Computer Networks 110 (2016) 46–57

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Green latency-aware data placement in data centers

Yuqi Fan[a,*], Hongli Ding[a], Lusheng Wang[a], Xiaojing Yuan[b]

[a] School of Computer and Information, Hefei University of Technology, Hefei, Anhui, 230009, China
[b] Department of Engineering Technology, University of Houston, Houston, Texas, 77004, USA

ARTICLE INFO

Article history:
Received 20 January 2016
Revised 5 August 2016
Accepted 19 September 2016
Available online 20 September 2016

Keywords:
Energy-efficient
Latency
Energy consumption of servers
Energy consumption of transport
Data deployment

ABSTRACT

Large-scale Internet applications provide service to end users by routing service requests to geographically distributed data centers. Two concerns exist in service provisioning by the data centers. One is that users require to experience low latency while accessing data from data centers. The other is to reduce the energy consumed by network transport and the servers in the data centers. In this paper, we tackle the problem of green data placement in data centers to strike a tradeoff among access latency, energy consumption of data centers and network transport. We propose two request-routing algorithms, GLDP-NS (Green Latency-aware Data Placement - No consideration of the current data placement Status of the server) and GLDP-WS (Green Latency-aware Data Placement - With consideration of the current data placement Status of the server). We show that the green latency-aware data placement problem is $\mathcal{NP}$-complete and algorithm GLDP-NS is a 3-approximation algorithm for the data placement problem without considering the data placement status of the server. We evaluate the performance of the proposed algorithms through simulations, and the simulation results demonstrate that the proposed algorithms can achieve good integrated cost performance of the latency, the energy consumption of data centers and network transport.

1. Introduction

Large-scale Internet applications, such as social networks, video distribution networks and content distribution networks, provide services to hundreds of millions of end users. The applications achieve enormous scalability and reduce access latency, by routing service requests to a set of geographically distributed data centers. For example, Google has more than 30 data centers in at least 15 countries with an estimated 900*K* servers [1] and Akamai (the biggest CDN corporation) has more than 95,000 servers in nearly 1,900 networks in 71 countries [2].

The issue of energy consumption in information technology equipment has been receiving increasing attention in recent years and there is an obvious need to reduce the greenhouse impact of the ICT sector [3–5]. The Energy Consumption Rating (ECR) Initiatives has published a specification on the energy assessment of networks and telecom equipments [6]. IEEE has ratified the IEEE P802.3az Energy-Efficient Ethernet (EEE) standard to address proactive reduction in energy consumption for networked devices [7]. It is expected that cloud computing will make significant contributions to reduce the energy consumption and carbon emissions effectively. However, [8] indicates that cloud services mainly focus on the performance of storage, processing and network transportation of data transmission between data centers and end users, with little consideration of the energy efficiency. The large-scale data centers hosting a large amount of servers are big consumers of electricity, which is used for servers and cooling system [9]. At the same time, the fast-expansion of Internet demand is also consuming increasingly more energy.

The surge of the usage of the cloud computing services makes many data centers be deployed all around the world. According to U.S. Environmental Protection Agency ENERGY STAR Program report, the data centers in USA consume 100 Billion *kWh* or 7.4 Billion dollars annually [10]. Currently, the data centers that power Internet-scale applications consume about 1.3% of the worldwide electricity supply [11]. The need to reduce energy consumption is driven by the engineering challenges and the cost of managing the energy consumption of large data centers and associated cooling [12]. Various approaches of energy saving of data centers have been proposed, such as dynamic voltage and frequency scaling (DVFS) control approaches [13–17], virtualization technologies [18–22], green resource reservation and allocation [23–28]. The DVFS scheme adjusts the CPU power (performance level) according to the offered load. Virtualization technology is based on loading more than one virtual machine (VM) on a physical server and,

* Corresponding author.
E-mail addresses: yuqi.fan@hfut.edu.cn (Y. Fan), Hong0li0Ding@gmail.com (H. Ding), lswang.enst@gmail.com (L. Wang), xyuan@uh.edu (X. Yuan).

http://dx.doi.org/10.1016/j.comnet.2016.09.015

thereby reducing the amount of hardware in use and improving the utilization of resources. In contrast, the scheme of green resource reservation and allocation can save more energy by powering down the components of computing servers.

Network transport is required to transmit data between users and data centers. The transmission and switching network equipments consume approximately 14.8% of the total ICT energy consumption, which will increase to 21.8% by 2020 [29]. The ever-growing size and number of network equipments also increase the energy consumption of the network [30] in both of the optical devices [31] and the electronic equipment [4,32][33,34]. are the first to come up with the novel idea towards green networking. Other research has been conducted on green networks since then, dealing with the energy consumption of network components [33,35–37], link data rate [7,38,39], and network design [40]. Some network components may be put into sleep mode during idle time to reducing energy consumption. The operators can adapt the link rate of network operation to the offered workload, reducing the energy consumed when actively processing packets.

For the cloud service users, latency is an important concern. The high access latency has been shown to have a negative economic impact [41], since both users and applications require low network latency. Some applications even require stringent latency guarantees in the order of nanoseconds [42]. Low latency will simplify application development and increase web application scalability [43]. The access latency between the users and data centers are related to the data center locations and Internet routing between the data centers and the users [44]. Recently, several proposals are put forward to reduce the network latency, which includes the rise of the data centers and the next generation of Ethernet switching chips [43]. Data centers can be built close to their users. New switching chips can promise to make their bandwidth plentiful and cheap.

There has been some work on reducing the electricity consumption and carbon emissions of the data centers and the networks in recent years. A request-routing scheme to minimize the electricity bill of multi-datacenter systems is proposed in [45]. [46] improves the algorithms in [45] on multi-region electricity markets to better capture the fluctuating electricity price to reduce electricity cost. [47] proposes a resource management framework allowing cloud providers to provision resources across a geo-distributed infrastructure with the aim to reduce operational costs and green SLA violation penalties, under the constraint that carbon emissions generated by the leased resources should not exceed a fixed bound. For the operational cost minimization problem in a distributed cloud computing environment that not only considers fair request rate allocations among web portals but also meets various Service Level Agreements (SLAs) between users and the cloud service provider [48], proposes an adaptive operational cost optimization framework incorporating time-varying electricity prices and dynamic user request rates, and devises an approximation algorithm to maximize the number of user requests admitted [49]. considers the joint optimization problem of minimizing carbon emission and electricity cost. [50] proposes an algorithm to geographically balance load while taking carbon emission into account. [51] adjusts the number of servers running in data centers for a tradeoff between latency and carbon emissions [8]. provides a method to calculate the energy consumption of the network, which can estimate the energy consumption required to transport one bit from a data center to a user through the Internet [52]. jointly considers the electricity cost, service level agreement (SLA) requirement, and emission reduction budget by exploiting the spatial and temporal variabilities of the electricity carbon footprint. [9] proposes a request-routing scheme, FORTE, allowing operators to strike a tradeoff among electricity costs, access latency, and carbon emissions. The carbon emissions of servers in the data centers are closely related with the amount of electricity consumed and the resources used to produce the electricity.

To the best of our knowledge, there is little information available in literature about considering the three factors of the latency, the energy consumption of the servers and the network transport when placing data in the data centers. In this paper, we tackle the problem of energy-efficient data placement in the data centers using an objective function that incorporates the three factors above.

The main contributions of this paper are as follows. We investigate the data placement problem to enable the tradeoff among the access latency, the energy consumption of the servers in the data centers, and the energy consumed by the network transport. Data placement cost calculation incorporates the three factors above, and propose two request-routing algorithms, *GLDP-WS* (Green Latency-aware Data Placement - With consideration of the current data placement Status of the server) and *GLDP-NS* (Green Latency-aware Data Placement - No consideration of the current data placement Status of the server) based on the proposed placement metric. We also conduct experiments through simulations to evaluate the performance of the proposed algorithms. Experimental results demonstrate the proposed algorithms are very promising.

The rest of the paper is organized as follows. The problem under study is formally defined in Section 2. The algorithms GLDP-NS and GLDP-WS are presented in Section 3. Section 4 reports the performance evaluation. The paper concludes in Section 5.

2. Problem formulation

Data centers serve users by providing the data required by the users. Each data chunk, i.e. each piece of data, required by the users must be placed in a server in a data center. A data chunk may be accessed by all the users. The data centers retrieve the data from the servers and transmit the data to the users through Internet when the users require the data. For example, a video-sharing website may place the videos in the data center servers, and the users worldwide can watch the videos retrieved by the website.

While placing a data chunk in a data center, we consider three factors: (1) the access latency of the data, (2) the energy consumption of the network transport for data transmission between the users and the data centers, and (3) the energy consumed by the servers in the data centers.

