Socially-Responsible Load Scheduling Algorithms for Sustainable Data Centers over Smart Grid

Jian He†, Xiang Deng†, Dan Wu‡, Yonggang Wen§, Di Wu†
†Department of Computer Science, Sun Yat-Sen University, Guangzhou, China
‡Network & Information Branch, Guangdong Electric Power Design Institute, Guangzhou, China
§School of Computer Engineering, Nanyang Technological University, Singapore

***Abstract*—Power consumption of data centers accounts for a significant portion of operational cost for service providers. However, it is not sufficient to minimize electricity cost only. Instead, service providers should also consider the sustainability of data centers. In this paper, we investigate the problem of socially-responsible load scheduling for sustainable data centers in a smart grid environment. We formulate the joint optimization of electricity cost, performance cost and social cost into a min-cost network flow model, which takes the diversity of regional electricity prices, user request latency, usage of renewable energy into account. We further propose both offline and online load scheduling algorithms to achieve the optimality of the objective cost function. Finally, a series of simulation experiments are conducted to evaluate the effectiveness of our proposed algorithms.**

I. Introduction

In the era of cloud computing, more and more Internet service providers intend to deploy their services (e.g., e-commerce, content distribution, gaming, social networking) in geographically distributed data centers, which are reliable, elastic and cost-effective. For example, Akamai, a leading commercial CDN provider, distributes its service on hundreds of thousands of servers spreading over one thousand data centers in different regions around the world. Energy cost of such large-scale distributed infrastructure is of increasing concern to both operators and the whole society. It was reported that the power consumption of data centers accounts for 3% of the total U.S. power consumption in 2011, and the percentage is expected to reach 15% in the near future [1]. The growing energy consumption of data centers enforces us to take sustainability into consideration when building next-generation IDC-based services.

Various technologies and solutions [1], [2], [3] have been proposed recently to cope with the energy consumption problem of data centers. Most research work aims at reducing the amount of energy usage and cutting the electricity bill. In a smart grid environment, however, many power grid companies provide automatic electricity distribution with diverse pricing strategies that exhibit both temporal and geographic variations. Modern service providers such as Google exploit the features of smart grid to reduce electricity cost by deploying their data centers at different regions.

In addition to reducing energy usage, it is also of great importance to reduce the reliance on brown energy (e.g., fossil fuels) for sustainability. Future sustainable data centers are expected to be entirely powered by renewable energy such as solar energy and wind [4]. Actually, there have existed some work to study the powering of data centers with a portfolio of green energy [5]. However, the price of renewable energy is normally much higher than that of brown energy in the current stage. For service providers with social responsibility, it is challenging to achieve a good balance between reducing the electricity cost and improving the usage of renewable energy.

In this paper, we consider the problem of socially-responsible load scheduling for Internet data centers to minimize electricity cost, performance cost and social cost jointly under a smart grid environment. The social cost is determined by the extent of renewable energy usage. With more prevalence of renewable energy usage, the social cost will be decreased correspondingly. For sustainable data centers, social cost is an important factor that should be optimized. Our idea is to formulate the optimization problem into a min-cost network flow problem and propose both offline and online load scheduling algorithms to tackle it. The effectiveness of our proposed algorithm is validated by extensive experiments. In summary, our main contributions in this paper can be listed as follows:

- We consider the joint optimization of electricity cost, performance cost and social cost for sustainable data centers via load scheduling. To the best of our knowledge, it is the first work to optimize the above three factors for Internet-scale service providers.
- We adopt network flow theory to model the optimal load scheduling problem theoretically, and transform the optimization problem into a min-cost network flow problem that integrates different types of cost into one model.
- We propose both offline and online algorithms based on gradient method to obtain the optimal service load scheduling strategy, and conduct a set of simulation experiments to examine the effectiveness of our proposed algorithms. The simulation results show that our algorithm can effectively reduce the total system cost and promote the usage of renewable energy.

The rest of this paper is organized as follows. Section II discusses related work. Section III presents the formulation of the joint optimization problem. In Section IV, we solve the optimization problem by both offline and online algorithms. In Section V, we conduct a set of simulation experiments to validate the effectiveness of our algorithms. Section VI concludes this paper and discusses our future work.

