Energy-Efficient and Latency-Aware Data Placement for Geo-Distributed Cloud Data Centers

Yuqi Fan(✉), Jie Chen, Lusheng Wang, and Zongze Cao

School of Computer and Information, Hefei University of Technology, Hefei, China
yuqi.fan@hfut.edu.cn

Abstract. Cloud computing technology achieves enormous scale by routing service requests from users to geographically distributed servers, typically located at different data centers. On one hand, energy consumption of data centers and networks has been receiving increasing attention in recent years. On the other hand, users require low latency during data access from data centers. In this paper, we tackle the problem of energy-efficient data placement in data centers, taking into account access latency, energy consumption of data centers and network transport. We propose two request-routing algorithms to determine the number of copies for each data chunk and the data centers accommodating the data chunk. Our simulation results have shown that the proposed algorithms are effective in terms of the tradeoff among the data access latency, the energy consumed by network transport and data centers.

Keywords: Energy-efficient · Latency · Energy consumption of servers · Energy consumption of network transport · Data placement

1 Introduction

Cloud computing technology achieves enormous scale by routing service requests from end users to a set of geographically distributed servers, typically located at different data centers. In order to reduce data access latency experienced by users, it is quite often to place the data in multiple data centers so that the users can access the data from nearby data centers. However, the data centers are large consumers of electricity, consuming about 1.3% of the worldwide electricity supply [1]. At the same time, a lot of energy needs to power the network equipments, which consume approximately 14.8% of the total ICT energy consumption [2].

There has been some work on reducing the delay, the electricity cost and consumption 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 [3]. [4] improves the algorithms in [3] on multi-region electricity markets to better capture the present electricity price situation. [5] proposes an adaptive operational cost optimization framework incorporating time-varying electricity

Q. Chen et al. (Eds.): ChinaCom 2016, Part II, LNICST 210, pp. 465–474, 2018.
DOI: 10.1007/978-3-319-66628-0_44

prices and dynamic user request rates. [6] considers the joint optimization problem of minimizing carbon emission and electricity cost. [7] 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. [9] 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. [10] proposes a request-routing scheme, FORTE, allowing operators to strive the tradeoff among electricity costs, access latency, and carbon emissions. Assuming each data chunk, i.e. each piece of data, is placed in only one data center, [11] proposes a request-routing scheme to strike the tradeoff among access latency, energy consumption of the data centers and the network transport during data placement.

In this paper, we tackle the data placement problem in geo-distributed cloud data centers, taking into account the access latency, the energy consumption of the servers in the data centers, and the energy consumed by network transport, assuming each data chunk can be placed in more than one data center. The main contribution of this work is two-fold: First, we investigate the data placement problem with the objective to strike the tradeoff among the three factors above. Second, we propose two efficient algorithms to determine the proper number of copies for each data chunk and the data centers accommodating the data chunk.

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

2 Problem Formulation

The network model that the data centers provide data services to the end users is similar to the one in [8,11], and 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).

$$\begin{aligned} e_I(u_i, dc_j) = 6(3\tfrac{P_{es}}{C_{es}} + \tfrac{P_{bg}}{C_{bg}} + \tfrac{P_g}{C_g} + 2\tfrac{P_{pe}}{C_{pe}}) \\ +2\tfrac{P_c}{C_c} h_c(u_i, dc_j) + \tfrac{P_w}{2C_w} h_c(u_i, dc_j) \end{aligned} \tag{1}$$

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. $h_c(u_i, dc_j)$ is the number of hops during the data transmission in the core network. We assume a server consumes 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 for various kind of servers at different loads is available [12]. The problem is formulated as follows.

Minimize:

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

Subject to:

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

$$\sum_{dc_j, s_m} rep(dc_j, s_m, d_k) \geq 1, \forall d_k \tag{4}$$

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

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

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

where $p(u_i \mid d_k)$ is the probability that a given request coming from user u_i is asking for data d_k, $s(d_k)$ 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, $rep(dc_j, s_m, d_k)$ indicates whether data d_k is placed in server s_m in data center dc_j, $rep(dc_j, s_m)$ indicates whether server s_m in data center dc_j has accommodated some data, $e_S(s_m, dc_j)$ 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 processing power of sever s_m in data center dc_j, and $C(s_m, dc_j)$ is the capacity of server s_m in data center dc_j.