The network model for the data transmission between the data centers and the users through Internet is shown in Fig. 1, which is similar to the one in [8]. The access network is modeled as a PON [53]. The energy consumption of the access network is largely independent of traffic volume [54]. Therefore, the access network does not influence the result as it is a fixed value. The energy $e_I(u_i, dc_j)$ required to transport one bit from a data center to a user through the Internet is estimated via Eq. (1) similar to similar to [8].

$$e_I(u_i, dc_j) = 6\left(3\frac{P_{es}}{C_{es}} + \frac{P_{bg}}{C_{bg}} + \frac{P_g}{C_g} + 2\frac{P_{pe}}{C_{pe}}\right.$$
$$\left. +2\frac{P_c}{C_c}h_c(u_i, dc_j) + \frac{P_w}{2C_w}h_c(u_i, dc_j)\right) \quad (1)$$

where P_{es}, P_{bg}, P_g, P_{pe}, P_c and P_w are the power consumed by the Ethernet switches, broadband gateway routers, data center gateway routers, provider edge routers, core routers, and WDM transport equipment, respectively. C_{es}, C_{bg}, C_g, C_{pe}, C_c and C_w are the capacities of the corresponding equipment in bits per second. The factor of six accounts for the power requirements for redundancy (factor of 2), cooling and other overheads (factor of 1.5), and the fact that current network typically operate at under 50% utilization [55] while still consuming almost 100% of maximum power

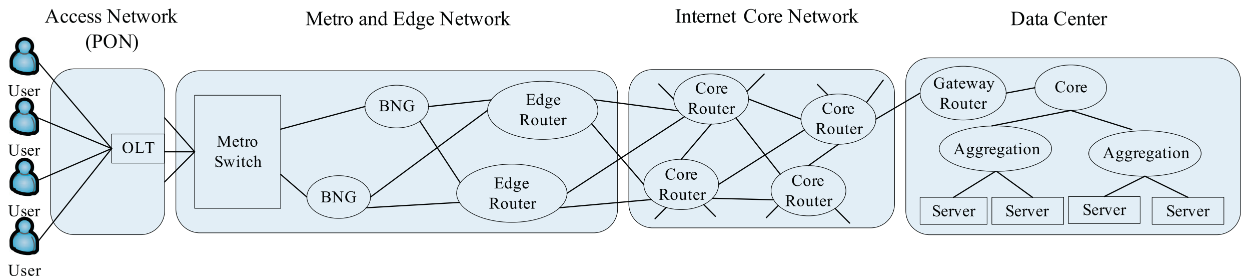

Fig. 1. Schematic of networks connecting users to a data centers and the data center infrastructure.

[56] (factor of 2). We assume the power usage effectiveness (PUE) of the Internet is 1.5, where PUE is a measure of how efficiently a data center delivers power to the computing equipment. The factor of three for Ethernet switches is to include the Ethernet switches in the metro network as well as the Ethernet switches in the core inside the data center. The factor of two for edge routers is to include the edge router in the edge network and the gateway router in the data center. The factor of two for core routers allows for the fact that core routers are usually provisioned for future growth of double the current demand [57]. The factor of $h_c(u_i, dc_j)$ accounts for the number of hops during the data transmission in the core network. In general, the distance between user u_i and data center dc_j provides $h_c(u_i, dc_j)$ step increases.

We assume a server will consume the full-system power when the server is on, because (1) it is an estimator accurate enough to determine the relative rank in energy consumption; (2) no general analytical model of server energy consumption is available for various server models at different work loads [58,59].

Some notations used in this paper are listed as follows:

- $p(u_i|d_k)$ is the probability that a given request is asking for data d_k and it comes from user u_i;
- $s(d_k)$ is the size of data d_k;
- $l(u_i, dc_j, d_k)$ is the average latency between user u_i and data center dc_j for data d_k;
- $rep(dc_j, s_m, d_k)$ is 1 if data d_k is placed in server s_m in data center dc_j; otherwise, it is 0;
- $rep(dc_j, s_m)$ is 1 if server s_m in data center dc_j has been placed some data chunks; otherwise, it is 0;
- $e_S(s_m, dc_j)$ is the average energy consumption of server s_m in data center dc_j;
- $PUE(dc_j)$ is the PUE of data center dc_j;
- $P_{s_m}^{dc_j}$ is the average working power of sever s_m in data center dc_j;
- $C(s_m, dc_j)$ is the storage capacity of server s_m in data center dc_j.

In this paper we aim to find a proper placement location for each data chunk, with the objective to strike a tradeoff among the data access latency, the energy consumption of the severs and the network transport for data access. The problem is formulated as follows.

Minimize:

$$\lambda_1 \sum_{u_i, dc_j, s_m, d_k} rep(dc_j, s_m, d_k) p(u_i \mid d_k) l(u_i, dc_j, d_k)$$
$$+\lambda_2 \sum_{dc_j, s_m} rep(dc_j, s_m) e_S(dc_j, s_m)$$
$$+\lambda_3 \sum_{u_i, dc_j, s_m, d_k} s(d_k) rep(dc_j, s_m, d_k) p(u_i \mid d_k) e_I(u_i, dc_j) \quad (2)$$

Subject to:

$$rep(dc_j, s_m) = min\left(1, \sum_{d_k} rep(dc_j, s_m, d_k)\right), \forall dc_j, s_m \quad (3)$$

$$\sum_{dc_j, s_m} rep(dc_j, s_m, d_k) = 1, \forall d_k \quad (4)$$

$$\sum_{u_i} p(u_i \mid d_k) = 1, \forall d_k \quad (5)$$

$$e_S(dc_j, s_m) = P_{s_m}^{dc_j} * PUE(dc_j) \quad (6)$$

$$\sum_{d_k} rep(dc_j, s_m, d_k) s(d_k) \le C(s_m, dc_j), \forall dc_j, s_m \quad (7)$$

λ_1, λ_2, and λ_3 in Eq. (2) are constant normalized weights used for weighting among the three sub-objectives of the latency, the energy consumption of the servers in the data centers and the energy consumed by the network transport. Eq. (3) mandates the data placement incurs access delay and energy consumption. Eq. (4) requires each data chunk to be placed in a data center. Eq. (5) determines the request for a data chunk comes from one of the users. Eq. (6) defines that the energy consumption of the servers should take into account the PUE of the data center. Eq. (7) dictates the size of the data stored in a server cannot exceed the capacity of the server.

3. Green latency-aware data placement algorithms

Note that the data placement status of the servers may be different when a data chunk is to be placed. Depending on how to calculate the energy consumption cost of the server, we divide the data placement problem into two sub-problems: (1) Data Placement problem with No consideration of the current data placement Status of the server (*DPNS*), where the server energy consumption cost of accommodating a data chunk is closely related to the power of the server; (2) Data Placement problem With consideration of the current data placement Status of the server (*DPWS*), where the server energy consumption cost of a server is only incurred when the server has to accommodate some data without regard to the number of data placed in the server.

Example 1. In Fig. 2, data d_k needs to be placed, and both server s_a in data center dc_1 and server s_b in data center dc_2 have enough storage capacities to accommodate data d_k. The average working power of servers s_a and s_b is 1000*W* and 900*W*, respectively. Data centers dc_1 and dc_2 have the same PUE. The distances from each user to data centers dc_1 and dc_2 are also the same. Server s_a has been placed some data, while server s_b has not. In problem DPWS, data d_k will be placed in server s_a, since the server energy consumption cost is viewed as 0. In contrast, data d_k will be placed in server s_b in problem DPNS, since the power of server s_b is less than that of server s_a.

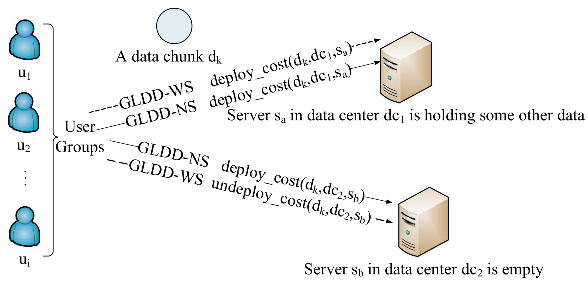

Fig. 2. Example of data placement for DPWS and DPNS.

Theorem 1. *Data placement problem DPNS is $\mathcal{NP}$-complete.*

Proof. We show that the *Generalized Assignment Problem (GAP)*, a well-known $\mathcal{NP}$-complete problem, can be reduced to the data placement problem DPNS [60]. The GAP problem is defined as follows: A pair (B, S) where B is a set of M bins (knapsacks) and S is a set of N items. Each bin $C_{m'} \in B$ has capacity $c[m']$, and for each item k and bin $C_{m'}$ we are given a size $s[k, m']$ and a profit $p(k, m')$. The objective is to find a subset $U \subseteq S$ of items that has a feasible packing in B, such that the profit is maximized.

