

3.3 Energy- and QoS-aware VM Placement Model

The optimization objective of VM placement is to minimize total energy consumption while satisfying the global QoS guarantee. If the requested maximum resources of the virtual machine are allocated, the cloud service can run satisfactorily on this virtual machine [19]. By rewriting Eqs. (1) and (2), energy- and QoS-aware VM placement in a NCDC can be formulated as a multi-objective constraint optimization problem, i.e., a minimization problem of the overall energy consumption with a maximization problem of the overall QoS utility function (i.e., global QoS guarantee), given by

$$\text{Min}\sum_{j=1}^{n}\sum_{i=1}^{m}E_j x_{ij}, \tag{5}$$

$$\text{Max}\sum_{k=1}^{r}\frac{Q_k^{\max}-\sum_{j=1}^{n}\sum_{i=1}^{m}x_{ij}\cdot q_k(s_i)}{Q_k^{\max}-Q_k^{\min}}.w_k \tag{6}$$

subject to the QoS constraints and resource capacities satisfying the allocation constraints on the decision, as

$$\sum_{j=1}^{n}\sum_{i=1}^{m}q_k(s_i)\cdot x_{ij}\le C_k, 1\le k\le r, \tag{7}$$

$$\sum_{i=1}^{m}r_i^{cpu}x_{ij}<c_j^{cpu}, \tag{8}$$

$$\sum_{i=1}^{m}r_i^{mem}x_{ij}<c_j^{mem}, \tag{9}$$

$$\sum_{j=1}^{n}x_{ij}=1, i=1,2,...,m, \tag{10}$$

where n is the number of servers in the NCDC, m is the number of virtual machines, E_j is the total energy consumption of the *j-th* server, r_i^{cpu} and r_i^{mem} are the maximum CPU and memory requirements of the *i-th* virtual machine, respectively, and c_j^{cpu} and c_j^{mem} are the CPU and memory resource capacities of the *j-th* server, respectively. C_k is the QoS constraint value, and $C_k \ge q_k(S)$.

Eq. (7) states that the QoS aggregation value of all services must be less than the constraint value, Eqs. (8) and (9) state that the sum of the resource requirements of VMs must be less than the relevant server's resource capacity, and Eq. (10) shows that a VM can only be placed on one server such that x_{ij} =1 if the *i-th* VM is run on the *j-th* server, and x_{ij} =0 otherwise. Note that due to the heterogeneity of NCDCs, the c_j^{cpu} of the *j-th* server is not equal to the c_k^{cpu} of the *k-th* server, and the c_j^{mem} of the *j-th* server is not equal to the c_k^{mem} of the *k-th* server.

From Eqs. (5-10), we find that the energy- and QoS-aware VM placement is an NP-hard problem. The problem of finding the best VM placement is considered an optimization problem where the overall energy consumption must be minimized and the overall QoS utility value must be maximized while satisfying all constraints (Eqs. (7-10)). We thus propose an energy- and QoS-aware VM placement approach based on an improved PSO to solve the optimization problem to find the best VM placement operator with a tradeoff between energy consumption and global QoS guarantee.

4 Proposed VM Placement Approach

PSO [21] is a random search algorithm based on swarm intelligence. It shares many similarities with evolutionary computation techniques, and is easy to implement as there are few parameters to adjust. Moreover, compared with similar optimization algorithms, PSO algorithms have such advantages as faster execution and higher efficiency of problem solving [22,23]. At present, PSO has been successfully applied to many areas, such as function optimization, artificial neural network training, and fuzzy systems control. PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Thus, we attempt to use it to solve the energy- and QoS-aware VM placement optimization problem.

4.1 Particle Swarm Optimization

Each member of the swarm in PSO is called a particle, and represents a feasible solution of the search problem in question. Each particle has two parameters: velocity and position. The position of each particle is associated with a fitness value, which is often used to evaluate the quality of the solution. PSO begins by initializing a group of random particles, and iteratively finds the optimal solution. It imitates the interactive behavior of a foraging flock of birds. Each particle flies in the multi-dimension search space at a specified velocity while referring to the

best local position $X_{lbest,i}$ and the best global position X_g, and updates its velocity and position to move the swarm toward the best solutions as follows:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (X_{lbest,i}(t) - X_i^t) + c_2 r_2 (X_{gbest}(t) - X_i^t), \quad (11)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1}, \quad (12)$$

where V_i^t and V_i^{t+1} are the velocity before the update and the updated velocity, respectively, and X_i^t and X_i^{t+1} are the position before the update and the updated position, respectively. ω is called the inertial weight coefficient, and balances the local and global search capabilities of particles, and linearly decreases from 0.9 to 0.4 through the search process. c_1 and c_2 are positive constants, called cognitive learning factors, which enable the particle to learn, and r_1 and r_2 are random functions in the range [0,1].