978-1-4673-0911-0/12/$31.00

II. Related Work

In recent years, with the growing electricity demand of large-scale data centers, energy management of data centers have attracted lots of attention from industrial and academic fields.

Early work mostly focused on reducing energy consumption of single server machine by using CPU DVFS[6], low-power chipsets[7], advanced cooling technique[8], power control[9], etc. But the improvement of energy-efficient hardware design can not keep up with the paces of the development of large-scale distributed systems.

Another direction of research investigated how to save the total power consumption of a data center by dynamic server provisioning and load dispatching (such as [10], [11], [12], [13], [14]) according to the workload pattern. The basic idea is to turn on/off servers adaptively to meet demand changes. To this objective, it is necessary to predict the demand dynamics, activate enough number of servers beforehand, and carefully dispatch the workload to active servers. The use of visualization technique [15] can further save energy cost of data centers by consolidating multiple virtual machines to one physical server.

However, the above-mentioned research work doesn't consider the unique features of smart grid, in which electricity price exhibits both temporal and regional diversity. A number of papers (see [1], [2], [3], [5]) studied how to minimize the total electricity cost of a data center by exploiting price diversity in the smart grid. The idea is to perform energy-aware geographical load balancing, namely, to allocate service load to data centers with cheaper electricity prices. Factors such as workload pattern, latency requirement and device reliability are taken as constraints in their problem formulation. To better utilize the temporal variation of electricity prices, Wang et al. [16] and Urgaonkar et al. [17] considered the use of energy storage devices in data centers to shift demand peak away from high-price periods. Although electricity cost can be minimized, the amount of electricity usage may be increased due to cheaper prices[1]. To improve sustainability, it is also necessary to control the amount of electricity usage and increase the green extent of electricity.

Our paper differs from the previous work in that we consider energy management of data centers from a broader perspective. In addition to minimizing electricity cost, we also consider performance cost and social cost. We formulate the problem into a joint optimization problem with network flow theory. By using dynamic load scheduling for Internet-scale services, we are able to optimize the use of renewable energy without sacrificing user performance. Our work is essential for improving the sustainablity of future data centers in a smart grid environment.

III. Problem Formulation

We consider a simplified Internet-scale distributed system that consists of N data centers (as shown in Figure 1). Each data center hosts a number of servers and provides service to users spreading over M regions. For each data center, it is powered by an *Electricity Provider (EP)* in the same location. The electricity supplied to data centers can be generated by different types of energy, including solar, wind, coal, natural gas, hydroelectric, etc. We can use the green extent to evaluate the degree of renewable energy usage.

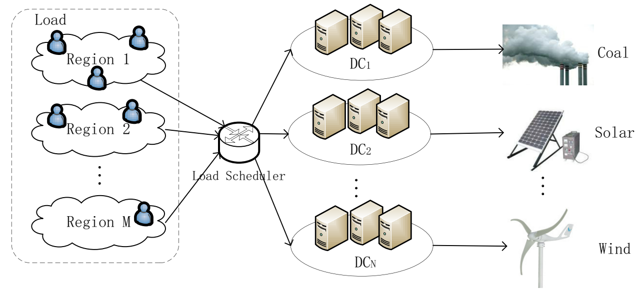

Fig. 1. A simplified Internet-scale distributed system

A user request is firstly routed to a load scheduler, which then selects an appropriate server in one data center to serve the user request. The selection is based on certain policies, such as geographic proximity, service performance, etc. Note that the load scheduler can be either centralized or distributed. For example, the load scheduler in a large CDN system is normally implemented by a distributed naming system. The total power consumption of a data center is determined by its service load level, which in turn depends on the scheduling algorithm of the load scheduler.

Our objective in this paper is three-fold: *(1) minimizing the electricity cost of data centers; (2) increasing the usage of renewable energy for sustainability; (3) improving user service quality in all regions*. We will develop a min-cost network flow model in which all the above three factors have been taken into account in the next section.