The server set consisting of all the servers in all the data centers in the data placement problem DPNS is $S' = \{s'_1, s'_2, s'_3, \ldots, s'_{m'}, \ldots\}$, where each server s_m in each data center dc_j corresponds to a server $s'_{m'}$ in S'. The capacity of server $s'_{m'}$ is denoted as $C(s'_{m'})$. The polynomial-time reduction from one instance of the GAP problem to one instance of the data placement problem DPNS is as follows. B, S, $c[m']$ and $s[k, m']$ in the GAP problem are equivalent to the set of servers S', the set of data chunks $\{d_1, d_2, d_3, \ldots\}$, the capacity of each server $C(s'_{m'})$, and the size of each data $s(d_k)$ in the data placement problem, respectively. Assuming the maximum cost of data placement in the DPNS problem is *maxcost*, the profit of placing data d_k in server s_m in data center dc_j is $maxcost - cost(d_k, dc_j, s_m)$. The profit $p(k, m')$ in the GAP problem is essentially the same as the profit of placing a specific data chunk d_k in a specific server $s_{m'}$ in the data placement problem DPNS. By this reduction, we can easily see that there is a solution to one instance of the data placement problem DPNS if and only if there is a solution to one instance of the GAP problem. Hence, the data placement problem DPNS is $\mathcal{NP}$-complete. □

For data placement problem DPWS, the cost to place data d_k in server s_m in data center dc_j is dynamic depending on whether server s_m has been accommodating some data, which is more complex than the data placement problem with DPNS. Therefore, the data placement problem DPWS is also $\mathcal{NP}$-complete.

We propose two algorithms GLDP-NS and GLDP-WS for the two data placement sub-problems DPNS and DPWS, respectively. If server s_m has not accommodated any data before placing data d_k, both algorithms GLDP-NS and GLDP-WS calculate the data placement cost with Eq. (8). Otherwise, the placement cost is computed via Eq. (9) with GLDP-WS, while the placement cost is calculated via Eq. (8) with GLDP-NS. That is, the data placement will not incur any additional server energy consumption if the server is accommodating some other data.

$$\begin{aligned} cost(d_k, dc_j, s_m) &= \lambda_1 \sum_{u_i} l(u_i, dc_j) p(u_i \mid d_k) + \lambda_2 e_S(dc_j, s_m) \\ &\quad + \lambda_3 \sum_{u_i} s(d_k) e_I(u_i, dc_j) p(u_i \mid d_k) \end{aligned} \tag{8}$$

$$\begin{aligned} cost'(d_k, dc_j, s_m) &= \lambda_1 \sum_{u_i} l(u_i, dc_j) p(u_i \mid d_k) \\ &\quad + \lambda_3 \sum_{u_i} s(d_k) e_I(u_i, dc_j) p(u_i \mid d_k) \end{aligned} \tag{9}$$

3.1. GLDP-NS Algorithm

Assuming M and N are the number of data center servers and the number of data chunks, respectively. The proposed algorithm GLDP-NS is described in Algorithm 1, which places the data in the data center servers. Specifically, the algorithm proceeds iteratively. Within each iteration, a single server is filled up with data. This procedure continues until all data are placed.

Function PDS (Place Data in a Server) shown in Algorithm 2 places data in a server $s'_{m'}$ so as to make the placement profit maximized. That is, PDS tries to maximize the placement profit of every unit capacity in server $s'_{m'}$. The profit density value is introduced to denote the ratio of profit $p(k, m')$ to data size $s(d_k)$. PDS keeps putting the data in $s'_{m'}$ in the non-ascending order of the profit density value, until server $s'_{m'}$ cannot accommodate any other data. Note that server $s'_{m'}$ may have some free space not large enough to accommodate the unplaced data, which reduces the placement profit of every unit capacity in server $s'_{m'}$. Therefore, if the total profit of the selected data is less than the profit of the first data $d_{k'}$ which cannot be put in $s'_{m'}$, PDS removes all the selected data in server $s'_{m'}$ and places data $d_{k'}$ in server $s'_{m'}$.

In algorithm GLDP-NS, ρ is the profit matrix, where each entry $\rho_{k,m'}$ in ρ is the profit of placing data d_k in server $s'_{m'}$. $\rho_{m'}$ indicates the profit matrix when dealing with server $s'_{m'}$. Algorithm GLDP-NS deals with all the servers iteratively with the initial profit matrix ρ, in which each entry is $\rho_{k,m'} = p(k, m')$. After performing function PDS for each server $s_{m'}$ using the profit matrix $\rho_{m'}$, GLDP-NS decomposes the profit matrix $\rho_{m'}$ into two profit matrices $\rho^1_{m'}$ and $\rho^2_{m'}$. This decomposition implies that $\rho^1_{m'}$ is identical to $\rho_{m'}$ with regard to server $s'_{m'}$; in addition, if data $d_k \in \overline{S_{m'}}$, then the placement of d_k is assigned the same profit as $\rho_{m'}[y, m']$ for all the servers in $\rho^1_{m'}$. All the other entries in $\rho^1_{m'}$ are assigned the value 0. The data are placed in the servers in descending order of server index. If data d_k is placed in server $s'_{m'}$, d_k is removed from the selected data set $\overline{S_\gamma}$, where $1 \le \gamma \le m' - 1$.

Theorem 2. *PDS described in Algorithm 2 is a 2-approximation algorithm for placing data in a server.*

Proof. A profit $p(k, m')$ can be obtained for each data chunk $d_k \in \{d_1, d_2, d_3, \ldots\}$ when placed in server $s'_{m'}$. Assuming $d_{k'}$ is the first data chunk which cannot be put in $s'_{m'}$, $f()$ is the profit function, K is the selected data set, K^* is the best solution for placing data on server $s'_{m'}$, and $p(\theta, m') = \max\{p(1, m'), p(2, m'), p(3, m'), \ldots\}$, we can get $f(K^*) \le \sum_{k=1}^{k'-1} p(k, m') + p(\theta, m')$. Note that $f(K) = \max\{\sum_{k=1}^{k'-1} p(k, m'), p(\theta, m')\} \ge 0.5(\sum_{k=1}^{k'-1} p(k, m') + p(\theta, m')) \ge 0.5 f(K^*)$. Therefore, the approximation ratio of PDS is 2. □

Theorem 3. *$F()$, $F_1()$, and $F_2()$ are functions to a problem with a set of constrains $\mathcal{C}$, and $F() = F_1() + F_2()$. If a is an α-approximate solution to $(\mathcal{C}, F_1())$ and $(\mathcal{C}, F_2())$, it is also an α-approximation solution to $(\mathcal{C}, F())$.*

Proof. Assuming a^*, a^*_1 and a^*_2 are the optimal solutions for $(\mathcal{C}, F())$, $(\mathcal{C}, F_1())$ and $(\mathcal{C}, F_2())$, respectively, $F(a) = F_1(a) + F_2(a) \ge \alpha * F_1(a^*_1) + \alpha * F_2(a^*_2) \ge \alpha * (F_1(a^*) + F_2(a^*)) \ge \alpha * F(a^*)$. □

Theorem 4. *Algorithm GLDP-NS described in Algorithm 1 is 3-approximation for the data placement problem DPNS.*

Proof. Assume $\rho(S)$ is the profit gained by data placement S. The proof is given by induction.

Base case: When there is only one server, the data placement returned by the algorithm, $S_M = \overline{S_M}$, is 2-approximation because $\overline{S_M}$ is the result produced by function PDS which is 2-approximation. Therefore, S_M is 3-approximation with $\rho(M)$.

Algorithm 1 GLDP-NS Algorithm

Set $\rho_1 \leftarrow \rho$ and $m' = 1$;
for $m' = 1, 2, 3, \dots, M-1$ **do**
  PDS(m') using $\rho_{m'}$ as the profit matrix, and let $\overline{S}_{m'}$ be the set of data selected by PDS(m');
  Decompose the profit matrix $\rho_{m'}$ into two profit matrices $\rho^1_{m'}$ and $\rho^2_{m'}$ such that for every x and y, where $1 \le x \le M$ and $1 \le y \le N$,
$$\rho^1_{m'}[y, x] = \begin{cases} \rho_{m'}[y, m'] & if \quad (y \in \overline{S_{m'}}) \quad or \quad (x = m') \\ 0 & Otherwise \end{cases} \qquad and \qquad \rho^2_{m'} = \rho_{m'} - \rho^1_{m'} \quad ;$$
  Set $\rho_{m'+1} \leftarrow \rho^2_{m'}$;
  Remove the column of server $s'_{m'}$ from $\rho_{m'+1}$;
end for
PDS(M) using ρ_M as the profit matrix;
$S_M = \overline{S_M}$
for $m' = M-1 .. 1$ **do**
  for each $d_k \in \overline{S_{m'}}$ **do**
    if $d_k \in \cup^M_{i=m'+1} \overline{S_i}$ **then**
      $A_{m'} = \overline{S_{m'}} - d_k$ $//A_{m'}$ is a temporary variable;
    end if
    $S_{m'} = \{\cup^M_{i=m'+1} S_i\} + A_{m'}$;
  end for
end for
Return S_1 in which m'-th element is the set of data placed on server $s_{m'}$.