To be applied to the energy- and QoS-aware VM placement problem, PSO must be improved as follows: 1) a traditional PSO is suitable only for solving a continuous optimization problem, and is unsuited to solving discrete optimization problems [20], which means that the parameters and operators of PSO must be redefined. 2) To apply PSO to solve the problem at hand, a new position update strategy and coding scheme must be designed. Thus, in this paper, we adopt the improved PSO as the key to solving the energy- and QoS-aware VM placement optimization problem.

4.2 Improved PSO

Based on our past work [20,24], our improvement for PSO focuses on 1) redefining the parameters and operators of PSO to solve the discrete optimization problem, i.e., the energy- and QoS-aware VM placement optimization problem, and 2) adopting a local fitness-first strategy to update particle position.

4.2.1 Redefining PSO

Traditional PSO is suitable only for solving continuous optimization problems, and fails to solve the energy- and QoS-aware VM placement optimization. Thus, we redefine the parameters and operators of PSO to solve a discrete optimization problem. Combined with the specific characteristics of the energy- and QoS-aware VM placement optimization problem, the parameters and operators of the PSO can be redefined by the following definitions.

Definition 2 (Particle Position). Particle position $X_i^t = (x_{i1}^t, x_{i2}^t, \ldots, x_{in}^t)$ is redefined as an n-bit vector that represents a feasible VM placement solution, where n is the length of the particle code, and is equal to the number of servers in a NCDC.

Definition 3 (Particle Velocity). Particle velocity $V_i^t = (v_{i1}^t, v_{i2}^t, \ldots, v_{in}^t)$ is redefined as an n-bit vector, and represents the adjustment decision of VM placement. V_i^t guides the particle position update operation to drive VM placement to adjust for the optimal solution. The value of every bit in the vector V_i^t is 0 or 1. It is 0 if the corresponding server and its VM are re-evaluated and adjusted, and is 1 otherwise.

Definition 4 (Subtraction Operator). We call Θ the subtraction operator redefined to calculate the difference between two VM placement solutions. As far as $X_i^t \Theta X_k^t$ is concerned, if the value of the corresponding bit of the solution X_i^t is equal to that of solution X_k^t, the value of the corresponding bit in the result is 1; otherwise, the value is 0. For example, (1, 1, 1) Θ (1, 0, 0)= (1, 0, 0).

Definition 5 (Addition Operator). We use $\oplus$ to represent an addition operator redefined to represent the particle velocity update operation because of its own inertial velocity, best position, and global best position during particle updates. Then, $P_1 V_1^t \oplus P_2 V_2^t \oplus \cdots P_n V_n^t$ states that a particle updates its velocity by using V_1^t with probability P_1, …, and V_n^t with probability P_n. We call the probability P_i ($\sum_{i=1}^{n} P_i = 1$) the inertial weight coefficient. For example, 0.3(0, 0, 1, 1) $\oplus$ 0.7(0, 1, 0, 1) = (0, #,#, 1). The probability that the value of the second bit is 0 is 0.3, and the probability that the value is 1 is 0.7. Here, the value of # is uncertain, and the bit value is called an uncertain bit value. The uncertain bit value affects the update of particle velocity. In the improved PSO, there are three inertial weight coefficients, P_{1i}, P_{2i}, P_{3i}, which are random functions in the range [0,1].

Definition 6 (Multiplication Operator). We call $\otimes$ the multiplication operator redefined to update particle position. $X_i^t \otimes V_k^{t+1}$ represents the position update operation of particle position vector X_i^t at any given time based on velocity vector V_k^{t+1}. The computation rule of $\otimes$ is as follows: 1) if the bit value of the velocity vector is 1, the corresponding bit of the position vector is not adjusted; 2) if the bit value of the velocity vector is 0, it is adjusted. For example, (1, 0, 1, 0) $\otimes$ (1, 1, 0,0), where (1, 0, 1, 0) is the position vector and (1, 1, 0,0) is the velocity vector. The third and fourth bit values of the velocity vector are all 0, which indicates that the status of the third and fourth server in the corresponding virtual machine placement solution should be updated.