A. Network Flow Model

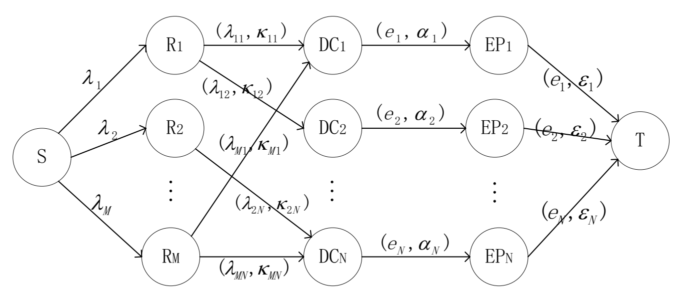

Fig. 2. Network flow model of for the energy management problem

Figure 2 depicts the network flow model for the load scheduling problem. In the figure, S and T are two virtual nodes that represent the source node and destination node. They are used to indicate the total service load and the total energy supply. The set of data centers are denoted by nodes $\{DC_j, j = 1, \cdots, N\}$, and the set of regions where user requests are generated are represented by nodes $\{R_i, i = 1, \cdots, M\}$. The set of electricity providers are represented by nodes $\{EP_k, k = 1, \cdots, N\}$. The flow λ_i over the edge between nodes S and R_i indicates the amount of service load generated by users in the i-th region, and the flow λ_{ij} over the edge between nodes R_i and DC_j refers to the amount of service load generated by users in the i-th region but distributed to the j-th data center. Note that $\lambda_i = \sum_{j=1}^{N} \lambda_{ij}$. The flow between nodes DC_j and EP_j is the amount of electricity used by the j-th data center.

In order to incorporate electricity cost, performance cost and social cost into the network flow model, we assign a cost to each edge in the graph. Electricity cost is defined as the product of the amount of used electricity and the electricity unit price. For performance cost, we only consider the latency incurred during serving user requests. We aim to minimize the average latency of all requests. Social cost is determined by the usage of renewable energy. The social cost is lower when more renewable energy is used to power data centers.

The associated cost for edges between DC_j and EP_j is the electricity unit price, denoted by α_j. For each edge between EP_j and T, the associated cost is denoted by ϵ_j, which is a factor to represent the "green" degree of energy supplied by EP_j. When more renewable energy is supplied by EP_j, the cost ϵ_j will be smaller. To take user performance into account, we also associate the edges between data centers and regions with a cost, which represents the latency to serve a request. For each edge between nodes R_i and DC_j, the associated latency cost is assumed to be determined by a non-decreasing convex function $\kappa_{ij}(\cdot)$. The convexity property of latency cost function follows the assumption made in [18] considering the increasing queueing delay when the amount of service load increases. The latency cost captures the revenue loss due to intolerable latency experienced by users. Other edges are associated with a cost 0. Notations used in this paper are summarized in Table I. Let EC, LC and SC be the electricity cost, latency cost and social cost respectively. Our objective in this paper is to minimize the total cost in the above model, namely, $EC + LC + SC$, where

TABLE I
NOTATION

Notation	Description
M	the number of user regions
N	the number of geographically distributed data centers
α_j	the electricity unit price of EP_j
ϵ_j	the green factor of energy supplied by EP_j
C_j	service capacity of data center j
λ_i	service load generated from region i
$\kappa_{ij}(\cdot)$	the latency cost between region i and data center j
γ	the service capacity of one server
S_j	the number of active servers in the j-th data center

$$\begin{aligned} EC &= \sum_{j=1}^{N} \alpha_j \cdot e_j, \\ LC &= \sum_{i=1}^{M} \sum_{j=1}^{N} \kappa_{ij}(\lambda_{ij}), \\ SC &= \sum_{j=1}^{N} \epsilon_j \cdot e_j. \end{aligned}$$

B. Optimization Problem

To formulate the optimization problem formally, we adopt the standard linear model used in [14], [19] to approximate the energy consumption of a server under a given service load. The linear energy consumption model is given by:

$$E^{server}(\nu) = E^{server}_{idle} + (E^{server}_{peak} - E^{server}_{idle}) \cdot \nu \quad (1)$$