Algorithm 2 PDS Algorithm (m')

Input: Data Request from users Probability Matrix $P(u_i \mid d_k)$
Input: Network Latency Cost Matrix $L(u_i, dc_j)$
Input: Network Power Cost Matrix $E_I(u_i, dc_j)$
Input: Servers Power Cost Matrix $E_S(u_i, dc_j)$
Input: Data Size Matrix $S(d_k)$
Output: $Rep(m', d_k)$
Calculate the profit of placing each data d_k in server $s'_{m'}$;
Sort d_k by the non-ascending order of the profit density value $\frac{p(k,m')}{s(d_k)}$ and put each data chunk in queue;
while server $s'_{m'}$ has enough capacity to accommodate data d_k **do**
  Select data d_k and put d_k in server $s'_{m'}$;
  $C(s'_{m'}) = C(s'_{m'}) - s(d_k)$;
end while
if the total profit of the selected data in $s'_{m'}$ < the profit of the first data $d_{k'}$ which cannot be put in $s'_{m'}$ **then**
  Remove all the selected data in $s'_{m'}$ and place $d_{k'}$ in $s'_{m'}$;
end if
Return $Rep(m', d_k)$.

Induction step: Suppose that $S_{m'+1}$ is a 3-approximation with respect to $\rho_{m'+1}$, and we prove that $S_{m'}$ is also a 3-approximation with respect to $\rho_{m'}$. There are three parts in profit matrix $\rho^1_{m'}$: (1) the column for server $s'_{m'}$, which is the same as the one in $\rho_{m'}$; (2) the rows for data in $\overline{S_{m'}}$, which is selected by PDS(m'); and (3) all the other entries with value 0. Note that the first two parts in $\rho^1_{m'}$ can contribute to the data placement, while the third part cannot. The best result for the first part is $2\rho^1_{m'}(\overline{S_{m'}})$, because $\overline{S'_m}$ is the result produced by function PDS. The optimal result for the second part is at most $\rho^1_{m'}(\overline{S_{m'}})$, because the placement of data in $\overline{S_{m'}}$ in $\rho^1_{m'}$ is assigned the same profit for all the servers. Therefore, $\overline{S_{m'}}$ is a $(1+2)$-approximation, 3-approximation, with $\rho^1_{m'}$. According to algorithm GLDP-NS, $\overline{S_{m'}}$, the set of data selected for server $s'_{m'}$ is a subset of $S_{m'}$. Therefore $\rho(S_{m'}) \ge \rho(\overline{S_{m'}})$, and $S_{m'}$ is a 3-approximation solution with $\rho^1_{m'}$.

Profit matrix $\rho^2_{m'}$ is basically the same as $\rho_{m'+1}$, except that $\rho^2_{m'}$ contains an extra 0-value column which is the column for server $s_{m'}$. $S_{m'+1}$ is 2-approximation with $\rho_{m'+1}$, and obviously $S_{m'+1}$ is 3-approximation with $\rho^2_{m'}$. Therefore, $S_{m'}$ is also 3-approximation with $\rho^2_{m'}$ because $S_{m'}$ contains the data in $S_{m'+1}$. Note that $\rho_{m'} = \rho^1_{m'} + \rho^2_{m'}$ and $S_{m'}$ is 3-approximation with $\rho^1_{m'}$ and $\rho^2_{m'}$. According to Theorem 3, $S_{m'}$ is a 3-approximation with $\rho_{m'}$.

Conclusion: By the principle of induction, algorithm GLDP-NS described in Algorithm 1 is 3-approximation for the data placement problem DPNS. □

The sort of data set takes $O(Nlog(N))$ time, and the processing of each data chunk runs in time $O(N)$. Therefore, function PDS can be performed in $O(Nlog(N))$ time, and the time complexity of algorithm GLDP-NS is $O(MNlog(N))$.

3.2. GLDP-WS Algorithm

The proposed heuristic algorithm GLDP-WS for the data placement problem DPWS is described in Algorithm 3. Algorithm GLDP-WS solves the multiple constrained optimization problem of data placement taking into account the latency, and the energy consumption of the servers and the network transport. The algorithm proceeds iteratively. Within each iteration, a single data chunk is placed. This procedure continues until all data are placed. GLDP-WS sorts and processes the data in non-ascending order of data size, since a data chunk with a larger data size incurs more energy consumption. When processing each data chunk d_k, GLDP-WS searches the servers in all data centers with the least cost to place data d_k. The data placement cost for each server is calculated if data d_k is accommodated by the server. The cost of placing data d_k in server s_m in data center dc_j incorporates the three factors of the latency, the energy consumed by the servers and the network transport, which is calculated via Eqs. (8) or (9). In this way, the data are inclined to be aggregated on a subset of the servers so that the number of servers used is reduced and the servers that are not needed can be turned off. In general, the access latency of a data chunk increases with the increase of the distance between the user and the data center accommodating the data. The energy consumed by the network transport for the data is estimated via Eq. (1).

The sort of $S(d_k)$ in GLDP-WS can be performed in $O(Nlog(N))$ time, and the processing of each data chunk d_k is executed within

Algorithm 3 GLDP-WS Algorithm

Input: Data Request from users Probability Matrix $P(u_i \mid d_k)$
Input: Network Latency Cost Matrix $L(u_i, dc_j)$
Input: Network Power Cost Matrix $E_I(u_i, dc_j)$
Input: Servers Power Cost Matrix $E_S(u_i, dc_j)$
Input: Data Size Matrix $S(d_k)$
Output: $Rep(dc_j, s_m, d_k)$

```
Sort S(d_k) by non-ascending order of data size and put each data chunk in queue.
while Queue of S(d_k) not empty do
  for each data center dc_j do
    for each server s_m in data center dc_j do
      if server s_m has enough capacity to accommodate data d_k then
        Calculate the cost to place data d_k on server s_m in data center dc_j with Eq. (9) or Eq. (8);
      end if
    end for
  end for
  Obtain server s_m in data center dc_j that incurs the least cost and has enough capacity to accommodate data d_k;
  C(s_m, dc_j) = C(s_m, dc_j)) - s(d_k).
  rep(dc_j, s_m, d_k)=true;
end while
Return Rep(dc_j, s_m, d_k)
```

Table 1
Average distance from users to data centers and the PUE of the data centers.

Datacenter	dc_1	dc_2	dc_3	dc_4	dc_5
Average distance	1000	900	800	700	600
PUE	1.7	1.5	1.6	1.3	1.4

$O(MN)$ time. Therefore, the time complexity of algorithm GLDP-WS is $O(MN + N\log(N))$.

4. Simulation

4.1. Simulation setup

We evaluate the performance of the proposed algorithms GLDP-WS and GLDP-NS by comparing them with the algorithm FORTE proposed in [9] which is the most similar to our work. FORTE tries to strike a tradeoff among the carbon footprint, the electricity costs of the servers and the latency, without consideration of the energy consumed by the network transport. The carbon emissions of the servers in the data centers are closely related to the amount of electricity consumed in a specific area. The objective of FORTE indicates that both the electricity costs and carbon emissions increase with the number of the servers used in the data centers. With FORTE, a data chunk may be placed in one or more data centers, while GLDP-WS and GLDP-NS place each data chunk in a data center. Furthermore, GLDP-WS and GLDP-NS aim to strike a tradeoff among the latency of data access, the energy consumed by the network transport and the servers in the data centers.

We use geographical distance as an approximation for latency similar to [9], because (1) it is sufficiently accurate to determine the relative rank in latency from datacenters to each end-user; (2) there is no general analytical model available for the delay in the network. Table 1 shows the relationship between the average distance between all users and a data center and the PUE of the data center, given the data center set $\{dc_1, dc_2, \cdots, dc_5\}$. The PUE for each data center is randomly generated in the range of [1.3,1.7]. The request for a data chunk from a user is random, and any request for a data chunk comes from one of the users. Table 2 dictates the quantity of data chunks with various sizes in percentage. Table 3 shows the number of servers with different parameters in the data centers. Table 4 lists the equipment used in the network [4], which are obtained from manufacturers' data sheets [61–63]. The energy consumption and capacities of network equipment are also given in Table 4. The edge routers are presumed to be located closely and do not require additional wavelength division multiplexed (WDM) transponder systems. The number of WDM and core routers in the core network is equal to the number of hops in the core network, which is related to the distance from the users to the data centers as described in Table 5. The number of hops has an upper bound of 16. The simulation is run till 95% confidence level is achieved.

Table 2
Number of data with different sizes.

Data size(GB)	1–5	5–10	10–20	20–30	30–50
Quantity(%)	30%	25%	20%	15%	10%

Table 3
Number of servers with different power and capacities in each data center.