Finally, based on the above five definitions, we improve PSO by transforming Eqs. (11-12) to Eqs. (17-18), as follows:

$$V_i^{t+1} = p_{1i} V_i^t \oplus p_{2i} (X_{lbest,i}(t) \Theta X_i^t) \oplus p_{3i} (X_{gbest}(t) \Theta X_i^t), \quad (17)$$

$$X_i^{t+1} = X_i^t \otimes V_i^{t+1}. \quad (18)$$

4.2.2 Local Fitness-first Strategy

Particle position update usually adopts a random selection strategy. However, the random selection strategy affects the overall convergence of PSO, which reduces the effectiveness of our approach. Hence, to enhance the quality of the solution, we propose a local fitness-first strategy to update particle position.

For ease of presentation, every bit in the first dimension of the particle is called the local position. The CPU utilization of all VMs running on this server in an optimization period $[t_1, t_2]$ is called local energy fitness, which is expressed as follows:

$$f^{e}_{lbest,i} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} (\sum_{j=1}^{m} u_{ij}(t)) \cdot dt, \quad (19)$$

where $u_{ij}(t)$ is the CPU utilization of the *i-th* VM running on the *j-th* server, and m is the total number of virtual machines running on the *j-th* server .

The QoS aggregation of all VMs running on the server is called local QoS fitness, and is represented as follows:

$$f^{q}_{lbest,i} = \sum_{k=1}^{r} \frac{Q^{max}_{j,k} - q_k(s_i)}{Q^{max}_{j,k} - Q^{min}_{j,k}}, \quad (20)$$

where $q_k(s_i)$ is the *k-th* QoS attribute value of the *i-th* VM running on the *j-th* server, r is the total number of QoS attributes, $Q^{max}_{j,k}$ is the maximum value of the k-th attribute in the *j*-th server, and $Q^{min}_{j,k}$ is its minimum value.

Based on Eqs. (19) and (20), local fitness can be determined by the following:

$$f_{lbest,i} = f^{e}_{lbest,i} + f^{q}_{lbest,i} / r, \quad (21)$$

For the local fitness-first strategy, when PSO needs to update a certain local position, the VM on the server with the maximum fitness is selected to fill the local position with a larger probability. Local fitness represents the CPU utilization and QoS aggregation of the server, and these are related to the energy consumption of the server and the QoS guarantee of the service running on the VM.

4.2.3Encoding Scheme

To improve the efficiency of the solutions, as shown in Fig. 1, we devise a 2D encoding scheme based on the character (a one-to-many mapping relationship between the server and the VM) of the energy-aware VM placement optimization problem. Where *n* denotes the number of servers, and *m* denotes the number of virtual machines placed on the same server.

As shown in Fig. 1, the First Dimension of a particle is a *n*-bit binary vector. Every bit in the vector is associated with a server in a NCDC. Here, "1" denotes that the corresponding server is active in the current VM placement solution, and "0" denotes otherwise. The Second Dimension of a particle is a set of subsets that comprises the VMs to be placed. Then each VM subset is associated with an active server. For example, the first bit value of the First Dimension of this particle is equal to 1, which means that the first server in the NCDC should be turned on. The first, second VM should be placed onto the first server. Compared with traditional one-dimension particle encoding, our designed two-dimension encoding scheme not only can effectively shorten the particle encoding length to reduce the search time but also can reflect the character of the VM static placement optimization problem. Hence, the encoding scheme is conducive to maintaining the current feasible solution and improving the convergence speed of the PSO.

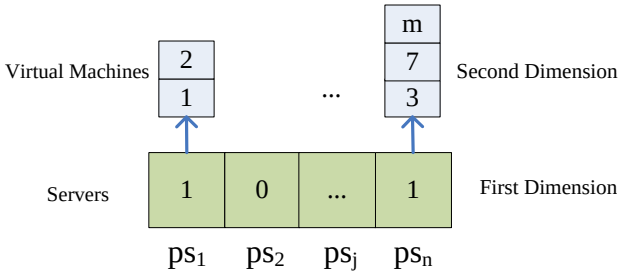

Fig. 1. Two-dimensional encoding scheme.

5 Simulation Evaluation

To evaluate the performance of our proposed approach, we compared it with other approaches in terms of energy consumption and the global QoS guarantee. Moreover, we studied some parameters of our proposed approach.

5.1Simulation Setup

Since the target system is a NCDC environment, it is extremely difficult to conduct repeatable, large-scale experiments on a real national infrastructure, which is required to evaluate and compare the proposed energy- and QoS-aware VM placement approach with other methods. Hence, to ensure the repeatability of experiments, we chose simulations as an alternative to evaluate the performance of our approach.

