# Optimal Power and Workload Management for Green Data Centers with Thermal Storage

Yuanxiong Guo*, Yanmin Gong*, Yuguang Fang*, Pramod P. Khargonekar* and Xiaojun Geng†
* Dept. of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA
† Dept. of Electrical and Computer Engineering, University of West Florida, Pensacola, FL 32514 USA
E-mail: {guoyuanxiong@, ymgong@, fang@ece., ppk@}ufl.edu, xgeng@uwf.edu

***Abstract*—Reducing the carbon footprint of data centers is becoming a primary goal of large IT companies. Due to the intermittency and unpredictability of renewable energy sources such as wind and solar, it is quite challenging to utilize them in data centers. In this paper, we explore the opportunities offered by delay-tolerant workloads and thermal storage to facilitate the renewable energy integration in data centers and meanwhile, reduce the cost of using brown energy (i.e., energy from the utility grid). A stochastic optimization problem is formulated to tackle the stochastic renewable generation and workload arrival processes. Then, an online control algorithm based on the Lyapunov optimization approach is proposed to solve it. Simulation results based on the real-world traces show the effectiveness of the algorithm in practice.**

***Index Terms*—Renewable energy, data center, thermal storage, Lyapunov optimization, batch workloads**

## I. Introduction

With the growing demand for large-scale computing resources, cloud computing is becoming very popular with different kinds of cloud services ranging from infrastructure-as-a-service, platform-as-a-service, to software-as-a-service. As the key underpinning for supporting these services, more and more new data centers are envisioned to be built in the near future. These data centers consume a huge amount of energy for their IT equipments and cooling infrastructures. Meanwhile, large IT companies pay more and more attention to their carbon footprints and propose to power their data centers with on-site renewable generation [1]. However, unlike conventional energy, renewable energy sources such as wind and solar are non-dispatchable and their availability depends on the ambient environment.

Data centers usually support a wide range of IT workloads, including both delay-sensitive, interactive applications and delay-tolerant, batch applications. Examples of delay-sensitive, interactive applications include web browsing, instant message, and searching. For the delay-tolerant, batch applications, typical examples are these jobs that can work well with the Amazon EC2 Spot Instances [2], such as background compression, elastic MapReduce, and scientific simulation. The flexibility of the delay-tolerant, batch workloads can be exploited to improve the renewable energy utilization by delaying their running to periods with abundant renewable generation.

Another opportunity is the thermal storage facility in a data center. Note that a large portion of data center power consumption comes from the cooling infrastructure. While large-scale electric energy storage, such as batteries, is still very expensive, thermal storage is much cheaper and can be leveraged to help integrate renewable energy. In fact, Apple has already deployed a chilled water storage facility in its green data center in Maiden, NC [1]. Excess renewable generation can be stored into the thermal storage for cooling purpose later.

Renewable-powered data centers are receiving more and more attention both in industry [1], [3] and in academia [4]–[7]. Previous studies [4], [5] explore the feasibility and benefits of using geographical load balancing for delay-sensitive interactive workloads to facilitate the integration of renewable sources into data centers. Scheduling of delay-tolerant batch workload and energy storage to help integrate renewable sources into a data center with on-site renewable generation is discussed in [6], [7]. However, all the aforementioned studies either assume perfect future information of renewable generation or workload arrivals [4]–[6], or do not consider the thermal storage [7].

In this paper, we exploit the flexibility provided by the above two components to help integration of renewable energy into data centers. Furthermore, considering that the electricity price in wholesale electricity market is time-varying, we also minimize the cost of using brown energy from the electric grid. Considering the randomness of the renewable generation and workload arrival processes, we first formulate the problem as a stochastic optimization problem. Then, based on the Lyapunov optimization techniques [8], we design an online algorithm to solve it approximately. Numerical simulations based on real-world traces are done to illustrate the effectiveness of our algorithm in practice. As far as we know, we are the first one to explore the opportunities offered by delay-tolerant workloads and thermal storage to help integrate renewable energy in a data center without assuming perfect prediction of future information.