Power(W)	Capacity(TB)	Number
2000	250	2
1700	200	4
1300	150	6
900	100	8
500	50	10

Table 4
Equipment in the network.

	Equipment	Capacity(Gb/s)	Power
Gateway Router	Juniper MX-960	660 [61]	5.1kW [61]
Ethernet Switch	Cisco 6509	160 [62]	3.8kW [62]
BNG	Juniper E320	60 [61]	3.3kW [61]
Provider Edge	Cisco 12816	160 [62]	4.21kW [62]
Core router	Cisco CRS-1	640 [62]	10.9kW [62]
WDM	Fujitsu 7700	40 [63]	136W/channel [63]

According to the data placement policy of FORTE, a data chunk may be placed in multiple data centers. Data d_k is placed in data center dc_j if $s(d_k)$ is large enough; that is, d_k should be among a top percentile of the large flows across a set of flows. The top percentile is defined by a percentile threshold. In our simulation, we set the percentile threshold as 0 to guarantee that all the data can be placed in some data center(s).

We investigate four cases in the simulation. *Case* 1 only considers the sub-objective of the latency in the objective function ($\lambda_1 = 1,\ \lambda_2 = \lambda_3 = 0$). In *case* 2, the server energy consumption in the data centers is the only objective and ignores the latency and the energy consumption of the network ($\lambda_2 = 1,\ \lambda_1 = \lambda_3 = 0$). The latency and the data center energy consumption are the two factors considered during data placement in *case* 3 ($\lambda_1 = \lambda_2 = 1$, $\lambda_3 = 0$). *Case* 4 ($\lambda_1 = \lambda_2 = \lambda_3 = 1$) tries to strike a three-way tradeoff among the latency, the energy consumed by the network transport and the servers in the data centers. Note that both the energy consumption of the network and the latency increase with the increase of the distance, and we will achieve the similar results to *case* 1 when the energy consumption of the network is the only objective during data placement. Therefore, we skip this case in the simulation description.

Table 5
Relationship between the distance from a user to a data center and the number of hops in the core network.

Distance(km)	[0, 300)	[300, 400)	[400, 500)	[500, 600)	[600, 700)	[700, 800)
hops	1	2	3	4	5	6
Distance(km)	[800, 900)	[900, 1000)	[1000, 1100)	[1100, 1200)	[1200, ∞)	
hops	8	10	12	14	16	

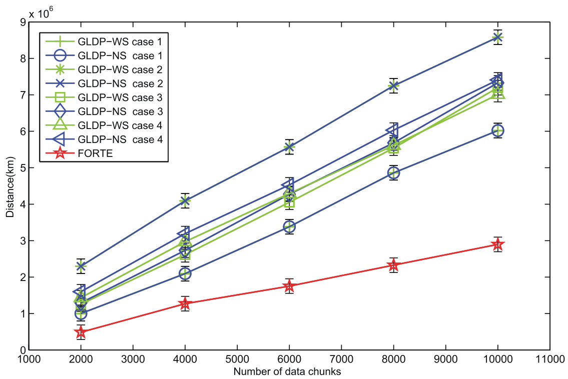

Fig. 3. Distance with the algorithms of GLDP-WS, GLDP-NS and FORTE.

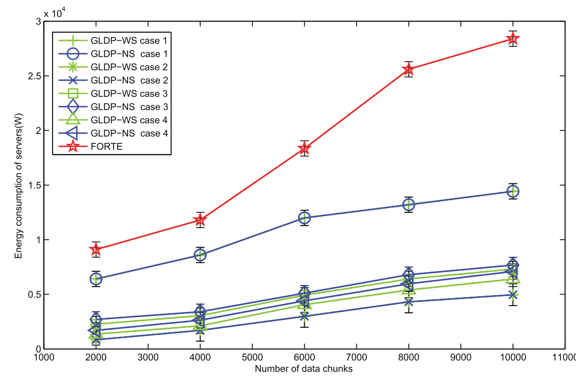

Fig. 4. Energy consumption of servers with the algorithms of GLDP-WS, GLDP-NS and FORTE.

4.2. Performance evaluation of the proposed algorithms

4.2.1. Impact of the number of data chunks

We first evaluate the performance of different algorithms GLDP-WS, GLDP-NS and FORTE by varying the number of data chunks, assuming the number of users is 1000.

Fig. 3 demonstrates that in general the distance increases with the growth of the number of data, which is also shown in Eqs. (2), (8) and (9). FORTE results in the least distance, because FORTE places each data chunk in one or more data centers and each user can access the data from the data center located closest to the user. GLDP-WS and GLDP-NS achieve the same result in both *case* 1 and *case* 2, since the data placement status of the servers in the data centers does not affect the cost calculation during the data placement in these two cases. Among the four cases, GLDP-WS and GLDP-NS perform the best in *case* 1, as distance is the only factor affecting the data placement decision process. *Case* 2 results in the worst performance, since *case* 2 only considers the energy consumption of the data centers and ignores the distance, which potentially makes the users access the data from a data center far away. GLDP-WS and GLDP-NS achieve better results in *case* 3 than in *case* 4. *Case* 4 tries to strike a three-way tradeoff among the distance, the energy consumed by the network transport and the data centers, and the decrease of the distance from the users to the data centers incurs less energy consumed by the networks.

Fig. 4 illustrates that the energy consumption of the servers in the data centers increases with the growth of the number of data, because we need more servers to accommodate the data. FORTE consumes the most energy, since FORTE places each data chunk in one or more data centers, while each data chunk is placed in a data center with GLDP-WS and GLDP-NS. GLDP-WS and GLDP-NS achieve the same performance in both *case* 1 and *case* 2, since the data placement status of the servers in the data centers has no impact on the data placement cost evaluation in these two cases. GLDP-WS and GLDP-NS perform the best in *case* 2, because data center energy consumption is the only factor affecting the data placement. *Case* 1 cares only about the distance, and hence the data may be accommodated by a data center close to the users which has a big PUE or uses the servers with high power. GLDP-WS and GLDP-NS obtain better results in *case* 4 than in *case* 3.

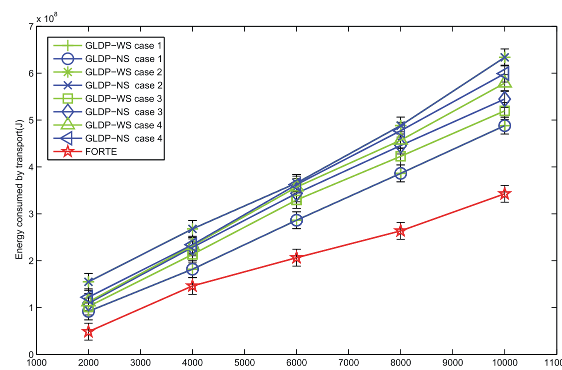

Fig. 5. Energy consumed by transport with the algorithms of GLDP-WS, GLDP-NS and FORTE.

Case 3 considers two factors of the distance and the data center energy consumption, while *case* 4 has to achieve the tradeoff among all the three factors and the increase of the distance also leads to more energy consumption in network transport. In *case* 3 and *case* 4, we observe GLDP-WS outperforms GLDP-NS. GLDP-WS considers the data placement status of the servers before placing data, and thereby the data are potentially accommodated by a subset of the servers.

Fig. 5 shows that the energy consumed by network transport increases with the growing number of data, since more data transfer incurs more energy consumption in the networks. FORTE results in the least energy consumed by network transport. With FORTE, the data go through shorter distances between the data centers and the users than with GLDP-WS and GLDP-NS, which potentially reduces the number of network devices needed for data transmission as shown in Eq. (1). GLDP-WS and GLDP-NS achieve the same results in both *case* 1 and *case* 2, since the data placement status of the servers in the data centers is ignored in these two cases. Among the four cases of GLDP-WS and GLDP-NS, *case* 1 achieves the best performance. The distance is the only factor considered during data placement, while the shorter distances for data access lead to less network equipments required for data access. In contrast, *case* 2 consumes the most energy, as *case* 2 only considers

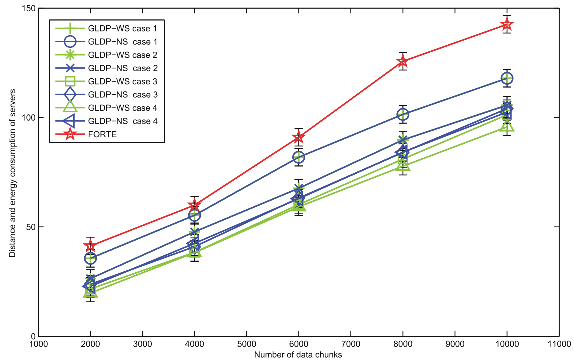

Fig. 6. Distance and energy consumption of servers with the algorithms of GLDP-WS, GLDP-NS and FORTE.