where E^{server}_{idle} is the energy consumed by an idle server, and E^{server}_{peak} is the energy consumed by a fully utilized server under peak load. The load factor ν is the ratio between the actual load and the peak load, and satisfies $\nu \in [0, 1]$. For simplicity, an ideal server provisioning algorithm is assumed to be used by the service operator. It means that the operator can activate enough number of servers to handle the assigned load without overloading any server and shut down all idle servers in the meanwhile. Thus, all the servers in the data center only have two states: active or shutdown. All servers are assumed to have the same capacity γ, which has been normalized by the total service load. The service load of a data center is evenly distributed to active servers within it. Given the scheduled service load, the minimal number of servers needed to be activated can be determined immediately. The impacts of server state transition to power consumption and server hardware reliability are not considered to simplify our model. By denoting the number of active servers in the j-th data center as S_j, the energy consumption of the j-th data center is given by:

$$e_j = E^{DC}_j(\mu_j) = S_j \cdot E^{server}(\mu_j / S_j) \quad (2)$$

where $\mu_j = \sum_{i=1}^{M} \lambda_{ij}$ and $\mu_j \le S_j \gamma$.

Given the entire service load is known a priori, the optimal socially-responsible load scheduling problem can be formulated as the following optimization problem **P1**:

$$\begin{aligned} \boldsymbol{P1}: \min \quad & EC + LC + SC \quad (3) \\ & = \sum_{j=1}^{N} E^{DC}_j(\sum_{i=1}^{M} \lambda_{ij}) \cdot (\alpha_j + \epsilon_j) + \sum_{i=1}^{M} \sum_{j=1}^{N} \kappa_{ij}(\lambda_{ij}) \\ \text{s.t.} \quad & \sum_{i=1}^{M} \lambda_{ij} \le C_j, \forall j \quad (3a) \\ & \sum_{j=1}^{N} \lambda_{ij} = \lambda_i, \forall i \quad (3b) \\ & \lambda_{ij} \ge 0, \forall i, j \quad (3c) \end{aligned}$$

where the constraint (3a) indicates that service load should not exceed the capacity of a data center, namely, C_j. The constraint (3b) indicates that all service load generated from any region will be distributed to feasible data centers. The optimization problem needs to determine the scheduling strategy of service load.

IV. Optimal Socially-responsible Load Scheduling for Sustainable Data Centers

In this section, we investigate the design of socially-responsible load scheduling for sustainable data centers. First, we design an offline algorithm that provides the optimal cost reduction. Next, we propose a practical online algorithm for dynamic service demand.

A. Design of Optimal Offline Algorithm

As the capacity of the j-th data center is C_j, given the service load μ_j for the j-th data center, the minimal number of required active servers is $\frac{\mu_j}{\gamma}$. Thus we can rewrite the optimization problem **P1** into:

$$\begin{aligned} \min \quad f(\boldsymbol{\lambda}) \quad = \quad & \sum_{j=1}^{N} (E_{idle}^{server} \cdot \frac{\mu_j}{\gamma} + \\ & (E_{peak}^{server} - E_{idle}^{server}) \cdot \mu_j) \cdot (\alpha_j + \epsilon_j) \\ & + \sum_{i=1}^{M} \sum_{j=1}^{N} \kappa_{ij}(\lambda_{ij}) \end{aligned}$$

where $\boldsymbol{\lambda} = \{\lambda_{ij}, i \in [1, M], j \in [1, N]\}$. The above optimization problem can be solved by using Lagrangian multiplier and the Lagrangian is given as below:

$$\begin{aligned} L(\boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{\eta}) \quad = \quad & f(\boldsymbol{\lambda}) + \sum_{j=1}^{N} \theta_j \cdot (\sum_{i=1}^{M} \lambda_{ij} - C_j) + \\ & \sum_{i=1}^{M} \eta_i \cdot (\lambda_i - \sum_{j=1}^{N} \lambda_{ij}) \end{aligned}$$

where $\theta_j \geq 0$, $\lambda_{ij} \geq 0, \boldsymbol{\theta} = \{\theta_j, j \in [1, N]\}, \boldsymbol{\eta} = \{\eta_i, i \in [1, M]\}$. We can solve the problem **P1** by applying dual composition. The Lagrangian can be rewritten as follows:

$$\begin{aligned} L(\boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{\eta}) \quad = \quad & \sum_{i=1}^{M} \{ \sum_{j=1}^{N} (g_j + \theta_j - \eta_i) \cdot \lambda_{ij} + \kappa_{ij}(\lambda_{ij}) \} \\ & - \sum_{j=1}^{N} \theta_j \cdot C_j + \sum_{i=1}^{M} \eta_i \cdot \lambda_i \end{aligned}$$