The reminder of this paper is organized as follows. Section II describes the models we use and the problem formulation. An online algorithm is proposed in Section III to solve the problem. Analytical and numerical results are presented in

This work was supported in part by the U.S. National Science Foundation under grants CNS-1147813, ECCS-1129061, ECCS-1129062, and Eric Professor endowment at the University of Florida.

The work of Dr. Fang was partially supported by the National Natural Science Foundation of China under Grant 61003300.

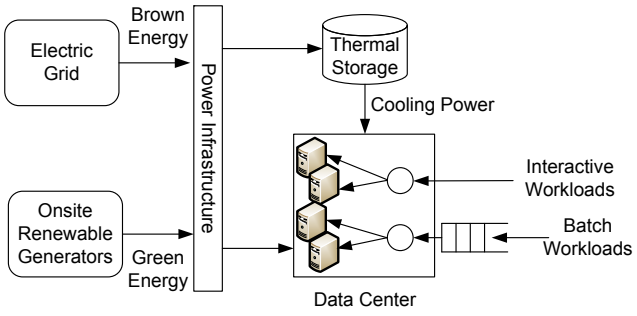



Fig. 1. Block diagram for system model

Section IV and Section V, respectively. Finally, conclusions are drawn in Section VI.

## II. Modeling and Optimization

We now describe the models we use in this paper to minimize the time-average brown energy cost in a data center. Assume the system is discrete-time. We consider a data center with on-site renewable generators (such as PV panels and wind turbines) and a thermal storage facility. Both interactive workloads and batch workloads would arrive at the data center for processing. The system we consider in this paper is depicted in Fig. 1, which is explained in detail as follows.

### A. The Workload Model

There are many different workloads in data centers. In general, they can be divided into the following two categories: delay-senstive, interactive applications and delay-tolerant, batch applications [6]. The interactive workloads usually process real-time user requests, which have to be completed within a certain time, i.e., there is a maximum response time. Delay-tolerant, batch jobs are often delay-tolerant, which can be scheduled to run at any time as long as the jobs are finished before the deadline, i.e., there is a maximum completion time.

In every period $t$, interactive workloads and batch workloads arrive at the data center. The average arrival rates of interactive and batch workloads during period $t$ are denoted as $\lambda_1(t)$ and $\lambda_2(t)$, respectively. A resource allocator in the data center would allocate appropriate servers to serve them. We assume the boundness on the arrival rates:

$$0 \le \lambda_1(t) \le \lambda_1^{max}, \quad 0 \le \lambda_2(t) \le \lambda_2^{max}. \tag{1}$$

As we have mentioned before, interactive workloads usually have a QoS requirement. As [4], we adopt the average queueing delay as the QoS metric and use the $M/GI/1/PS$ queuing model to analyze it. Suppose there are $n_1(t)$ active servers, each with service rate $\mu_1(t)$, allocated to serve the interactive workloads during period $t$. We have the following QoS constraints for interactive workloads:

$$\frac{1}{\mu_1 - \lambda_1(t)/n_1(t)} \le d_{max}, \tag{2}$$

$$n_1(t) \ge 0, \quad n_1(t) \in \mathbb{N}. \tag{3}$$

In contrast, the batch workloads can be buffered and served later. Denote the amount of buffered batch workloads at the end of period $t$ as $Q(t)$. Assume that $n_2(t)$ active servers, each with service rate $\mu_2$, are allocated to serve the batch workloads. Then, the dynamics of $Q(t)$ is as follows:

$$Q(t+1) = \max\{Q(t) - n_2(t)\mu_2, 0\} + \lambda_2(t), \tag{4}$$

$$n_2(t) \ge 0, \quad n_2(t) \in \mathbb{N}. \tag{5}$$

We need to ensure finite average delay for these buffered workloads. Therefore, we have the following QoS requirement for the batch workloads:

$$\overline{Q} < \infty, \tag{6}$$

where $\overline{Q}$ is the time average expected queue backlog for the batch workloads and is defined as:

$$\overline{Q} \triangleq \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{Q(t)\}. \tag{7}$$