CloudSim [25] is a modern simulation framework aimed at cloud computing environments, and supports the modeling of virtual resource allocation, energy consumption, service scheduling, and other functions. Our past work [26,27] extended CloudSim as FTCloudSim[5] by adding some modules to support more extensive experiments, such as fat-tree data center network construction, failure and repair event triggering, checkpoint image generation and storage, checkpoint-based service recovery, and so on. Apart from the above modules, the ability to simulate service applications with QoS guarantee was incorporated to support our experiments here by adding a QoS module to FTCloudSim.

[5] http://youtu.be/yMyz2gesywA

We simulated a NCDC of 1,000 heterogeneous physical servers. To reflect the heterogeneity of the NCDC, these servers were divided into two categories, i.e., HP ProLiant G4 with CPU (3720MIPS), memory (4GB), and peak energy consumption (117 Watts), and HP ProLiant G5 with CPU (5320 MIPS), memory (4GB), and peak energy consumption (135Watts). The servers had different configurations and energy consumption characteristics [28]. Each physical server runs an one or more data-intensive services/applications with four QoS attributes (i.e., Response Time, Availability, Throughput, Reliability) generated by 2,500 real Web services [29-31], i.e., the QWS dataset[6], where the response time represented a QoS constraint attribute.

Moreover, to better reflect actual VM requests, we simulated two types of resource request parameters of Amazon EC2 instances, i.e., Micro Instance with CPU (500MIPS) and memory (613MB), and Small Instance with CPU (1000MIPS) and memory (1700MB).

We compared this approach with the modified best fit decreasing (MBFD) approach proposed in [16], the first-fit algorithm (FF), and the best-fit algorithm (BF). FF and BF are straightforward greedy approximation algorithms. With First Fit, the servers are indexed in increasing order of remaining capacity. Each virtual machine is sequentially placed on the lowest indexed server onto which it will fit. With the Best Fit algorithm, each virtual machine is placed onto the server with smallest energy consumption that can host it. All experiments were conducted on the same computer running FTCloudSim. A sufficient number of repetition tests were executed to set the following parameters: the parameter of the server energy model c was set to 0.6, the initial population of the PSO was set to 20, and the maximum number of iterations was set to 30. Each experiment was run 10 times.

5.2Comparison of Energy Consumption

In this paper, we consider the energy consumption as the total energy consumption of all active servers. As shown in Fig. 2, we provided the comparison results.

Fig. 2 shows that our proposed approach enabled the data center operators to save more energy than other approaches, regardless of the number of virtual machine requests. Compared with the other two approaches, our approach saved approximately 35% more on energy. This is because the FF, BF, and MBFD lack global information (i.e., the energy consumption characteristics of heterogeneous servers in a NCDC), only account for multi-dimensional resource constraints, and do not consider the energy difference among different servers in the problem-solving process. However, our approach introduces an effective particle velocity and position update mechanism, which enables it to find a better virtual machine placement solution and enhances the convergence of the algorithm, thus improving the quality of the solution. As a result, our approach activates the smallest number of servers possible, and reduces the total energy consumption in a NCDC.

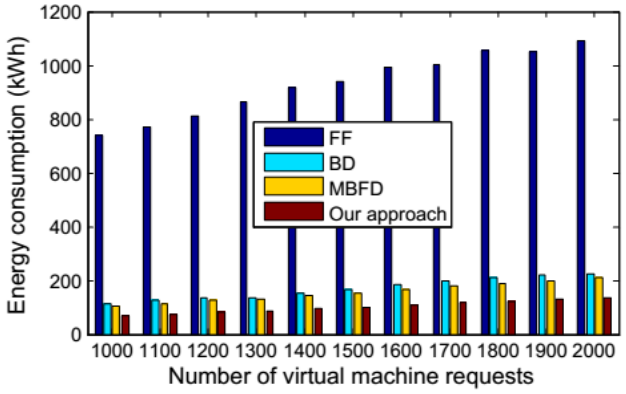

Fig. 2. Total energy consumption in terms of the number of virtual machine requests. Compared with other approaches, our approach saved approximately 35% more on energy.

5.3 Comparison of Global QoS Guarantee

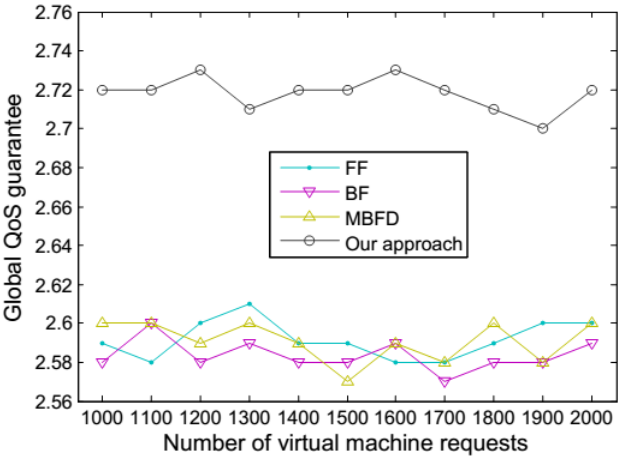

Fig. 3. Global QoS guarantee with respect to the number of virtual machine requests. Compared with other approaches, our approach significantly satisfied the global QoS guarantee for data-intensive services in a NCDC.