where $g_j = (E_{idle}^{server} \cdot \frac{1}{\gamma} + (E_{peak}^{server} - E_{idle}^{server})) \cdot (\alpha_j + \epsilon_j)$. Define $L_i(\boldsymbol{\lambda_i}, \boldsymbol{\theta}, \eta_i)$ as the i-th Lagrangian multiplier:

$$L_i(\boldsymbol{\lambda_i}, \boldsymbol{\theta}, \eta_i) \quad = \quad \sum_{j=1}^{N} (g_j + \theta_j - \eta_i) \cdot \lambda_{ij} + \kappa_{ij}(\lambda_{ij})$$

where $\boldsymbol{\lambda_i} = \{\lambda_{ij}, j \in [1, N]\}$. For given $\boldsymbol{\theta}, \eta_i$, $\boldsymbol{\lambda_i^*}$ is defined as follows:

$$\boldsymbol{\lambda_i^*} = \arg \min_{\boldsymbol{\lambda_i} \geq \mathbf{0}} [L_i(\boldsymbol{\lambda_i}, \boldsymbol{\theta}, \eta_i)], \quad \forall i$$

The dual problem is given by

$$\begin{aligned} \max \quad & h(\boldsymbol{\theta}, \boldsymbol{\eta}) = \sum_{i=1}^{M} h_i(\boldsymbol{\lambda_i}, \boldsymbol{\theta}, \eta_i) \\ & - \sum_{j=1}^{N} \theta_j \cdot C_j + \sum_{i=1}^{M} \eta_i \cdot \lambda_i \\ \text{s.t.} \quad & \boldsymbol{\theta} \geq \mathbf{0} \end{aligned}$$

where $h_i(\boldsymbol{\lambda_i}, \boldsymbol{\theta}, \eta_i) = L_i(\boldsymbol{\lambda_i^*}, \boldsymbol{\theta}, \eta_i)$. From the convexity of latency cost function, we have

$$\begin{cases} \lambda_{ij}^* = 0, \text{if } \kappa_{ij}'(\lambda_{ij}) + (g_j + \theta_j - \eta_i) > 0, \forall \lambda_{ij} \geq 0 \\ \kappa_{ij}'(\lambda_{ij}^*) + (g_j + \theta_j - \eta_i) = 0, \text{otherwise} \end{cases}$$

then optimal results can be obtained by applying gradient method with the following updates:

$$\theta_j(t+1) \quad = \quad [\theta_j(t) - \varphi \cdot (\sum_{i=1}^{M} \lambda_{ij} - C_j)]^+ \tag{4}$$

$$\eta_i(t+1) \quad = \quad \eta_i(t) - \omega \cdot (\lambda_i - \sum_{j=1}^{N} \lambda_{ij}) \tag{5}$$

where φ and ω are sufficiently small positive step-sizes. The offline algorithm to solve Problem **P1** is provided in **Algorithm 1**. The algorithm generates the optimal service load scheduling strategy $\boldsymbol{\lambda}$.

Algorithm 1 Offline Algorithm for Optimal Load Scheduling

Input:
$C_j, \forall j$; $\lambda_i, \forall i$;
$E_{idle}^{server}, E_{peak}^{server}$;
γ; $\alpha_j, \epsilon_j, \forall j$; $\kappa_{ij}, \forall i, j$.

Output:
Optimal service load scheduling strategy λ_{ij};
Optimal total cost $f(\boldsymbol{\lambda})$.

1: Initialization step: set $\theta_j \geq 0$, and η_i be a random value.
2: Each data center j updates θ_j according to equation (4) and broadcasts the new value to each region.
3: Each region i updates η_i according to equation (5) and broadcasts the new value to each data center.
4: Set $t \leftarrow t+1$ and go to step 2(until satisfying termination criterion).
5: **Return:** λ_{ij} and $f(\boldsymbol{\lambda})$.

B. Design of Online Optimization Algorithm

We further consider a realistic setting in which service load from all regions are time-varying and electricity price in the smart grid is also dynamic. Generally, there are two types of pricing strategies considering timing difference: *real-time pricing* and *day-ahead pricing*. In our algorithm design, we consider a day-ahead pricing strategy in which the electricity provider announces electricity prices for the following one day. The prices can be different for different time slots (e.g., hour) across the day.