Denote the total number of servers in the data center as $N$. Then, we have

$$n_1(t) + n_2(t) \le N, \quad \forall t. \tag{8}$$

### B. The Renewable Model

Motivated by the recent industry practice [1], we assume a data center would build on-site renewable generators near its location, which can reduce the transmission and distribution losses with the energy delivery. Denote the amount of on-site renewable energy generated at the data center as $r(t)$. Since renewable sources, mainly solar or wind, are variable, they may vary a lot even within one period (i.e., 10 min) in our scenario. As [5], we assume that there exists a modest electric energy storage facility (such as the UPS units) located in each data center which can provide perfectly "smoothing" to variable on-site renewable generation. Under this assumption, the renewable generation can be regarded as being constant during one period.

### C. The Thermal Storage Model

Motivated by the recent industry practice [1], We assume that the data center has a chilled water/ice storage system as a thermal storage facility and can use it to help cool the data center. Denote by $E_{max}$ the capacity of the thermal storage, by $E(t)$ the stored energy level at the beginning of period $t$, by $e^+(t)$ the charge rate, and by $e^-(t)$ the discharge rate during period $t$. In general, the stored energy in the thermal storage would decrease over times (i.e., self-discharge) even without discharging. Moreover, there is conversion loss during both the charging and discharging processes. Therefore, the general model of thermal storage can be as follows:

$$E(t+1) = \alpha(t)\left(E(t) + \eta^+ e^+(t) - e^-(t)/\eta^-\right), \tag{9}$$

where $\alpha(t) \le 1$ denotes the self-discharge rate and depends on the time-varying ambient temperature, $\eta^+ \le 1$ denotes the charge efficiency, and $\eta^- \le 1$ denotes the discharge efficiency.

For simplicity, we assume in this paper that $\alpha(t) = 1, \forall t$. Moreover, we combine $\eta^+$ and $\eta^-$ together to get the round-trip efficiency $\eta = \eta^+\eta^-$ and assume that conversion loss occurs only during the charging process. We also redefine $E(t)$ as the usable stored energy level at the beginning of time $t$. Under the above assumptions and definitions, we have the following dynamics for thermal storage:

$$E(t+1) = E(t) + \eta e^+(t) - e^-(t), \tag{10}$$

Within one control period, the thermal storage can either charge or discharge, but not both. That is,

$$e^+(t) > 0 \Rightarrow e^-(t) = 0, \quad e^-(t) > 0 \Rightarrow e^+(t) = 0. \tag{11}$$

However, we will temporarily ignore this constraint and decide the optimal charge or discharge control actions. Later we will construct an optimal control decision that can satisfy the above constraint.

Considering the physical constraints, the thermal storage facility usually has upper bounds on both the charge and discharge rates as follows:

$$0 \le e^+(t) \le e^+_{max}, \tag{12}$$

$$0 \le e^-(t) \le e^-_{max}. \tag{13}$$

For each period $t$, we need to ensure that the energy level in the thermal storage always satisfies the following:

$$0 \le E(t) \le E_{max}. \tag{14}$$

The initial energy level of the thermal storage is assumed to be $E(0) \in [0, E_{max}]$.

### D. The Energy Consumption Model

Energy consumption in data centers mainly consists of two parts: one is the energy consumption for powering the IT equipments and the other is for powering the cooling infrastructure to keep the IT equipments in a proper temperature range.

As [9], we assume that each server would consume either its maximum power $P_0$ when active or zero when inactive. Hence, the power consumption $P_{it}(t)$ for IT equipments at period $t$ is

$$P_{it}(t) = (n_1(t) + n_2(t))P_0. \tag{15}$$

We assume the following linear cooling energy consumption model:

$$P_{cool}(t) = \beta P_{it}(t). \tag{16}$$

where $\beta$ is a coefficient ranging from 0.1 to 1 according to the current industry measurement [3]. Without loss of generality, we assume that discharging more cooling energy from the thermal storage than needed is not allowed. Therefore, we have another constraint:

$$e^-(t) \le P_{cool}(t). \tag{17}$$

### E. The Cost Model

Since a majority of electricity in the utility grid is produced from carbon-intensive fossil fuels, it is called brown energy, while the electricity from the on-site renewable generator is called green energy. As analyzed in [10], the electricity price from the wholesale electricity market is time-varying. We denote the electricity price from the utility grid at period $t$ is $p(t)$. We assume boundness on $p(t)$ as $0 \le p(t) \le p_{max}$. As [5], [6], we assume that the marginal cost of renewable energy from the on-site renewable generation is zero so as to incentivize the usage of green energy. During period $t$, the cost of using brown energy from the utility grid is:

$$C(t) = p(t)\Big[(1+\beta)(n_1(t) + n_2(t))P_0 + e^+(t) - e^-(t) - r(t)\Big]^+. \tag{18}$$

### F. The Optimization Problem

In this paper, we are interested in minimizing the time-average expected brown energy cost. Based on the above models, our problem can be formulated as the following stochastic optimization: Choose $(n_1(t), n_2(t), e(t), \forall t)$ to

$$\text{minimize} \quad \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left\{C(t)\right\}, \tag{19}$$

subject to constraints (2), (3), (4), (5), (6), (8), (10), (12), (13), (14), and (17).

## III. Proposed Solution

The challenge to solve the optimization problem above is that the statistics of $\lambda_1(t)$, $\lambda_2(t)$, $r(t)$, and $p(t)$ may not be known and we need to design an optimal control algorithm under uncertainty. Inspired by the recently developed Lyapunov optimization technique [8], we propose an online control algorithm which requires minimum information of the random dynamics in the system and can be easily implemented online. However, a particular challenge in applying the Lyapunov optimization technique to our problem (19) is the constraint (14), which brings "time-coupling" property to the problem across different periods. To resolve it, we design a perturbed Lyapunov function which can decouple the original problem over different periods. The online control algorithm we propose can achieve the range of $O(1/V)$ within the optimal objective value with the queue backlog increase in the order of $O(V)$, where the control parameter $V$ is also constrained by the thermal storage capacity.

First, we define a perturbed Lyapunov function as follows:

$$L(t) \triangleq \frac{1}{2}\left[(Q(t))^2 + (E(t) - \theta)^2\right], \tag{20}$$

where $\theta$ is a constant to be specified later. Now define $\mathbf{K}(t) = (Q(t), E(t))$, and define a one-slot conditional Lyapunov drift as follows:

$$\triangle(t) \triangleq \mathbb{E}\{L(t+1) - L(t) \mid \mathbf{K}(t)\}. \tag{21}$$

Here the expectation is taken over the randomness of workload arrivals, electricity price, renewable generation, as well as the randomness in choosing the control actions. Then, following the Lyapunov optimization framework, we add a function of the expected cost over one slot (i.e., the penalty function) to (21) to obtain the following *drift-plus-penalty* term:

$$\triangle_V(t) \triangleq \triangle(t) + V\mathbb{E}\{C(t) \mid \mathbf{K}(t)\}, \tag{22}$$

where $V$ is a positive control parameter to be specified later. Then, we have the following lemma regarding the *drift-plus-penalty* term:

***Lemma* 1.** *For any feasible control action under constraints* (4)*,* (5)*,* (8)*,* (10)*,* (12)*, and* (13) *that can be implemented at period $t$, we have*

$$\begin{aligned}\triangle_V(t) \le B &+ \mathbb{E}\{(E(t)-\theta)(\eta e^+(t) - e^-(t)) \mid \mathbf{K}(t)\} \\ &+ \mathbb{E}\{Q(t)(\lambda_2(t) - n_2(t)\mu_2) \mid \mathbf{K}(t)\} \\ &+ V\mathbb{E}\{C(t)|\mathbf{K}(t)\}\end{aligned} \tag{23}$$

*where*

$$B = \frac{1}{2}\left[\max\{(\eta e^+_{max})^2, (e^-_{min})^2\} + (\lambda_2^{max})^2 + (N\mu_2)^2\right]. \tag{24}$$