In this experiment, we evaluated the global QoS guarantee obtained by comparison with the results of all approaches. The number of virtual machine requests was in the range [1000, 2000] with four QoS attribute requests. Because the four QoS attributes had different units or scopes, we designed a QoS utility function to map the vector of QoS values into a single real value. The QoS utility function scaled all attributes values to the domain [0, 1] for uniform computations on multi-dimensional QoS attributes depending on the servers, as shown in Definition 1. Hence, the global QoS guarantee ranged from 0 to 4.

[6]http://www.uoguelph.ca/~qmahmoud/qws/

Fig. 3 shows the results of a comparison of the global QoS guarantee. The global QoS guarantee of our approach was 2.72 on average, higher than those of other approaches. Our approach thus significantly satisfied the global QoS guarantee for data-intensive services in a NCDC. This is because other approaches focused on local QoS optimality. However, local QoS optimality cannot satisfy the global QoS guarantee of all data-intensive services. Hence, our approach exhibited outstanding performance (lowest energy consumption and highest global QoS guarantee) for data-intensive services in a NCDC.

5.4 Study of Parameters

In this section, we study the effect of the parameters of our proposed approach on energy consumption, global QoS guarantee, and computation time. As shown in Figs. 4-7, the parameters were the server energy parameter *c*, the number of virtual machines, the number of QoS constraints, and the weight of QoS *w.* In our experiments, the number of QoS attributes was four, and the number of virtual machines was 1,000. The number of heterogeneous physical servers was 1,500.

5.4.1 Effect of the Server Energy Parameter c

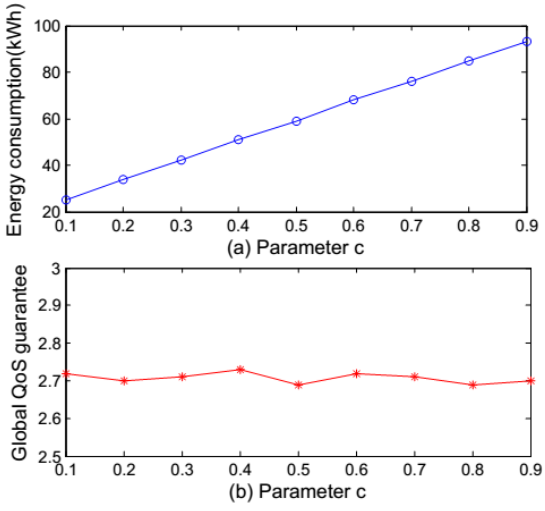

Fig. 4. Effect of parameter *c*. The parameter represents the fraction of energy consumed by the server when idle. The lower the energy consumption of our approach, the lower the parameter *c*. The global QoS guarantee of our approach was not substantially affected by *c*.

Fig. 4 (a) and (b) show the effect of *c* on our virtual machine placement approach. To clearly show its impact, we varied the value of *c* from 0.1 to 0.9 with a step value of 0.1. The number of QoS constraints was 1, and we set w=0.8 in the experiment. The figure shows the following: (1) energy consumption significantly increased when the value of *c* increased from 0.1 to 0.9. This observation indicates that the lower the energy consumption of our approach, the lower the value of *c*, i.e., the more idle the server; (2) the global QoS guarantee was not substantially influenced by the value of the parameter *c*.

5.4.2 Effect of the Number of QoS Constraints

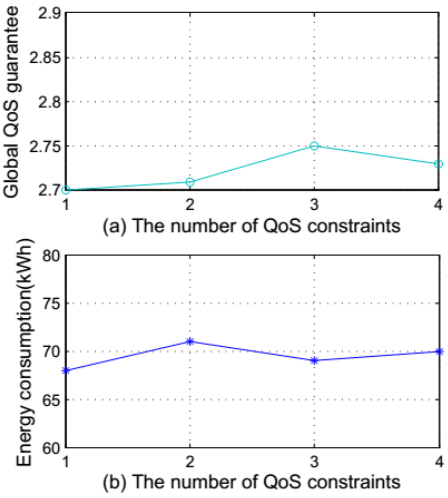

Fig. 5. Effect of the number of QoS constraints. The number of QoS constraints represents users' QoS requirements for data-intensive services in a NCDC. The global QoS guarantee and energy consumption of our approach were not substantially affected by this parameter.