Because the service load from each region is dynamic, we first need to predict the service load in the next time slot in order to optimize load scheduling. Similar to [20], we assume that the total service load generated by users in each region can be predicted. Let $\overline{\lambda}_i$ denote the random variable which represents the actual total service load generated from the i-th region. Denote the mean and variance of $\overline{\lambda}_i$ by Λ_i and σ_i^2 respectively. For convenience, let $\overline{\boldsymbol{\lambda}} = [\overline{\lambda}_1, \cdots, \overline{\lambda_M}]$, $\boldsymbol{\Lambda} = [\Lambda_1, \cdots, \Lambda_M]$ and $\boldsymbol{\sigma} = [\sigma_1, \cdots, \sigma_M]$. Assume that $\overline{\lambda}_i$ follows Gaussian distribution $\overline{\lambda}_i \sim \boldsymbol{N}(\Lambda_i, \sigma_i)$ (note that our method can be easily extended to other distributions). To reduce the probability of under-provisioning, the predicted service load input λ_i should satisfy that the actual service load can be met with high probability:

$$P(\lambda_i > \overline{\lambda}_i) \leq \delta$$

where δ is a small constant. Because $\overline{\lambda}_i$ follows a Gaussian distribution, we have

$$\lambda_i \geq \boldsymbol{E}[\overline{\lambda}_i] + \rho\sqrt{\boldsymbol{var}[\overline{\lambda}_i]} = \Lambda_i + \rho\sigma_i$$

where $\rho = F^{-1}(1-\delta)$. To further reduce the probability of under-provisioning, we can reserve an additional fraction of service capacity of each data center. It is also feasible to turn on additional δ percentage of servers to meet the demand.

Other parameters for load scheduling are not changed frequently. With the predicted load as the input, we can then divide the online optimization problem into a set of one-shot optimization problems. The online algorithm for optimal load scheduling is described in **Algorithm 2**.

Algorithm 2 Online Algorithm for Optimal Load Scheduling

Input:

$C_j, \forall j$;
$E_{idle}^{server}, E_{peak}^{server}$;
Λ_i and σ_i for random variable $\overline{\lambda_i}$;
γ; $\alpha_j, \epsilon_j, \forall j$; $\kappa_{ij}, \forall i, j$.

Output:

Optimal service load scheduling strategy λ_{ij};
Optimal total cost $f_\delta(\boldsymbol{\lambda})$.

1: Initialization step: set $\theta_j \geq 0$, and η_i be a random value.
2: Compute the predicted service load $\lambda_i = \Lambda_i + \rho\sigma_i$.
3: Set $C_j = C_j(1-\delta), \forall j$.
4: Each data center j updates θ_j according to equation (4) and broadcasts the new value to each region.
5: Each region i updates η_i according to equation (5) and broadcasts the new value to each data center.
6: Set $t \leftarrow t+1$ and go to step 4(until satisfying termination criterion).
7: Set the provisioning active servers in data center j as $\lceil \frac{\sum_{i=1}^{M} \lambda_{ij}}{\gamma} \rceil \cdot (1+\delta)$.
8: **Return:** λ_{ij} and $f_\delta(\boldsymbol{\lambda})$(replacing $\frac{\mu_j}{\gamma}$ with $\frac{\mu_j}{\gamma}(1+\delta)$).

V. Simulation Experiments

In this section, we conduct a series of simulation experiments to evaluate the effectiveness of our proposed algorithms.

A. Experiment Settings

To make our experiment more realistic, we use the regional electricity prices obtained from the U.S. electricity market [21] to drive our simulation. In the experiment, the time is divided into multiple slots, but we make no assumption on the scale of time slots. In the default settings, there are twenty user regions and nine operational data centers. As the latency consists of networking and queueing delay, the latency cost function exhibits convex property. Similar to [18], we define the latency cost function between a data center and a region as $\kappa(x) = \frac{d}{1000} + \frac{1}{C-x}$, where d is the geographical distance, x is the total service load scheduled to a data center, C is the maximum service load capacity. The constant 1000 is a scalar to represent the increasing rate of latency. Note that the latency cost function can be replaced by any other convex function. To simplify our experiment, we assume that geographical distance $d \sim U(1, 100)$ and social cost $\epsilon \sim U(1, 10)$.