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

3 Energy-Efficient Latency-Aware Data Deployment Algorithms

We propose an Energy-efficient Latency-aware Data Deployment algorithm (ELDD) for the problem. The algorithm shown in Algorithm 1 consists of two

Algorithm 1. *Algorithm ELDD*

Input: The large data segment set $D_{k'}^{l}$
Output: The set of working servers
for all $d_{k'}^{l}$ **do**
    Merge.
    Sort the data centers.
    Assume all the data centers have accommodated large data segment $d_{k'}^{l}$.
    Assign each user to the data center with the least cost that holds $d_{k'}^{l}$.
    for all dc_j **do**
        Evaluate the cost of turning off the server accommodating $d_{k'}^{l}$ in dc_j.
        Turn off the server if shutting down the server will lead to cost saving.
    end for
end for

stages: (1) The data chunks are merged into large data segments so that each data segment consumes nearly the full capacity of a server. (2) The proper servers are found to accommodate each large data segment $d_{k'}^{l}$.

A data chunk with high access probability is more likely to be placed in more than one data center to reduce the energy consumption of network transport and the access delay. The data chunks are sorted by the non-ascending order of the total access probability from all the users in algorithm ELDD. The algorithm proceeds iteratively using greedy strategy. Within each iteration, the algorithm performs procedure Merge to put multiple data chunks into a large data segment, under the constraint that the large data segment does not exceed the storage size of server. The large data segments are formed one by one. This procedure continues until all data chunks are put merged.

After obtaining the large data segment set with procedure Merge, algorithm ELDD searches for the proper servers to accommodate each large data segment $d_{k'}^{l}$. The basic rationale of algorithm ELDD is to iteratively turning off the servers. Initially, algorithm ELDD places each large data segment $d_{k'}^{l}$ in all the data centers. Therefore, all the users can access the required data from the closest data center to reduce the energy consumption of network transport and the access delay. The effect of turning off the server accommodating large data segment $d_{k'}^{l}$ in each data center is evaluated. The cost of placing data chunk d_k on server s_m in data center dc_j is calculated via Eq. (8). If a server possessing $d_{k'}^{l}$ is turned off, the users accessing $d_{k'}^{l}$ from the server have to acquire $d_{k'}^{l}$ from the next closest data center. The server will be shut down if the inactive server can reduce the placement cost. The procedure repeats for each large data segment set, until all the large data segments are placed into some server(s).

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

Theorem 1. *Assume the number of data centers and users are $\mathcal{D}$ and $\mathcal{U}$, respectively. The time complexity of algorithm ELDD is $O(\mathcal{D}\mathcal{U} + \mathcal{D}\log\mathcal{D})$.*

Proof: The time complexity of sorting the data centers is $O(\mathcal{D} \log \mathcal{D})$. Assume two arrays *Leastcost* and $NextLeastcost$, each with the length of $\mathcal{U}$. *Leastcost*$[\nu] = \omega$ denotes that the working data center with the least cost to accommodate the data required by user u_ν is the data center with the ω-th least cost for user u_ν. $NextLeastcost$ is similar to *Leastcost*, which is to store the level of the working data center with the next to the least placement cost to accommodate the data required by user u_ν. Initially, each user can access the data from the data center with the least cost, since each data chunk has a copy in all the data centers. Therefore, *Leastcost*$[\nu] = 1$ and $NextLeastcost[\nu] = 2$ for each ν. For each *Leastcost*$[\nu] = \omega$, we evaluate the cost of turning off server in the data center and assigning user u_ν to the data center with the next to the least placement cost, if possible. If it leads to cost saving by turning off the server in the data center, *Leastcost* and $NextLeastcost$ will be updated. ν increases from 1 in the range of $[1, \mathcal{D}]$, and the traverse of *Leastcost* and $NextLeastcost$ runs in $O(\mathcal{D}\mathcal{U})$ time. Therefore, the time complexity of algorithm ELDD is $O(\mathcal{D}\mathcal{U} + \mathcal{D} \log \mathcal{D})$. $\square$

Note that we can deal with the data centers in different orders while placing a data chunk in the data centers. We propose ordering method *ELDD-Standard* which sorts the servers in the data centers in a non-descending order of the average processing power of the servers. Another ordering criteria is defined via Eq. (9).

$$S_j = f_j - \sum_i max\{0, v_i - c_{i,j}\} \tag{9}$$

where f_j denotes the server energy consumption of data center dc_j, v_i is the integrated cost of the data access latency, energy consumption of the network transport and the data centers while placing the data in the closest working data center, and $c_{i,j}$ indicates the cost of the data access latency an energy consumption of network transport by assigning user u_i to data center dc_j. We propose sorting method *ELDD-Fast* which sorts the data centers in the non-descending order of S_j. For Simplicity, we call ELDD-Standard and ELDD-Fast as Standard and Fast, respectively.