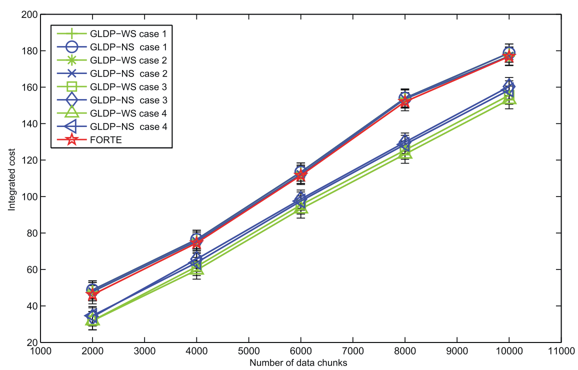

Fig. 7. Integrated cost with the algorithms of GLDP-WS, GLDP-NS and FORTE.

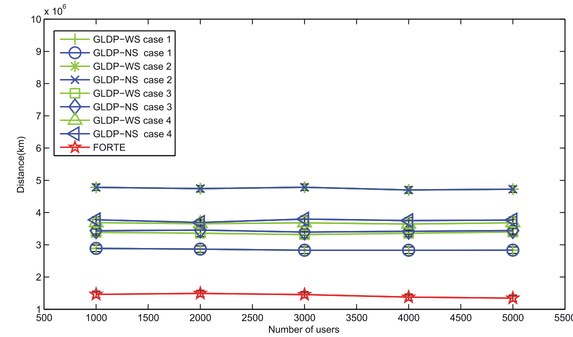

Fig. 8. Latency with the algorithms of GLDP-WS, GLDP-NS and FORTE as the increasing number of users.

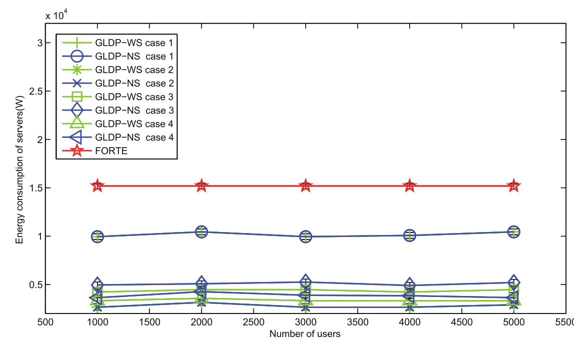

Fig. 9. Energy consumption of servers with the algorithms of GLDP-WS, GLDP-NS and FORTE as the increasing number of users.

the energy consumption of the data centers and ignores the energy consumed by the networks, which may increase the data transmission distances between the users and the data centers. GLDP-WS and GLDP-NS perform better in *case* 3 than in *case* 4. *Case* 4 considers all the three factors, and the distance from the users to the data centers has a negative impact on the energy consumed by the networks.

The performance of the cost incorporating the distance and the energy consumption of the data centers is given in Fig. 6. GLDP-WS and GLDP-NS outperform FORTE in all cases, and GLDP-WS in *case* 4 achieves the best performance. Both GLDP-NS and GLDP-WS with *case* 1 outperforms FORTE up to 12.3%, while the results of GLDP-WS with *case* 4 are better than those of FORTE from 26% to 50%. A data chunk with FORTE may be placed in one or more data centers, which reduces the distance at the cost of incurring more energy consumption of the data centers. The simulation results of GLDP-WS and GLDP-NS are the same in both *case* 1 and *case* 2. In *case* 3 and *case* 4, GLDP-WS obtains better results than GLDP-NS, as GLDP-WS considers the data placement status of the servers in the data centers. The data are inclined to be aggregated on a subset of the servers so that the number of servers used is reduced and the servers that are not needed can be turned off.

The performance in terms of the integrated cost of the distance, the energy consumed by the data center servers and the network transport is depicted in Fig. 7. GLDP-WS and GLDP-NS achieve better results than FORTE in *case* 3 and *case* 4. GLDP-WS in *case* 4 incurs the least integrated cost, which improves the performance of FORTE from 7.5% to 29%, while GLDP-NS in *case* 4 outperforms FORTE from 4% to 22%. However, FORTE requires slightly less integrated cost than our proposed algorithms in *case* 1 and *case* 2. GLDP-WS and GLDP-NS care about only one of the two factors of the distance and the data center energy consumption in these two cases, and each data chunk is placed in a data center, which potentially increases the distance and the energy consumed by the network transport. GLDP-WS and GLDP-NS obtain the same results in both *case* 1 and *case* 2. The consideration of the placement status of the data center servers enables GLDP-WS to obtain better performance than GLDP-NS, due to the aggregation of data on a subset of servers.

4.2.2. Impact of the number of users

We then study the impact of the number of users on the performance of different algorithms, assuming the number of data chunks is set at 5000.

The simulation results in Fig. 8 show that in general the distance keeps stable with various number of users. When the number of data chunks is fixed, the growth of the number of users decreases the probability that each data chunk is accessed by each user. As a result of multiple data center placement of data with FORTE, data are accessed from the data centers close to the users. Therefore, FORTE leads to the least distance. GLDP-WS and GLDP-NS achieve the same results in both *case* 1 and *case* 2 because of the irrelevance of the data placement status of the servers in data centers. GLDP-WS and GLDP-NS perform the best in *case* 1, since the distance is the only objective of the data placement. *Case* 2 results in the worst distance performance, since *case* 2 only considers the energy consumption of the data centers and ignores the distance, and potentially places the data in the data centers that may be far from the users. *Case* 3 obtains better distance performance than *case* 4, since the increase of the distance from the users to the data centers potentially increases the energy consumed by the networks.

Fig. 9 illustrates the energy consumption of the servers also keeps steady because of the fixed number of data. FORTE consumes the most energy, since the multiple data center data placement requires more servers. The simulation results of GLDP-WS and GLDP-NS are the same in both *case* 1 and *case* 2, because the data place-

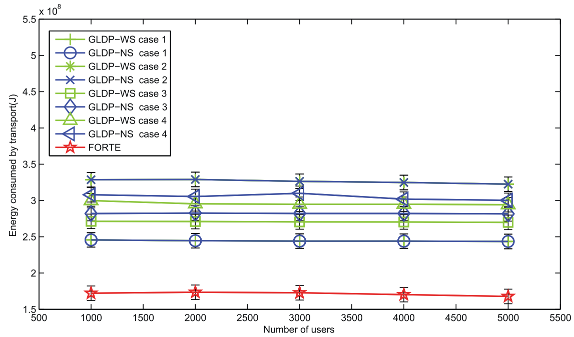

Fig. 10. Energy consumed by transport with the algorithms of GLDP-WS, GLDP-NS and FORTE as the increasing number of users.

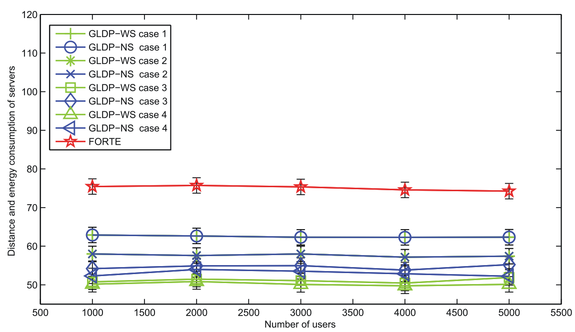

Fig. 11. Latency and energy consumption of servers with the algorithms of GLDP-WS, GLDP-NS and FORTE as the increasing number of users.

ment status of the servers in the data centers is not considered in these two cases. *Case* 2 achieves the best performance among the four cases, since the objective of *case* 2 is to reduce the data center energy consumption. On the contrary, *case* 1 results in the most energy consumption of the servers with the objective of minimizing the distance by placing the data in the data center located close to the users, while the data center has to place the data on the servers with high power or the data center has a big PUE. GLDP-WS and GLDP-NS perform better in *case* 4 than in *case* 3. *Case* 3 considers two factors of the distance and the data center energy consumption, while *case* 4 has to achieve the tradeoff among all the three factors. The distance from the users to the data centers has a negative impact on the energy consumed by the network transport. GLDP-WS outperforms GLDP-NS in both *case* 3 and *case* 4. With GLDP-WS, the data are potentially accommodated by a subset of the servers, which reduces the server resource used.

Fig. 10 shows the energy consumed by the network transport also keeps stable as the number of users increases, since the number of data chunks is fixed. FORTE results in the least energy consumed by the network transport, because data may traverse shorter paths in the network than with GLDP-WS and GLDP-NS. GLDP-WS and GLDP-NS achieve the same results in *case* 1 and *case* 2. GLDP-WS and GLDP-NS in *case* 1 achieve the best performance, because a shorter data transmission path may require less network energy consumption. GLDP-WS and GLDP-NS in *case* 3 obtain better results than *case* 4, also because the length of a shorter data transmission path has a positive impact on the network energy consumption performance.