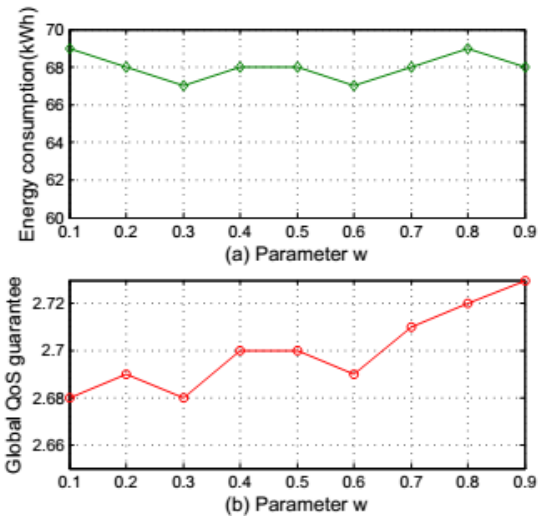

Fig. 6. Effect of parameter *w*. *w* represents the weight of each QoS attribute. The energy consumption of our approach was not substantially affected by *w*. The global QoS guarantee significantly increased when the value of *w* increased from 0.6 to 0.9.

Fig. 5 (a) and (b) show the effect of the number of QoS constraints on our virtual machine placement approach. To clearly show its impact, we varied the number of QoS constraints from one to four with a step value of 1. We set *c*=0.6 in the experiment. The weight of response time is set as 0.8. The weights of other attributes are randomly generated between 0 and 0.2. The sum of the weights is1. The figure shows that the global QoS guarantee and energy consumption of our approach were not substantially affected by the number of QoS constraints.

5.4.3 Effect of the parameter w

Fig. 6 (a) and (b) show the effect of parameter *w* on our virtual machine placement approach. To clearly show its impact, we varied the value of *w* from 0.1 to 0.9 with a step value of 0.1. The number of QoS constraints was one, and we set *c*=0.6 in the experiment. The figure shows the following: (1) the global QoS guarantee significantly increased when the value of *w* increased from 0.6 to 0.9. This observation indicates that the better the global QoS guarantee of our approach, the higher the value of parameter *w*; (2) energy consumption was not substantially influenced by the value of *w*; (3) our approach exhibited its best performance for values of *w*in the interval [0.7, 0.9].

5.4.4 Effect of the Number of Virtual Machines

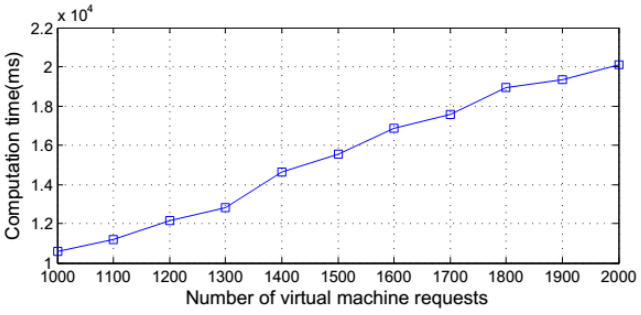

Fig. 7. Computation time of our proposed placement approach. The figure shows that the computation time of our approach was an approximately linear relationship between the numbers of virtual machine requests and computation time. The computation time of our approach was very short.

Fig. 7 shows the effect of the number virtual machine requests on our virtual machine placement approach. To show its impact clearly, we varied the number of virtual machine requests from 1,000 to 2,000 with a step value of 100. The number of QoS constraints was one, and we set *c*=0.6 and *w*=0.8 in the experiment. The figure shows that the computation time increased slowly when the number of virtual machine requests increased. This observation indicates that the computation time of our proposed placement approach was an approximately linear relationship between the number of virtual machine requests and the time cost. This means that our placement approach attained satisfactory scalability with an approximate optimal solution of the virtual machine placement problem. Fig. 7 shows that the computation time increased slowly when the number of virtual machine requests increased. This observation indicates that the computation time of our proposed placement approach was an approximately linear relationship between the number of virtual machine requests and the time cost. This means that our placement approach attained satisfactory scalability with an approximate optimal solution of the virtual machine placement problem.