4 Simulation

We evaluate the performance of the proposed algorithms Standard and Fast by comparing them with the algorithms FORTE [10] and GLDD [11]. 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 GLDD places each data chunk in a data center. Similar to GLDD, Standard and Fast strike a tradeoff among the factors considered in GLDD. However, a data chunk may be placed in one or more data centers with Standard and Fast, which is similar to FORTE.

We use geographical distance as an approximation for latency similar to [10,11]. The request for a data chunk from a user is random, and any request for a data chunk comes from one of the users. Each data center hosts 200 servers, each with the capacity of $2TB$ and the power of 500 W. The equipments used in the network are the same as [13]. The quantity of data chunks, the average distance between the users and the data centers, the PUE of the data centers, and the number of WDM and core routers are set as the same as [11].

4.1 Impact of Various Number of Data Chunks

In this subsection, we investigate the performance of Standard, Fast, GLDD and FORTE with regard to the distance, the energy consumed by the network transport and the servers in the data centers versus different number of data chunks, assuming the number of users is 1000.

Figure 1 demonstrates that in general the distance increases with the increase of the number of data chunks, which is also shown in Eqs. (2) and (8). 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. Each data chunk has only one copy with GLDD, and each user may not be able to access the data from the closest data center. Standard and Fast may place each data chunk in one or more data centers. However, the number of data copies with Standard and Fast may be potentially less than that with FORTE, since Standard and Fast also consider the factors of energy consumed by the network transport and the data centers. Therefore, FORTE leads to the least distance and GLDD results in the largest distance. Standard only considers the energy consumption of the data centers while evaluating the cost of turning off the servers. In contrast, Fast takes into account the energy consumption of the network transport and the data access latency, in addition to the energy consumption of the data centers. Therefore, Fast potentially places more copies of the data than Standard, which leads to less distance than Standard.

Figure 2 illustrates that the energy consumption of the servers in the data centers increases with the increase of number of data chunks, because more servers are needed to accommodate the data. FORTE consumes the most energy, since FORTE places more copies of the data. Each data chunk is placed in only one data center with GLDD, and hence GLDD requires the least energy. Fast potentially places more copies of the data than Standard, which makes Fast consume more energy than Standard.

Figure 3 shows that the energy consumed by network transport increases with the increasing number of data chunks, since more data transfer incurs more energy consumption in the network. FORTE results in the least energy consumed by the network transport. With FORTE, the data go through shorter distances between the data centers and the users than with GLDD, Standard and Fast, which potentially reduces the number of network devices needed for the data transfer as shown in Eq. (1). With GLDD, each data chunk is placed only in one data center. The data access has to experience largest distance, and hence requires the most number of network devices, which makes GLDD consume the most network transport energy consumption. Fast potentially leads to less distance and less network devices than Standard, and hence Standard results in more energy consumed by the networks.

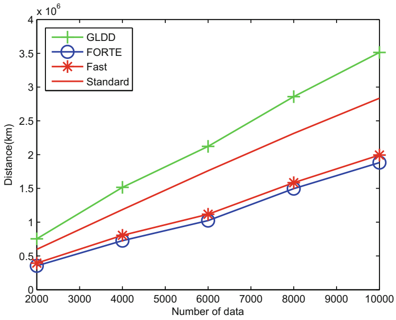

Fig. 1. Distance with the algorithms of Standard, Fast, GLDD and FORTE as the increasing number of data chunks.

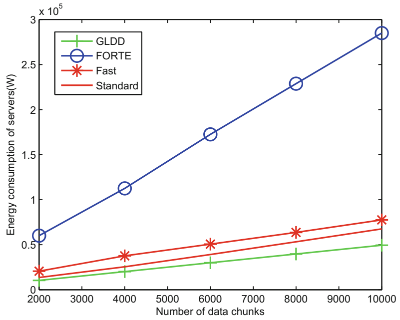

Fig. 2. Energy consumption of servers with the algorithms of Standard, Fast, GLDD and FORTE as the increasing number of data chunks.

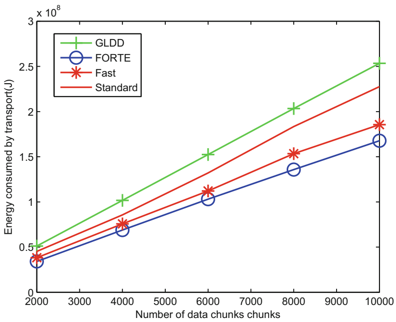

Fig. 3. Energy consumed by transport with the algorithms of Standard, Fast, GLDD and FORTE as the increasing number of data chunks.