The performance of the cost incorporating the distance and the energy consumption of data centers is depicted in Fig. 11. GLDP-WS and GLDP-NS achieve better performance than FORTE, and GLDP-WS in *case* 4 achieves the best performance. Both GLDP-NS and GLDP-WS in *case* 1 outperform FORTE about 16%, while the results of GLDP-WS in *case* 4 are better than those of FORTE about 33%. With FORTE, more copies of data require significantly more

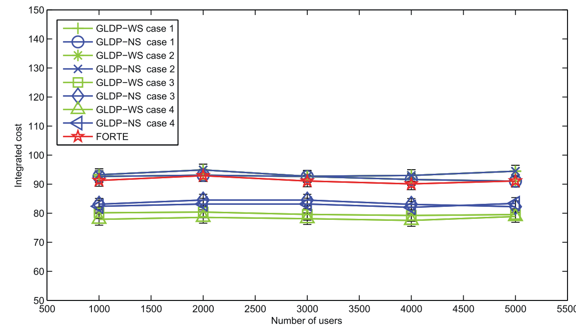

Fig. 12. Integrated cost with the algorithms of GLDP-WS, GLDP-NS and FORTE as the increasing number of users.

energy to power more working servers. GLDP-WS and GLDP-NS achieve the same results in *case* 1 and *case* 2. *Case* 1 only considers the distance and ignores the energy consumption of the servers, while *case* 2 only considers the energy consumption of the servers. The two cost factors have the same impact on the incorporated cost performance with GLDP-WS and GLDP-NS in *case* 1 and *case* 2. Therefore, the incorporated cost of the two factors is also the same. GLDP-WS performs better than GLDP-NS in *case* 3 and *case* 4. With the consideration of the data placement status of the servers in the data centers, the data are aggregated on some of the servers so as to reduce the number of servers used.

The performance in terms of the integrated cost of the distance, the energy consumed by data centers and the network transport is given in Fig. 12. The results of GLDP-WS and GLDP-NS in *case* 3 and *case* 4 are better than FORTE, since the reduce of the distance and the network energy consumption with FORTE cannot compensate the increase of the data center energy consumption. GLDP-WS in *case* 4 obtains the best results which are better than FORTE about 12%, while GLDP-NS in *case* 4 improves the performance of FORTE about 9%. GLDP-WS and GLDP-NS achieve the same results in *case* 1 and *case* 2, because the three cost factors with GLDP-WS and GLDP-NS in *case* 1 and *case* 2 are the same. Among the 4 cases, GLDP-WS in *case* 1 achieves the worst performance, which is slightly worse than FORTE about 4.5%. GLDP-WS outperforms GLDP-NS in *case* 3 and *case* 4, as the consideration of the data placement status of the servers aggregates the data on the servers which are already accommodating some other data, thereby reducing the number of working servers.

5. Conclusions

Large-scale Internet applications provide service to end users by routing service requests to geographically distributed data centers. Currently, the data centers and the network transport that power the applications consume significant worldwide electricity supply. At the same time, latency is an important concern for the end users. In this paper, we tackled the problem of green latency-aware data placement in the data centers. The objective was to reduce the energy consumed by network transport and the servers in the data centers, while reducing the access latency. We proposed two request-routing algorithms, GLDP-WS (Green Latency-aware Data Placement - With consideration of current data placement Status of the server) and GLDP-NS (Green Latency-aware Data Placement - No consideration of current data placement Status of the server), to strike a tradeoff among the three factors above during data placement. GLDP-NS is a 3-approximation algorithm for the data placement problem without consideration of the server data placement status. We also conducted experiments through simulations. Experimental results demonstrate that the proposed algorithms can achieve good integrated cost performance of the latency, the energy consumption of data centers and network transport.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China [61272540]; and Anhui Provincial Natural Science Foundation of China [1608085MF142].