*Proof:* First, by squaring both sides of (4) and using the fact that $(\max\{Q(t)-n_2(t)\mu_2, 0\})^2 \le (Q(t)-n_2(t)\mu_2)^2$, we obtain

$$\begin{aligned}\frac{Q(t+1)-Q(t)}{2} \le \frac{(n_2(t)\mu_2)^2}{2} + \frac{(\lambda_2(t))^2}{2} \\ - Q(t)[n_2(t)\mu_2 - \lambda_2(t)].\end{aligned} \tag{25}$$

Due to the boundness of $n_2(t)$ and $\lambda_2(t)$, we have

$$\frac{(n_2(t)\mu_2)^2}{2} + \frac{(\lambda_2(t))^2}{2} \le \frac{(N\mu_2)^2 + (\lambda_2^{max})^2}{2} \tag{26}$$

Similarly, by squaring both sides of (10) and using the boundness of $e(t)$, we arrive at the following:

$$\begin{aligned}&\frac{(E(t+1)-\theta)^2 - (E(t)-\theta)^2}{2} \\ &\le \frac{\max\{(\eta e^+_{max})^2, (e^-_{max})^2\}}{2} + (E(t)-\theta)(\eta e^+(t) - e^-(t))\end{aligned} \tag{27}$$

Summing the two inequalities above together, taking expectation of both sides over $\mathbf{K}(t)$, and adding the penalty term, we arrive at the lemma. ∎

We now present our algorithm. The main design principle of our algorithm is to choose control actions that greedily minimize the R.H.S. of (23).

*Proposed Online Control Algorithm:* Initialize $V$ and $\theta$. At the beginning of each period $t$,

1) Observe the system states $(\lambda_1(t), \lambda_2(t), p(t), r(t))$ and the queue states $(Q(t), E(t))$
2) Choose $n_1(t)$, $n_2(t)$, $e^+(t)$, and $e^-(t)$ to minimize

$$(E(t)-\theta)(\eta e^+(t) - e^-(t)) - Q(t)n_2(t)\mu_2 + Vp(t)h \tag{28}$$

subject to (2), (3), (5), (8), (12), (13), (17), and

$$h \ge 0, h \ge (1+\beta)(n_1(t)+n_2(t))P_0 + \eta e^+(t) - e^-(t) - r(t).$$

3) Update $Q(t)$ and $E(t)$ according to (4) and (10), respectively.
4) Continue to the next period.

Note that the slack variable $h$ is introduced in the optimization problem (28) to transform the nonlinear function $[\cdot]^+$ into a linear form. For each period $t$, the optimization problem (28) is a mixed-integer linear programming (MILP) problem. However, in practice, a data center usually contains thousands of servers, of which a large fraction are active [9]. Hence, we can relax the integer constraints on $n_1(t)$ and $n_2(t)$, round the resulting solution without significant cost penalties, and get a simple linear program, which can solved efficiently by simplex algorithm. The MILP can also be solved efficiently by commercial solvers such as CPLEX [11].

Another issue is that when solving the problem (28), the obtained optimal solution $(n_1^*(t), n_2^*(t), (e^+(t))^*, (e^-(t))^*, \forall t)$ may not satisfy the constraint (11). In this case, let $A = \eta(e^+(t))^* - (e^-(t))^*$ and we define the actual charge and discharge rates as follows:

$$(e^+(t))' = \begin{cases} A/\eta & \text{if } A \ge 0, \\ 0 & \text{otherwise.}\end{cases} \tag{29}$$

$$(e^-(t))' = \begin{cases} -A & \text{if } A < 0, \\ 0 & \text{otherwise.}\end{cases} \tag{30}$$

It is not difficult to see that the control decision defined above is also an optimal solution to the problem (28).