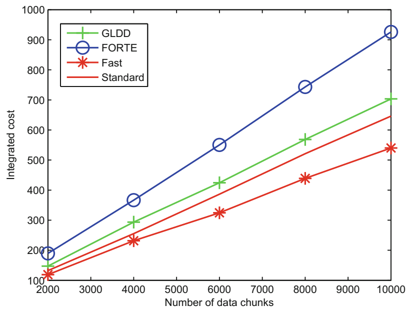

Fig. 4. Integrated cost with the algorithms of Standard, Fast, GLDD and FORTE as the increasing number of data chunks.

The performance in terms of the integrated cost of the distance, the energy consumed by the servers in the data centers and the network transport is depicted in Fig. 4. λ_1, λ_2 and λ_3 are all set as 1, so that all the three factors will have equal impact on the data placement decision. Standard, Fast and GLDD consider all the three factors of the data access latency, and the energy consumption incurred by the network transport and the data centers, while FORTE does not consider the energy consumption of the network transport. Therefore, Standard, Fast and GLDD achieve better results than FORTE.

4.2 Impact of Various Number of Users

In this subsection, we compare Standard and Fast, with GLDD and FORTE versus different number of users, assuming the number of data chunks is 5000.

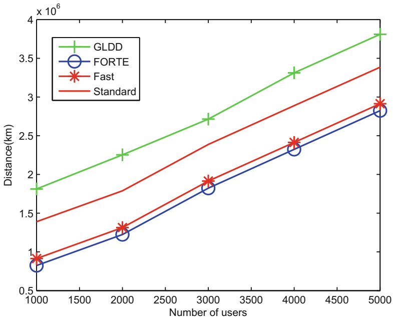

Fig. 5. Latency with the algorithms of Standard, Fast, GLDD and FORTE as the increasing number of users.

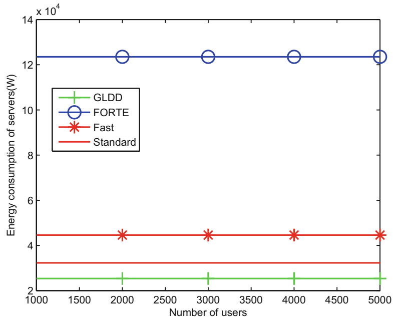

Fig. 6. Energy consumption of servers with the algorithms of Standard, Fast, GLDD and FORTE as the increasing number of users.

The simulation results in Fig. 5 show that in general the distance keeps stable with various number of users. When the number of data chunks is fixed, the increase of the number of users decreases the probability that each data chunk is accessed by each user. FORTE leads to the least distance and GLDD results in the largest distance. The number of copies with Standard and Fast may be potentially less than the number of copies with FORTE, and GLDD places each data chunk in only one data center. Standard only considers the energy consumption of the data centers while evaluating the cost of turning off the servers. In contrast, Fast takes into account the energy consumption of the network transport and the data access latency, in addition to the energy consumption of the data centers. Therefore, Fast potentially places more copies of the data than Standard, and results in less distance than Standard.

Figure 6 illustrates the energy consumption of the servers keeps steady because of the fixed number of data. FORTE consumes the most energy, since FORTE potentially creates the most number of data copies. GLDD places each data chunk in only one data center, and hence requires the least number of servers, which leads to the least server energy consumption. Standard achieves better performance than Fast, since Fast places the data in more data centers and thus requires more servers.

Figure 7 shows the energy consumed by the network transport increases with the growth of the number of users, since more users access the data through the network. FORTE achieves the best performance, because users can access the data from the closest data centers. GLDD consumes the most energy, as each data chunk is placed in only one data center so that the users go through largest distance to access the data. Fast outperforms Standard, since the users can access the data from the closer data centers with Fast than Standard.

The performance in terms of the integrated cost of the distance, the energy consumed by the servers in the data centers and the network transport is given in Fig. 8. By considering the three factors of the latency, the energy consumption

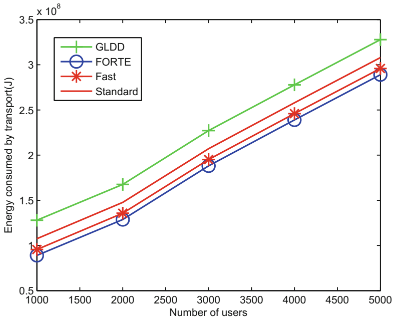

Fig. 7. Energy consumed by transport with the algorithms of Standard, Fast, GLDD and FORTE as the increasing number of users.