References

[1] R. Miller, Data center knowledge, report: Google uses about 900,000 servers, http://tiny.cc/gservers, 2015.
[2] Akamai reports fourth quarter 2010 and full-year 2010 financial results, http://tiny.cc/afyu9, 2015.
[3] J.G. Koomey, Growth in Data Center Electricity Use 2005 to 2010 Estimating Total Power Consumption by Servers in the U. S. and the World, Technical Report, 2007.
[4] J. Baliga, R. Ayre, K. Hinton, W.V. Sorin, R.S. Tucker, Energy consumption in optical ip networks, J. lightwave technol. 13 (27) (2009) 2391–2403.
[5] T. Singh, P.K. Vara, Smart metering the clouds, in: Proceedings of 18th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2009), IEEE, Piscataway, NJ, 2009, pp. 66–71.
[6] A. Alimian, B. Nordman, D. Kharitonov, Network and telecom equipment-energy and performance assessment test procedure and measurement methodology, ECR Initiative, 2008.
[7] Ieee p802.3az energy efficient ethernet task force, http://www.ieee802.org/3/az/, 2015.
[8] J. Baliga, R.W.A. Ayre, K. Hinton, R.S. Tucker, Green cloud computing: balancing energy in processing, storage, and transport, Proc. IEEE 99 (2011) 149–167.
[9] P.X. Gao, A.R. Curtis, B. Wong, S. Keshav, ItÃíÃ¯s not easy being green, in: Proceedings of ACM SIGCOMM 2012, ACM, New York, Helsinki, Finland, 2012, pp. 211–222.
[10] Data center report to congress, Technical report, U.S. Environmental Protection Agency, 2007.
[11] J.G. Koomey, Growth in data center electricity use 2005 to 2010, A report by Analytics Press, completed at the request of The New York Times, 2011.
[12] J. Liu, F. Zhao, X. Liu, W. He, Challenges towards elastic power management in internet data centers, in: Proceedings of 29th International Conference on Distributed Computing Systems (ICDCS 2009), Montreal, Quebec, Canada, 2009, pp. 65–72.
[13] Y. Cho, N. Chang, Energy-aware clock-frequency assignment in microprocessors and memory devices for dynamic voltage scaling, IEEE Trans. Comput. Aid D 26 (6) (2007) 1030–1040.
[14] A. Varma, B. Ganesh, M. Sen, S.R. Choudhury, L. Srinivasan, B. Jacob, A control-theoretic approach to dynamic voltage scheduling, in: Proceedings of the 2003 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES 2003), ACM, New York, San Jose, California, USA, 2003, pp. 255–266.
[15] O. Sarood, L.V. Kale, A cool load balancer for parallel applications, in: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC 2011), ACM, New York, Seattle, Washington, USA, 2011. 21:1–21:11
[16] O. Sarood, E. Totoni, L.V. Kalë, Cool load balancing for high performance computing data centers, IEEE Trans. Comput. 61 (12) (2012) 1752–1764.
[17] T. Horvath, T. Abdelzaher, K. Skadron, X. Liu, Dynamic voltage scaling in multitier web servers with end-to-end delay control, IEEE Trans. Comput. 56 (4) (2007) 444–458.
[18] R. Nathuji, K. Schwan, Virtualpower: coordinated power management in virtualized enterprise systems, in: Proceedings of 21st ACM Symposium on Operating Systems Principles (SOSP 2007), ACM, New York, Stevenson, Washington, USA, 2007, pp. 265–278.
[19] N. Kim, J. Cho, E. Seo, Energy-credit scheduler: an energy-aware virtual machine scheduler for cloud systems, Future Gener. Comput. Syst. 32 (2014) 128–137.
[20] Z. Xiao, W. Song, Q. Chen, Dynamic resource allocation using virtual machines for cloud computing environment, IEEE Trans. Parallel Distrib. 24 (6) (2013) 1107–1117.
[21] G. Katsaros, J. Subirats, J.O. Fitó, J. Guitart, P. Gilet, D. Espling, A service framework for energy -aware monitoring and VM management in clouds, Future Gener. Comput. Syst. 29 (8) (2013) 2077–2091.
[22] G. Han, W. Que, G. Jia, L. Shu, An efficient virtual machine consolidation scheme for multimedia cloud computing, Sensors 16 (2) (2015) 246.
[23] M. Lin, A. Wierman, L.L.H. Andrew, L.L.H. Andrew, Dynamic rightsizing for power-proportional data centers, in: Proceedings of INFOCOM 2011, IEEE, Piscataway, NJ, ShangHai, China, 2011, pp. 1098–1106.
[24] D. Cavdar, F. Alagoz, A survey of research on greening data centers, in: Proceedings of Global Telecommunications Conference (GLOBECOM 2012), IEEE, Washington, DC, USA, Anaheim, California, USA, 2012, pp. 3237–3242.
[25] P. Wang, Y. Qia, X. Liu, Power-aware optimization for heterogeneous multi-tier clusters, J. Parallel Distrib. Comput. 74 (1) (2014) 2005–2015.
[26] J. Shuja, K. Bilal, S.A. Madani, S.U. Khan, Data center energy efficient resource scheduling, Cluster Comput. 17 (4) (2014) 1265–1277.
[27] M. Zotkiewicz, M. Guzek, D. Kliazovich, P. Bouvry, Minimum dependencies energy-efficient scheduling in data centers, IEEE T Parallel Distrib. (2016) 1–14, doi:10.1007/s00607-015-0481-6.
[28] S. Goyal, S. Bawa, B. Singh, Green service level agreement (gsla) framework for cloud computing, Computing (2016) 1–15, doi:10.1007/s00607-015-0481-6.
[29] M. Pickavet, W. Vereecken, S. Demeyer, P. Audenaert, B. Vermeulen, C. Develder, D. Colle, B. Dhoedt, P. Demeester, Worldwide energy needs for ict: the rise of power-aware networking, in: Proceedings of Second International Symposium on Advanced Networks and Telecommunication Systems (ANTS 2008), IEEE, Piscataway, NJ, Bombay, India, 2008, pp. 1–3.
[30] K.H. Jayant Baliga, R.S. Tucker, Energy consumption of the internet, in: Proceedings of International Conference on Optical Internet held jointly with the 32nd Australian Conference on Optical Fibre Technology (COIN-ACOFT 2007), Melbourne, Australia, 2007, pp. 1–3.
[31] W.V. Heddeghem, M.D. Groote, W. Vereecken, D. Colle, M. Pickavet, P. Demeester, Energy-efficiency in telecommunications networks: Link-by-link versus end-to-end grooming, in: Proceedings of 2010 14th Conference on Optical Network Design and Modeling (ONDM 2010), Kyoto, Japan, 2010, pp. 1–6.
[32] S. Aleksic, Analysis of power consumption in future high-capacity network nodes, J. Opt. Commun. Netw. 1 (2009) 245–258.
[33] M. Gupta, S. Singh, Greening of the internet, in: Proceedings of ACM SIGCOMM 2003, ACM, New York, Barcelona, Spain, 2003, pp. 19–26.
[34] K. Christensen, B. Nordman, R. Brown, Power management in networked devices, Computer 37 (8) (2004) 91–93.
[35] M. Allman, K. Christensen, B. Nordman, V. Paxson, Enabling an energy-efficient future internet through selectively connected end systems, in: Proceedings of Sixth Workshop on Hot Topics in Networks (HotNets-VI), ACM, New York, Atlanta, Georgia, USA, 2003, pp. 1–6.
[36] C. Gunaratne, K. Christensen, B. Nordman, Managing energy consumption costs in desktop pcs and lan switches with proxying, split tcp connections, and scaling of link speed, Int. J. Netw. Manag. 15 (2005) 297–310.
[37] G. Ananthanarayanan, R.H. Katz, Greening the switch, in: Proceedings of Workshop on Power Aware Computing and Systems (HotPower '08), USENIX, Berkeley, CA, USA, San Diego, CA, USA, 2008. 7–7.
[38] C. Gunaratne, K. Christensen, B. Nordman, S. Suen, Reducing the energy consumption of ethernet with adaptive link rate(alr), IEEE Trans. Comput. 57 (2008) 448–461.
[39] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, D. Wetherall, Reducing network energy consumption via sleeping and rate-adaptation, in: Proceedings of 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI '08), USENIX, Berkeley, CA, USA, San Francisco, California, USA, 2008, pp. 323–336.
[40] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, S. Wright, Power awareness in network design and routing, in: Proceedings of INFOCOM 2008, IEEE, Piscataway, NJ, Phoenix, Arizona, USA, 2008, pp. 457–465.
[41] E. Schurman, J. Brutlag, The user and business impact of server delays, additional bytes, and http chunking in web search, in: Presentation at the OÃíÃ¯Reilly Velocity Web Performance and Operations Conference, San Jose, California, USA, 2009.
[42] M. Lee, S. Goldberg, R.R. Kompella, G. Varghese, Fine-grained latency and loss measurements in the presence of reordering, in: Proceedings of ACM SIGMETRICS 2011, ACM, New York, San Jose, California, USA, 2011, pp. 329–340.
[43] S.M. Rumble, D. Ongaro, R. Stutsman, M. Rosenblum, J.K. Ousterhout, Its time for low latency, in: Proceedings of 13th Workshop on Hot Topics in Operating Systems (HotOS XIII), USENIX, Berkeley, CA, USA, Napa, CA, USA, 2011. 11–11
[44] Z. Wu, H.V. Madhyastha, Understanding the latency benefits of multi-Cloud webservice deployments, Comput. Commun. Rev. 43 (2) (2013) 13–20.
[45] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, B. Maggs, Cutting the electric bill for internet-scale systems, in: Proceedings of ACM SIGCOMM 2009, ACM, New York, Barcelona, Spain, 2009, pp. 123–134.
[46] L. Rao, X. Liu, L. Xie, W. Liu, Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment, in: Proceedings of INFOCOM 2010, IEEE, Piscataway, NJ, San Diego, California, USA, 2010, pp. 1–9.
[47] A. Amokrane, R. Langar, M.F. Zhani, R. Boutaba, G. Pujolle, Greenslater: on satisfying green slas in distributed clouds, IEEE Trans. Netw. Service Manag. 12 (3) (2015) 363–376.
[48] Z. Xu, W. Liang, Carbon-aware online control of geo-distributed cloud servicesoperational cost minimization of distributed data centers through the provision of fair request rate allocations while meeting different user slas, Comput. Netw. 83 (2015) 59–75.
[49] K.T. Le, R. Bianchini, T.D. Nguyen, O. Bilgir, M. Martonosi, Capping the brown energy consumption of internet services at low cost, in: Proceedings of 2010 International Green Computing Conference (IGCC 2010), IEEE, Piscataway, NJ, Chicago, Illinois, USA, 2010, pp. 3–14.
[50] Z. Liu, M. Lin, A. Wierman, S.H. Low, L.L.H. Andrew, Greening geographical load balancing, in: Proceedings of ACM SIGMETRICS 2011, ACM, New York, San Jose, California, USA, 2011, pp. 233–244.
[51] J. Doyle, D. O'Mahony, R. Shorten, Server selection for carbon emission control, in: Proceedings of 2nd ACM SIGCOMM Workshop on Green Networking, ACM, New York, Toronto, ON, Canada, 2011, pp. 1–6.
[52] Z. Zhou, F. Liu, R. Zou, J. Liu, H. Xu, H. Jin, Carbon-aware online control of geo-distributed cloud services, IEEE Trans. Parall Distrib. 12 (3) (2015) 1–14, doi:10.1109/TPDS.2015.2504978.

[53] P. Chanclou, S. Gosselin, J.F. Palacios, V.L. Alvarez, E. Zouganeli, Overview of the optical broadband access evolution: a joint article by operators in the ist network of excellence e-photon/one, IEEE Commun. Mag. 44 (2006) 29–35.
[54] J. Baliga, R. Ayre, W.V. Sorin, K. Hinton, R.S. Tucker, Energy consumption in access networks, in: Proceedings of 2009 Optical Fiber Communication Conference and Exhibition/National Fiber Optic Engineers Conference (OFC/NFOEC 2008), San Diego, California, USA, 2008, pp. 1–3.
[55] Spring-nextel applied research groupipmon project, http://ipmon.sprint.com, 2005.
[56] L. Ceuppens, Planning for energy efficiency-networking in numbers, in: Proceedings of 2009 Optical Fiber Communication Conference and Exhibition/National Fiber Optic Engineers Conference (OFC/NFOEC 2009), San Diego, California, USA, 2009, pp. 192–211.
[57] E.B. Desurvire, Capacity demand and technology challenges for lightwave systems in the next two decades, J. Lightwave Technol. 24 (2006) 4697–4710.
[58] C. Mobius, W. Dargie, A. Schill, Power consumption estimation models for processors, virtual machines, and servers, IEEE Trans. Parallel Distrib. 25 (6) (2014) 1600–1614.
[59] D. Economou, S. Rivoire, C. Kozyrakis, Full-system power analysis and modeling for server environments, in: Proceedings of 33rd Annual International Symposium on Computer Architecture Workshop on Modeling, Benchmarking and Simulation (MoBS), Boston, MA, 2006.
[60] R. Cohen, L. Katzir, D. Raz, An efficient approximation for the generalized assignment problem, Inf. Process Lett. 100 (2006) 162–166.
[61] Juniper data sheets, http://www.juniper.net, 2015,
[62] Cisco data sheets, http://www.cisco.com, 2015.
[63] Fujitsu data sheets, http://www.fujitsu.com, 2015.

Yuqi Fan is an associate professor and an assistant to the Dean at School of Computer and Information at Hefei University of Technology, China. He got his Ph.D. in Computer Science and Engineering at Wright State University in 2009. He received both B.S. and M.S. degrees in computer science and engineering from Hefei University of Technology in 1999 and 2003, respectively. His research interests include computer networks, SDN, cloud computing, cyber-phsical systems, and wavelength-division-multiplexing (WDM) network design and optimization.