Online server and workload management for joint optimization of electricity cost and carbon footprint across data centers

Zahra Abbasi, Madhurima Pore, and Sandeep K. S. Gupta
IMPACT Lab (http://impact.asu.edu/), Arizona State University, Tempe, AZ
zahra.abbasi, madhurima.pore, and sandeep.gupta@asu.edu

***Abstract*—Internet data centers, typically distributed across the world in order to provide timely and reliable Internet service, have been increasingly pressurized to reduce their carbon footprint and electricity cost. Particularly, data centers will soon be required to abide by carbon capping polices which impose carbon footprint limits to encourage brown energy conservation. We propose an online algorithm, called OnlineCC, for minimizing the operational cost while satisfying the carbon footprint reduction target of a set of geo-distributed data centers. OnlineCC makes use of Lyapunov optimization technique while operating without long-term future information, making it attractive in the presence of uncertainties associated with data center information e.g., input workload. We prove that OnlineCC achieves a near optimal operational cost (electricity cost) compared to the optimal algorithm with future information, while bounding the potential violation of carbon footprint target, depending on the Lyapunov control parameter, namely V. We also give a heuristic for finding V which significantly shortens the search space to adjust its value. Finally, we perform a trace-based simulation study and a small scale experiment to complement the analysis. The results show that OnlineCC reduces cost by more than 18% compared to a prediction-based online solution while resulting in equal or smaller carbon footprint.**

***Keywords*-data centers, cloud computing, carbon capping, carbon neutrality, cost saving, energy sustainability.**

I. Introduction

Large-scale Internet applications, e.g., those provided by content distribution networks (CDN), are deployed across multiple data centers in which the web contents are replicated, making it possible to provide services for users across the world efficiently and reliably. The massive energy consumption by these data centers translates into huge electricity bills and massive carbon emissions (mainly CO_2), since most of the electricity in the U.S. and in most other countries comes from non-renewable and high-carbon fuels. Therefore, data center providers are called to reduce/cap their carbon footprint from environmental activist groups (e.g., greenpeace group) and organizations and governments around the world (e.g., European Union Emission Trading System). In response, some large companies such as Google and Microsoft already took the initiative toward achieving carbon neutrality (a.k.a. net-zero) [1]. This can be achieved by capping the carbon footprint and purchasing carbon credits for the remaining offset.

Given the above requirements, our primary goal is to design a global server and workload management scheme across a set of geo-distributed data centers (a cloud). The scheme is designed not only to meet the quality of service requirements of users in different locations, but also to reduce the electricity cost (dollar per Joule) and cap the carbon footprint (CO_2 emission per Joule) of the cloud toward achieving carbon neutrality. The idea is to shift the workload toward data centers that offer green power or low electricity cost at a time, and adjust the number of active servers in proportion to the input workload. However, the problem is challenging due to the tradeoff between the electricity cost and the carbon footprint, as well as the intermittent nature of the renewables.

Data centers in a cloud are usually diverse in terms of their energy efficiency, electricity cost and carbon emission factors. Servers in different data centers have different computation capabilities and power consumption characteristics. Further, data centers get their primary power from the grid. Various parameters such as the availability of fuel type, the market, the environment, and the time of day affect the electricity cost and the carbon emission of utilities. Furthermore, usually there is no correlation between electricity prices and carbon emission factors of utilities in different regions [2]. Therefore, increasing the use of low-cost energy in a cloud does not necessarily reduce the carbon footprint of the cloud.

The above discussion, highlights the tradeoff between the energy cost and the carbon footprint of a cloud. Such a tradeoff raises the question whether there is any solution which minimizes the electricity cost, yet satisfies the carbon cap requirement of the data centers. Due to the intermittent nature of the available renewable energy, and variability of the data center workload, the carbon cap is defined over a long term operation of a data center (e.g., a year). Particularly, renewable energy sources which are deployed by both the grid and data centers (in the form of on-site solar and wind energy) affects the possible magnitude of the carbon cap. Therefore, the solution of electricity cost minimization subject to the carbon cap can only be found offline, when all future information (e.g., the availability of renewable energy, workload and electricity cost) is available. Hence, a challenging task is to design an online solution, which does not have access to the entire future information, and yet competitively minimizes the cost under the cloud's carbon cap requirement with respect to the offline solution.