6 Conclusions

With an increasing amount of international, large-scale scientific research incorporating data-intensive services in their procedures, energy consumption with the global QoS guarantee becomes a crucial issue for VM placement in NCDCs. In contrast to past work in the area, we proposed in this paper an energy- and QoS-aware VM placement optimization approach by eliminating the assumption of server homogeneity, adding the global QoS guarantee, and considering the VM placement optimization problem with a tradeoff between energy consumption and the global QoS guarantee in NCDCs. To effectively solve the VM placement optimization problem, we improved PSO by redefining its parameters and operators, and adopted a local fitness-first strategy to update particle position. Following this, based on a novel 2D particle encoding scheme, we used the improved PSO to find the optimal virtual machine placement with a tradeoff between energy consumption and the global QoS guarantee. Experimental results showed that our proposed approach can reduce energy consumption while satisfying the global QoS guarantee.

Our proposed energy-aware virtual machine placement is in a tree-like datacenter network, which is adopted by the commercialized data center. When a datacenter adopts other topologies, the work cannot save too much energy. Moreover, the service that is hosted in the virtual machines is single service. When the virtual machines host a composited service, our placement approach is not suitable. Hence, our future work in the area will focus on solving the limitations of our approach, such as improving support for composite services hosted in virtual machines, and supporting other topologies in NCDCs.

ACKNOWLEDGMENTS

This work was supported by the NSFC (61202435 and 61472047), the Beijing Municipal Natural Science Foundation (4132048), and the Fundamental Research Funds for Central Universities (2014ZD01). Ching-Hsien Hsu is the corresponding author.

REFERENCES

[1] L. Wang, J. Tao, R. Ranjan, H. Marten, A. Streit, J. Chen, and D. Chen, "G-Hadoop: MapReduce across distributed data centers for data-intensive computing," Future Generation Computer Systems, 3, vol. 29, pp. 739-750, 2013.

[2] S. Weijing, W. Lizhe, R. Ranjan, J. Kolodziej, and C. Dan, "Towards Modeling Large-Scale Data Flows in a Multidatacen-

ter Computing System With Petri Net," IEEE Systems Journal, 2, vol. 9, pp. 416-426, 2015.

[3] Q. Liang, J. Zhang, Y.-h. Zhang, and J.-m. Liang, "The placement method of resources and applications based on request prediction in cloud data center," Information Sciences, 20, vol. 279, pp. 735-745, 2014.

[4] J. Koomey, "Growth in data center electricity use: 2005 to 2010," Oakland, CA, 2011.

[5] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," in Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'12), 2012, pp. 211-222.

[6] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in Proceedings of the 29th Conference on Information Communications (INFOCOM'10), 2010, pp. 1154-1162.

[7] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, "A scalable application placement controller for enterprise data centers," in Proceedings of the 16th International Conference on World Wide Web (WWW'07), 2007, pp. 331-340.

[8] V. Petrucci, E. V. Carrera, O. Loques, J. C. B. Leite, and D. Mosse, "Optimized management of power and performance for virtualized heterogeneous server clusters," in Proceedings of the 11th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid 11), 2011, pp. 23-32.

[9] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," Cluster Computing, 1, vol. 12, pp. 1-15, 2009.

[10] K. Le, R. Bianchini, J. Zhang, Y. Jaluria, J. Meng, and T. D. Nguyen, "Reducing electricity cost through virtual machine placement in high performance computing clouds," in Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage, and Analysis (HiPC'11), 2011, pp.1-12.

[11] L. Wang, S. U. Khan, D. Chen, J. Kołodziej, R. Ranjan, C.-Z. Xu, and A. Zomaya, "Energy-aware parallel task scheduling in a cluster," Future Generation Computer Systems, 7, vol. 29, pp. 1661-1670, 2013.

[12] X. Jing and J. A. B. Fortes, "Multi-objective Virtual machine placement in virtualized data center environments," in Proceedings of IEEE/ACM Int'l Conference on Green Computing and Communications (GreenCom'10), 2010, pp. 179-188.

[13] W. Shao-Heng, P. P. W. Huang, C. H. P. Wen, and W. Li-Chun, "EQVMP: Energy-efficient and QoS-aware virtual machine placement for software defined data center networks," in Proceedings of the International Conference on Information Networking (ICOIN'14), 2014, pp. 220-225.

[14] H. Goudarzi and M. Pedram, "Energy-efficient virtual machine replication and placement in a cloud computing system," in Proceedings of the IEEE Fifth International Conference on Cloud Computing (CLOUD'12), 2012, pp. 750-757.

[15] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," In Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid 10), 2010, pp. 826-831.

[16] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Future Generation Computer Systems, 5, vol. 28, pp. 755-768, 2012.