We adopt the method used in [22] to generate the service load in each region. The regions are divided into three classes, which represent user population at different scales. The number of regions in each class and their corresponding parameters are given as follows:

- Class I: 10 regions, with mean $\Lambda_i \sim U(1000, 5000)$ and variance $\sigma_i \sim U(100, 200)$;
- Class II: 5 regions, with mean $\Lambda_i \sim U(5000, 10000)$ and variance $\sigma_i \sim U(100, 500)$;
- Class III: 5 regions, with mean $\Lambda_i \sim U(10000, 15000)$ and variance $\sigma_i \sim U(100, 1000)$;

For parameters of server energy model, we set $E_{idle}^{server} = 60$ and $E_{peak}^{server} = 100$ according to [14]. The service capacity of a server γ is 10, and the capacity of any data center is 20000. The parameter δ used in the prediction technique is 0.02.

For comparison, we also consider another service load scheduling strategy, called *"Cheap-First Strategy"*, in which user requests from a region are always first served by the data center with the cheapest electricity cost.

B. Experiment Results

In the experiments, we compare the efficiency of our proposed offline algorithm (**Algorithm 1**), online algorithm (**Algorithm 2**) and cheap-first strategy.

Fig. 3 illustrates the normalized total cost for different load scheduling strategies. The offline algorithm achieves the optimality in all time slots. The efficiency ratio of the online algorithm is around 1.1 compared with the offline algorithm. In addition, The offline algorithm can reduce $16\% \sim 26\%$ of the total cost compared with the Cheap-First strategy.

To analyze different costs in the energy management problem, we calculate electricity cost and social cost separately. Fig. 4 illustrates the normalized electricity cost for different load scheduling strategies. As expected, the cheap-first strategy achieves the least electricity cost. The online algorithm incurs

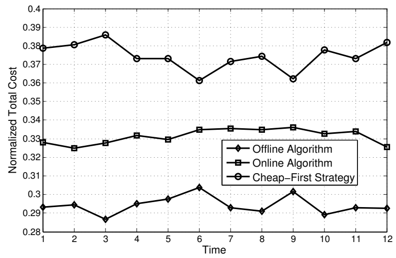

Fig. 3. Total cost under different load scheduling strategies

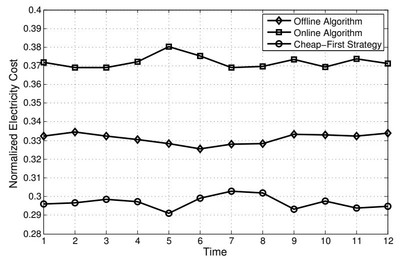

Fig. 4. Electricity cost EC under different load scheduling strategies

more electricity cost as we reserve additional δ percentage of data center capacity and increase δ percentage of active servers to mitigate service load violation, while the efficiency ratio is only $1.09 \sim 1.15$.

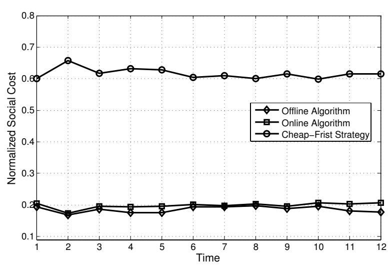

Fig. 5. Social cost SC under different load scheduling strategies

Fig. 5 illustrates the normalized social cost SC under different load scheduling strategies. Social cost is a measure to evaluate the usage of renewable energy. Our proposed algorithms can achieve near-optimality compared with the cheap-first strategy. The reduction of social cost is about 75%. The reason of the high reduction is that the unit social cost of the two EPs with the lowest electricity unit prices is as high as 9.

VI. Conclusion

In this paper, we investigate the optimal load scheduling problem which aims at minimizing electricity cost, latency cost and social cost jointly. The energy management problem can be modeled by network flow theory and transformed into a min-cost network flow problem. By applying dual decomposition, we design offline and online algorithms to achieve the optimality. Our simulation results confirm the effectiveness of our proposed algorithms. After this seminal work on the sustainability of data centers, we plan to consider the combination of reliable server provisioning that addresses hardware reliability and geographical load scheduling in optimizing the renewable energy usage in our future work. It is also of interest to study how to incentivize service providers to adopt a more green and sustainable design.