We devise **OnlineCC**, an online workload and server management algorithm to minimize the electricity cost while satisfying the carbon cap requirement of a set of geo-

This research has been funded in parts by NSF CNS grants #0834797, #0855277, and #1218505. Thanks to Intel Corp. for equipment gift.

distributed data centers using only one hour ahead future information (§III). OnlineCC is based on the recently developed technique of Lyapunov optimization that enables the design of online control algorithms for time-varying systems such as data center workload management [3], [4]. We show that **OnlineCC can get time averaged cost within O(1/V) of the offline optimal solution** (see Theorem 1). More importantly, we further extend the Lyapunov optimization technique to find **the maximum carbon cap violation that OnlineCC yields in the worst case**, which is within $O(V)$ (see Theorem 1). The joint cost and carbon footprint management of a cloud can also be performed using a prediction-based online heuristic solution (similar to [5]), namely OnlineH, where the cap is managed in a best-effort manner over the prediction window. We perform a real-world trace based simulation study to evaluate OnlineCC compared to the optimal offline solution and OnlineH (§IV). The results show that OnlineCC with appropriate V value yields nearly optimal performance and surpasses that of OnlineH, particularly when clouds' carbon cap is tight (close to the minimum achievable carbon footprint) and prediction error is high (15-20% more electricity cost saving). The superiority of OnlineCC compared to OnlineH comes at the expense of increased complexity for adjusting the parameter V. To tackle this challenge, we give a heuristic guideline which significantly decreases the search space to adjust V (§III-A1). We also perform a small scale experimental study to show the effectiveness of OnlineCC in optimizing cost and carbon footprint with satisfactory performance (§V).

Table I
SYMBOLS AND DEFINITIONS.

Sym.	Definition	Sym.	Definition
t	slot index	g	power draw from grid
S	total # of slots	p^{tot}	total power cons.
j	frontend index	r	renewable harvesting
i	data center index	ε^g	grid carbon emission
N	# of data centers	ε^r	renew. carbon emission
μ	service rate	η	total carbon emission
λ	workload arrival rate	Σ	carbon cap
d^{ref}	reference delay	$\bar{\eta}$	time-avg carbon cap
d'^{ref}	service reference delay	α	electricity price
d''	network delay	X	virtual queue
Y	total # of servers	V	Lyapunov control param.
y	# of active servers	θ	see periodicity assumption
y^{slack}	slack for y	X_{lim}	see Lemma 1

II. SYSTEM MODEL

We consider N geographically distributed data centers, each containing at most Y_i servers. For simplicity, we assume severs have two power states: *active* and *inactive*, with zero power consumption in the inactive state. Further, we assume servers in a data center are homogeneous in terms of their service rate and power consumption.

End users' requests first arrive at one of M front-end proxy servers. The proxy servers then decide how to distribute the requests to data centers according to the policies dictated by our workload management scheme.

We consider a discrete-time model by dividing the entire budgeting period (e.g., typically a year) into S time slots each of which has a duration that is short enough to capture the electricity cost variation yet long enough to prevent computation and network overhead. At each slot, the workload management solution must reconcile a number of competing objectives, e.g., reducing the electricity cost, maintaining requests' delay requirement and the cloud carbon footprint cap in order to decide on (i) the workload distribution of data centers, denoted by $\lambda_{i,j}(t)$, i.e., the workload arrival rate from front-end j to data center i, and (ii) the number of active servers at each data center i, denoted by $y_i(t)$ (the rest of the servers i.e., $Y_i - y_i(t)$ are set to inactive to save the unnecessary idle power). We also assume that the workload management's decision making takes place in a centralized location, such that the information about the system is collected at a single point, and then the solutions are passed to the data centers and front-ends. We consider a delay-sensitive Internet workload for data centers, and that data centers power their servers from both the on-site renewable energy, denoted by r_i, and the grid energy, denoted by g_i.

A. Performance Model

End users to experience a high quality of service, their delay should not go above a *reference delay*, d^{ref}, as tolerated by the application. The delay experienced by a user consists of service delay, i.e., data center delay, and network delay, i.e., the delay between the user and the data center. To model the service delay we use M/M/n queuing theory model to decide on the number of active servers in a data center in such a way that the average service delay is bounded. Further, we add a slack to the number of active servers to avoid any sudden increase in delay due to the workload spikes. In the M/M/n model, given that all requests are queued, the average delay, $\bar{d}$, is as follows: $\bar{d} = \frac{1}{\lambda - n\mu}$, where n denotes the number of servers, μ denotes the service rate, and λ denotes the workload arrival rate.