## IV. Algorithm Analysis

In this section, we prove that the proposed algorithm is approximately optimal with trade-offs between cost saving, average delay, and thermal storage capacity. Considering that data centers are usually over provisioned in practice, we further assume that the data center has enough severs to serve even the peak demand without buffering, i.e., $\lceil(\lambda_1^{max})/(\mu_1 - 1/d_{max})\rceil + \lambda_2^{max}/\mu_2 \le N$. Then, the results are summarized in the following theorem.

***Theorem* 1.** *Suppose that $0 < V \le V_{max}$, where $V_{max} = (E_{max} - \eta e^+_{max} - e^-_{max})/p_{max}$, and $\theta = Vp_{max}/\eta + e^-_{max}$ Then under our proposed algorithm, we have the following:*

1) *The queue $Q(t)$ is deterministically upper bounded as follows:*

$$0 \le Q(t) \le \frac{Vp_{max}P_0(1+\beta)}{\mu_2} + \lambda_2^{max}. \tag{31}$$

2) *The thermal storage level $E(t)$ is always in the range $[0, E_{max}]$.*
3) *If the system states $(\lambda_1(t), \lambda_2(t), p(t), r(t))$ are i.i.d. over periods, then the time average expected cost under our proposed algorithm is within the bound $B/V$ of the*

*optimal value, i.e.,*

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\{C(t)\right\} \leq C^* + B/V. \quad (32)$$

*Proof:* Due to the space limitations, we only provide a sketch of our proof. Similar proof ideas can also be found in our previous work [12], [13].

1) We prove the result by induction. First, if $Q(t) \leq (Vp_{max}P_0(1+\beta))/\mu_2$, the maximum increase during one period is $\lambda_2^{max}$. Therefore, we obtain the upper bound in this case. Second, if $Q(t) > (Vp_{max}P_0(1+\beta))/\mu_2$, our algorithm would choose the maximum possible value for $n_2(t)$. Then, the amount of arrival workloads cannot be larger than the served amount by the total capacity assumption. Therefore, the queue length cannot increase. This completes the proof.
2) Similarly, we prove it by induction. First, if $\theta < E(t) \leq E_{max}$, our algorithm would choose the maximum possible value for $e^-(t)$. Therefore, we have $0 \leq \theta - e^-_{max} \leq E(t+1) \leq E(t) \leq E_{max}$. Second, if $\theta - Vp(t) \leq E(t) \leq \theta$, we have $0 \leq \theta - Vp(t) - e^-_{max} \leq E(t) - e^-_{max} \leq E(t+1) \leq E(t) + e^+_{max} \leq \theta + e^+_{max} \leq E_{max}$, where we have used the facts that $p(t) \leq p_{max}$ and $V \leq V_{max}$. Finally, if $0 \leq E(t) < \theta - Vp(t)$, our algorithm would choose the maximum possible value for $e^+(t)$. Therefore, we have $0 \leq E(t) \leq E(t+1) \leq E(t) + e^+_{max} \leq E_{max}$, where we have used the fact that $V \leq V_{max}$. This completes the proof.
3) As we have mentioned before, our proposed algorithm is always trying to greedily minimize the R.H.S. of the upper bound (23) over all feasible control policies under constraints (4), (5), (8), (10), (12), and (13). According to the framework of Lyapunov optimization, there exists a stationary and randomized policy that can satisfy the above constraints while providing the following guarantees:

$$\mathbb{E}\{\eta\hat{e}^+(t)\} = \mathbb{E}\{\hat{e}^-(t)\}, \quad (33)$$

$$\mathbb{E}\{\hat{n_2}(t)\mu_2\} \geq \mathbb{E}\{\lambda_2(t)\}, \quad (34)$$

$$\mathbb{E}\{\hat{C}(t)\} = \hat{C}^*. \quad (35)$$

Note that this policy can only be derived based on detailed statistics, which usually has the problem of "curse of dimensionality" if solved by dynamic programming. Moreover, this policy may not be feasible to the original optimization problem because of the constraint (14). Moreover, $\hat{C}^* \leq C^*$ due to less constrained for the stationary control policy. In the following, we use the existence of such a policy to derive the performance bound of our proposed algorithm. By substitute this policy into the R.H.S. of (23), we obtain the following:

$$\begin{aligned}\triangle_V(t) &\leq B + Q(t)\mathbb{E}\{\lambda_2(t) - \hat{n_2}(t)\mu_2 \mid \mathbf{K}(t)\} \\ &\quad + V\mathbb{E}\{\hat{C}(t) \mid \mathbf{K}(t)\} \\ &\leq B + V\hat{C}^* \leq B + VC^*. \end{aligned} \quad (36)$$