Acknowledgement

This work has been supported by NSFC (61003242), Program for New Century Excellent Talents in University (NCET-11-0542), the Fundamental Research Funds for the Central Universities (12LGPY53), and Guangzhou Pearl River Science and Technology Rising Star Project (No. 2011J2200086).

References

[1] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the Electric Bill for Internet-Scale Systems. In SIGCOMM,2009.

[2] L. Rao, X. Liu, L. Xie, and W. Liu. Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-Electricity-Market Environment. In INFOCOM, 2010.

[3] J. Li, Z. Li, K. Ren, and X. Liu. Towards Optimal Electric Demand Management for Internet Data Centers. In IEEE TRANSACTIONS ON SMART GRID, 2012.

[4] D. Gmach, Y. Chen, A. Shah, J. Rolia, C. Bash, T. Christian, and R. Sharma. Profiling Sustainability of Data Centers. In ISSST, 2010.

[5] Z. Liu, M. Lin, A. Wierman, S. Low, and L.H. Andrew. Greening Geographical Load Balancing. In SIGMETRICS, 2011.

[6] M. Weiser, B. Welch, A. Demers, and S. Shenker. Scheduling for Reduced CPU Energy. In OSDI, 1994.

[7] L.A. Barroso and U. Holzle. The Case for Energy Proportional Computing, Computer, vol. 40, no. 12, pp. 33C 37, 2007.

[8] C.D. Patel, C.E. Bash, R. Sharma, and M. Beitelmal. Smart Cooling of Data Centers. In Proceedings of IPACK, 2003.

[9] X. Liu, J. Heo, L. Sha, and X. Zhu. Adaptive Control of Multi-Tiered Web Application Using Queuing Predictor. In NOMS, 2006.

[10] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao. Energy-Aware Server Provisioning and Load Dispatching for Connection-Intesive Internet Services.In NSDI, 2008.

[11] S. Nedevschi, L. Popal, G. Iannaccone, S. Ratnasamy, and D. Wetherall. Reducing Network Energy Consumption via Sleeping and Rate-Adaptation. In NSDI, 2008.

[12] T. Horvath, T. Abdelzaher, K. Skadron, and X. Liu. Dynamic voltage scaling in multitier Web servers with end-to-end delay control. In IEEE Transactions on Computers, 2007.

[13] M. Lin, A. wierman, L.L.H. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In INFOCOM, 2011.

[14] V. Mathew, R. K. Sitaraman, and P. Shenoy. Energy-Aware Load Balancing in Content Delivery Networks. In INFOCOM,2012.

[15] R. Nathuji, and K. Schwan. VirtualPower: Coordinated Power Management in Virtualized Enterprise Systems.In ACM SIGOPS,2007.

[16] D. Wang, C. Ren, A. Sivasubramaniam, B. Urgaonkar, and H. Fathy. Energy Storage in Datacenters: What, Where, and How much? In SIGMETRICS, 2012.

[17] R. Urgaonkar, B. Urgaonkar, M. J. Neely, and A. Sivasubramaniam. Optimal Power Cost Management Using Stored Energy in Data Centers. In SIGMETRICS, 2011.

[18] W. Jiang, R. Shen, J. Rexford, and M. Chiang. Cooperative Content Distribution and Traffic Engineering in an ISP Network. In SIGMETRICS, 2009.

[19] Y.C. Jin, Y.G. Wen and Q.H. Chen. Energy Efficiency and Server Virtualization in Data Centers: An Empirical Investigation. In 1st IEEE INFOCOM Workshop on Communications and Control for Sustainable Energy Systems: Green Networking and Smart Grids,2012.

[20] D. Niu, Z. Liu, B. Li, and S. Zhao. Demand forecast and performance prediction in peer-assisted on-demand streaming systems. In INFOCOM, 2011.

[21] United States Regional Electricity Prices,available online at www.eia.doe.gov.

[22] D. Niu, H. Xu, B. Li, and S. Zhao. Quality-assured cloud bandwidth auto-scaling for video-on- demand applications. In INFOCOM, 2012.