[17] G. Dasgupta, A. Sharma, A. Verma, A. Neogi, and R. Kothari, "Workload management for power efficiency in virtualized data centers," Communications of the ACM, 7, vol. 54, pp. 131-141, 2011.

[18] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy-aware server provisioning and load dispatching for connection-intensive internet services," in Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI'08), 2008, pp. 337-350.

[19] G. Khanna, K. Beaty, G. Kar, and A. Kochut, "Application performance management in virtualized server environments," in Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS'06), 2006, pp. 373-381.

[20] W. Shangguang, L. Zhipiao, Z. Zibin, S. Qibo, and Y. Fangchun, "Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers," in Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS'13), 2013, pp. 102-109.

[21] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of the 1995 IEEE International Conference on Neural Networks, 1995, pp. 1942-1948.

[22] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis, "Particle swarm optimization for integer programming," in Proceedings of the IEEE 2002 Congress on Evolutionary Computation (CEC'02), 2002, pp. 1582-1587.

[23] L. Wang, H. Geng, P. Liu, K. Lu, J. Kolodziej, R. Ranjan, and A. Y. Zomaya, "Particle Swarm Optimization based dictionary learning for remote sensing big data," Knowledge-Based Systems, vol. 79, pp. 43-50, 2015.

[24] S. G. Wang, Q. B. Sun, H. Zou, and F. C. Yang, "Web service selection based on adaptive decomposition of global QoS constraints in ubiquitous environment," Journal of Internet Technology, 5, vol. 12, pp. 757-768, 2011.

[25] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, 1, vol. 41, pp. 23-50, 2011.

[26] A. Zhou, S. Wang, Q. Sun, H. Zou, and F. Yang, "FTCloudSim: A simulation tool for cloud service reliability enhancement mechanisms," in Proceedings of ACM/IFIP/USENIX International Middleware Conference (Middleware'13), Demo and Poster Track, 2013, pp. 1-2.

[27] A. Zhou, S. Wang, Z. Zheng, C. Hsu, and M. Lyu, "On cloud service reliability enhancement with optimal resource usage," IEEE Transactions on Cloud Computing, 99, vol. pp. pp. 1-14, 2014.

[28] 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 and Computation: Practice and Experience, 13, vol. 24, pp. 1397-1420, 2012.

[29] E. Al-Masri and Q. H. Mahmoud, "Investigating Web services on the World Wide Web," in Proceedings of the 17th International Conference on World Wide Web(WWW'08), 2008, pp. 795-804.

[30] M. Alrifai, D. Skoutas, and T. Risse, "Selecting skyline services for QoS-based web service composition," in Proceedings of the 19th International Conference on World Wide Web (WWW'10), 2010, pp. 11-20.

[31] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient QoS-aware service composition," in Proceedings of the 18th international conference on World Wide Web (WWW'09), 2009, pp. 881-890.

Shangguang Wang is associate professor at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He received his Ph.D. degree at BUPT in 2011. His PhD thesis was awarded as outstanding doctoral dissertation by BUPT in 2012. Dr. Wang is 2015-2016 President of the Service Society Young Scientist Forum in China, General Co-Chair of ICCSA 2016, Application Track Co-Chair of IEEE SCC 2015, and Program Chair of the 2014 International Conference on Internet of Vehicles (IOV), Program Chair of the 2014 International Symposium on Cloud and Service Computing (SC2), and Special Track Chair of APSCC 2014. His research interests include Service Computing, Cloud Computing, and QoS Management.

Ao Zhou received her M.E. in computer science and technology from Beijing University of Posts and Telecommunications in 2012. She is currently a Ph.D. candidate at Beijing University of Posts and Telecommunications. Her research interests include cloud computing and service reliability.

Ching-Hsien Hsu is professor in the Department of Computer Science and Information Engineering at Chung Hua University, Taiwan. His research interests include high performance computing, cloud computing, parallel and distributed systems, and ubiquitous/pervasive computing and intelligence. He has been involved in more than 100 conferences and workshops as chair, and more than 200 conferences/workshops as program committee member. He is the editor-in-chief of the *International Journal of Grid and High Performance Computing*, and has served on the editorial boards of approximately 20 international journals.

Xuanyu Xiao is currently an undergraduate student at Beijing University of Posts and Telecommunications. His research interests include design of Microwave devices, intelligent hardware, and Mobile Cloud.

Fangchun Yang received his Ph.D.in communications and electronic systems from the Beijing University of Posts and Telecommunication in 1990. He is currently professor at the Beijing University of Posts and Telecommunication, China. He has published six books and more than 80 papers. His current research interests include network intelligence, service computing, communications software, soft-switching technology, and network security. He is a fellow of the IET.