Taking the expectation of both sides, using the law of iterative expectation, and summing over $t = 0, 1, \ldots, T-1$, we have

$$V\sum_{t=0}^{T-1}\mathbb{E}\{C(t)\} \leq BT + VTC^* - \mathbb{E}\{L(T)\} + \mathbb{E}\{L(0)\}. \quad (37)$$

Dividing both sides by $T$, let $T \to \infty$, and using the facts that $\mathbb{E}\{L(0)\}$ is finite and $E\{L(T)\}$ is nonnegative, we arrive at the following performance guarantee:

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\{C(t)\} \leq C^* + B/V, \quad (38)$$

where $C^*$ is the optimal objective value, $B$ is a constant, and $V$ is a control parameter.

■

## V. Numerical Evaluation

In this section, we evaluate our algorithm under real-world traces of renewable energy generation, workload arrivals, and electricity price.

### A. Evaluation Setup

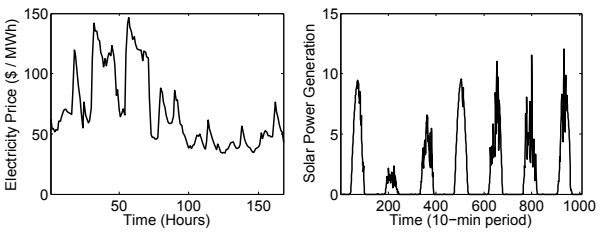
In this part, we introduce the default settings that are used throughout the evaluations unless otherwise stated. We consider a data center with a thermal storage facility and on-site PV panels. The length for one period is 10-min and the time-horizon in the evaluations is 20 days or 4800 slots.

**Wholesale Electricity Price.** We use the day-ahead hourly locational marginal prices (LMPs) from NYISO [14] as the data trace for the electricity price $p(t)$. A portion of the data trace during one week is shown in Fig.2(a).

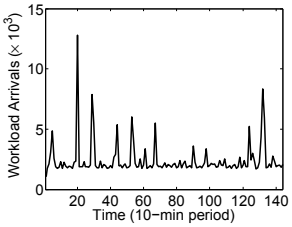
**Renewable Energy Source.** The data trace for the solar power is obtained from the Measurement and Instrumentation Data Center (MIDC) [15] at the National Renewable Energy Laboratory. It has solar irradiance measurements every 10 minutes and is shown in Fig.2(b) for one week. The trace is scaled properly so that it can meet half of the power consumption for the data center on average.

**Workload Traces.** We use the historical Hadoop (an open source implementation of MapReduce) traces on a 600-machine cluster at Facebook [16] to calculate the average 10-min workload arrival rate. A portion of the workload trace is shown in Fig.2(c) for one day. We scale the workload trace separately for interactive and batch workloads so that their IT resource requirements (in terms of of CPU-periods) are the same.

**System Parameters.** The maximum queuing delay for interactive workloads is set to be 0.1 second. We set the average service rate of interactive workloads for each server to be 20 requests per second. The average service rate of batch workloads for each server is set to be 1/100 request per second. The total number of servers $N = 1000$. Each active server consume $P_0 = 0.05$ kWh during each period. The cooling coefficient $\beta$ is set to be 1. Moreover, we set $e^+_{max} = e^-_{max} = 5$ kWh and $\eta = 1$.