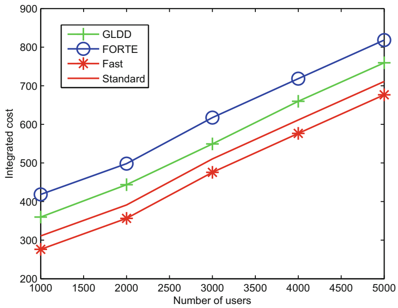

Fig. 8. Integrated cost with the algorithms of Standard, Fast, GLDD and FORTE as the increasing number of users.

of data centers and network transport, Standard and Fast outperforms FORTE and GLDD without the limitation of the number of data copies. Fast performs better than Standard because Fast potentially places the data in more data centers, and the decreased cost of network transport energy consumption can compensate the increased energy consumed by the data centers.

5 Conclusions

Cloud computing technology enables large-scale Internet applications to 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 electricity. At the same time, latency is also an important concern for the end users. In this paper, we tackled the problem of energy-efficient and latency-aware data placement in data centers. The objective was to reduce the energy consumed by network transport and data center servers, while reducing access latency. We proposed two efficient algorithms to determine the proper number of copies for each data chunk and the data centers accommodating the data chunk. Our simulation results have shown that the proposed algorithms are effective in terms of the tradeoff among the data access latency, the energy consumed by network transport and data centers.

Acknowledgments. The work reported in this paper was supported in part by Anhui Provincial Natural Science Foundation (1608085MF142).

References

1. Koomey, J.G.: Growth in data center electricity use 2005 To 2010. In: A Report by Analytics Press, Completed at the Request of The New York Times (2011)

2. Pickavet, M., Vereecken, W., Demeyer, S., Audenaert, P., Vermeulen, B., Develder, C., Colle, D., Dhoedt, B., Demeester, P.: Worldwide energy needs for ICT: the rise of power-aware networking. In: 2008 2nd International Symposium on Advanced Networks and Telecommunication Systems, Piscataway, NJ, Bombay, India, pp. 1–3, 15–17 December 2008
3. Qureshi, A., Weber, R., Balakrishnan, H., Guttag, J., Maggs, B.: Cutting the electric bill for internet-scale systems. In: 2009 ACM SIGCOMM conference on Data communication, New York, Barcelona, Spain, pp. 123–134, 17–21 August 2009
4. Rao, L., Liu, X., Xie, L., Liu, W.: Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment. In: 2010 IEEE International Conference on Computer Communications, Piscataway, NJ, San Diego, California, USA, pp. 1–9, 15–19 March 2010
5. Xu, Z., Liang, W.: Operational cost minimization of distributed data centers through the provision of fair request rate allocations while meeting different user SLAs. Comput. Netw. **83**, 59–75 (2015)
6. Le, K.T., Bianchini, R., Nguyen, T.D., Bilgir, O., Martonosi, M.: Capping the brown energy consumption of internet services at low cost. In: 2010 International Green Computing Conference, Piscataway, NJ, Chicago, Illinois, USA, pp. 3–14, 15–18 August 2010
7. Doyle, J., O'Mahony, D., Shorten, R.: Server selection for carbon emission control. In: 2nd ACM SIGCOMM Workshop on Green Networking, New York, Toronto, ON, Canada, pp. 1–6, 5–19 August 2011
8. Baliga, J., Ayre, R.W.A., Hinton, K., Tucker, R.S.: Green cloud computing balancing energy in processing, storage, and transport. Proc. IEEE **99**, 149–167 (2011)
9. Zhou, Z., Liu, F., Zou, R., Liu, J., Xu, H., Jin, H.: Carbon-aware online control of geo-distributed cloud services. IEEE Trans. Parallel Distrip. Syst. **12**(3), 1–14 (2015)
10. Gao, P.X., Curtis, A.R., Wong, B., Keshav, S.: It's not easy being green. In: 2012 ACM SIGCOMM Computer Communication Review, New York, Helsinki, Finland, pp. 211–222, 13–17 August 2012
11. Fan, Y., Ding, H., Hu, D.: Green latency-aware data deployment in data centers: balancing latency, energy in networks and servers. In: 2014 ACM SIGCOMM Workshop on Distributed Cloud Computing, New York, Chicago, USA, pp. 45–46, 18 August 2014
12. Mobius, C., Dargie, W., Schill, A.: Power consumption estimation models for processors, virtual machines, and servers. IEEE Trans. Parallel Distrip. Syst. **25**(6), 1600–1614 (2014)
13. Baliga, J., Ayre, R., Hinton, K., Sorin, W.V., Tucker, R.S.: Energy consumption in optical IP networks. J. Lightwave Technol. **13**(27), 2391–2403 (2009)