(a) Electricity price (b) Solar generation




(c) Interactive workload arrival

Fig. 2. Real-world data traces used in the evaluations

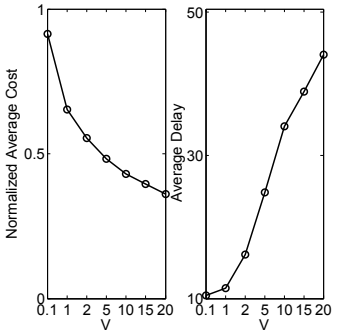



Fig. 3. The normalized average cost and delay performance of our algorithm with different $V$

### B. Numerical Results

Due to the space limit, we only discuss two sets of experimental results.

1) *Impact of $V$ on the tradeoffs between cost and delay.* In this set of evaluations, we choose different thermal storage capacities, set $V = V_{max}$, and observe the corresponding average cost and average workload delay in our proposed algorithm. We can observe from Fig. 3 that with the increase of thermal storage capacity $V$, our algorithm can lower average cost with a tradeoff in the workload delay, which validates the results of algorithm analysis in Theorem 1.
2) *Algorithm Comparison.* We compare our algorithm with another benchmark scheme that always serve the workloads whenever they arrive without any consideration on electricity price, renewable energy availability, or storage. We choose a moderate thermal storage size $E_{max}$ such that it is enough to satisfy the cooling energy demand for 3 hours with average workload arrival rates and $V = V_{max}$. The average cost achieved under the two schemes is depicted in Fig.4. We can observe that

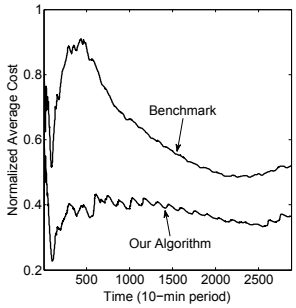



Fig. 4. Total cost comparison between our algorithm and the benchmark

our algorithm can achieve around 30% improvement compared with the benchmark in the brown energy cost.

## VI. Conclusion

Leveraging the flexibility offered by delay-tolerant workloads and thermal storage, we design an online control algorithm to facilitate green energy integration while reducing the brown energy cost in a data center. The algorithm we developed is provably optimal and can operate without any perfect future information or detailed statistics. Numerical evaluations based on the real-workload traces show that our algorithm can indeed achieve good performance in practice.

## References

[1] Apple and the Environment. [Online]. Available: http://www.apple.com/environment/renewable-energy/
[2] Amazon EC2 Spot Instances. [Online]. Available: http://aws.amazon.com/ec2/spot-instances
[3] Google's PPAs: What, How, and Why. [Online]. Available: http://www.google.com/about/datacenters/energy.html
[4] Z. Liu, M. Lin, A. Wierman, S. Low, and L. Andrew, "Greening geographical load balancing," in *SIGMETRICS*, 2011, pp. 233–244.
[5] ——, "Geographical load balancing with renewables," in *GreenMetrics*, 2011.
[6] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, "Renewable and cooling aware workload management for sustainable data centers," in *ACM Sigmetrics/Performance*, 2012.
[7] S. Ren, "Strategic pricing and resource allocation: Framework and applications," Ph.D. dissertation, University of California, Los Angeles, 2012.
[8] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool Publishers, 2010.
[9] M. Lin, Z. Liu, A. Wierman, and L. L. H. Andrew, "Online algorithms for geographical load balancing," in *International Green Computing Conference*, 2012.
[10] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *SIGCOMM*, 2009, pp. 123–134.
[11] IBM ILOG CPLEX Optimizer V12.3. [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/
[12] Y. Guo, Z. Ding, Y. Fang, and D. Wu, "Cutting down electricity cost in internet data centers by using energy storage," in *GLOBECOM*, 2011.
[13] Y. Guo and Y. Fang, "Electricity cost saving strategy in data centers by using energy storage," *Parallel and Distributed Systems, IEEE Transactions on*, 2013.
[14] California ISO Open Access Same-time Information System (OASIS). [Online]. Available: http://oasis.caiso.com/
[15] NREL: Measurement and Instrumentation Data Center. [Online]. Available: http://www.nrel.gov/midc/
[16] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz, "The case for evaluating mapreduce performance using workload suites," in *IEEE MASCOTS*, 2011.