Let λ_j be the time-varying average workload arrival rate at each front-end j. Then, workload distributions, i.e., $\lambda_{i,j}$, should be decided based on the following constraints, where d'^{ref}, denotes the reference average service delay, $d''_{i,j}(t)$ denotes the average network delay, n denotes the number of active servers to guarantee d'^{ref}, $0 \le y^{slack} \le 1$ denotes the slack, and y denotes the total number of active servers including the slack:

$$\begin{aligned}
&\textstyle\sum_i \lambda_{i,j}(t) = \lambda_j(t),\ \forall j,t\ \text{[service]},\\
&n_i(t)\mu_i > \textstyle\sum_j \lambda_{i,j}(t),\ \forall i,t\ \text{[queuing stability]},\\
&\tfrac{1}{n_i(t)\mu - \sum_j \lambda_{i,j}(t)} \le d'^{ref},\ \forall i,t\ \text{ [service delay] },\\
&(d^{ref} - (d'^{ref} + d''_{i,j}(t))\lambda_{i,j}(t) \ge 0,\ \forall i,j,t\ \text{ [total delay]},\\
&y_i(t) = (1 + y_i^{slack})n_i(t) \le Y_i,\ \forall i,t\,\text{[data center capacity]}.
\end{aligned} \tag{1}$$

In the above, "service" constraint guarantees providing service for all requests, "queuing stability" constraint ensures the queuing model convergence, "service delay" constraint

estimates the service delay using the M/M/n queuing model and guarantees maintaining reference average service delay, and "total delay" asserts that the sum of service and network delay is below the reference, d^{ref}. We keep the equations in linear form, specifically by bounding both the service delay and the total delay. We validated this model in §V.

B. Power Supply and Demand Model

Each data center i should receive the power required for its active servers and cooling system. Given that the linear formulation (1) decides the number of active servers based on the input workload, active servers on average are utilized near their peak utilization. Given (1), the average one-slot energy consumption of an active server, denoted by p_i, can be obtained by profiling (p_i can be estimated by including the per-server data center cooling energy). Then $y_i(t)p_i$ estimates the total one-slot energy consumed by active servers in data center i, denoted by p_i^{tot} [6]–[8]. Using more sophisticated data center power models to incorporate the complex thermal interferences of servers in order to account for cooling energy [9] is left for future work. The power draw from the grid and the available renewable energy should supply the total power demand:

$$p_i^{tot}(t) = g_i(t) + r_i(t), \forall i \text{ and } t. \tag{2}$$

C. Carbon footprint capping

Each data center i is associated with carbon emission intensities for the power source from the grid denoted by $\varepsilon_i^g(t)$ and its on-site renewable denoted by $\varepsilon_i^r(t)$ in units of CO_2 g/J. The total carbon footprint of the cloud, within slot t can be written as:

$$\eta(t) = \sum_i \eta_i(t) = \sum_i \varepsilon_i^g(t)g_i(t) + \varepsilon_i^r(t)r_i(t). \tag{3}$$

The cloud desires to follow the long-term carbon capping target, denoted by Σ, which is typically expressed for a year of operation of a data center. Mathematically, for $\bar{\eta}=\frac{\Sigma}{S}$, the long term carbon capping constraint can be written as follows:

$$\frac{1}{S}\sum_{t=0}^{S-1}\eta(t) \leq \bar{\eta}. \tag{4}$$

D. Cost minimization and carbon footprint capping

We focus on the electricity operational costs rather than the capital costs (e.g., building data centers, renewable energy installation cost). We also set the renewable energy operational cost to zero, since the primary cost for solar panels and wind turbines is the construction cost. Further, data centers would like to maximize the utilization of their on-site renewable energy. At each slot t, the operation cost is power procurement cost across all the data centers i.e., $\text{cost}(t)=\sum_i g_i(t)\alpha_i(t)$, where α denotes the electricity cost. Finally, we formulate the offline workload distribution strategy over the cloud to minimize the long-term average electricity cost of the cloud, which is demonstrated in the following optimization problem, namely **P1**:

$$\begin{aligned} &\text{minimize}_{g,r,y,\lambda} \quad \bar{\text{cost}} = \tfrac{1}{S}\textstyle\sum_{t=0}^{S-1}\sum_i g_i(t)\alpha_i(t), \\ &\text{subject to:} \qquad (1), (2), \text{ and } (4). \end{aligned} \tag{5}$$

Observe that some of the variables are real (i.e., g_i and $\lambda_{i,j}$) and some are integer (i.e., y_i). It can be proven that the well-known NP-hard Fixed-Charge-Min-Cost-Flow problem can be reduced to **P1** [6]. However relaxing y to a real variable has a very negligible impact on the cost given the thousands of servers in data centers. Therefore, we consider solving **P1** where all its decision variables are real. In this way, **P1** can be optimally solved using linear programming. Carbon capping constraint (4) couples the solution of **P1** over slots. Therefore, it is natural that optimally solving **P1** requires complete offline information (e.g., workload arrival rate, electricity price) which is impractical. To ensure there exist at least one feasible solution to **P1** and to design online solution we make the following assumptions which are practically not too constraining:

- **Boundedness assumption**: The cloud carbon footprint on every slots is upper-bounded by η_{max} which implies that the workload arrival rate and the carbon intensity associated with the cloud are finite for t=0, …, $S-1$, that is true due to the finite number of servers.
- **Feasibility assumption**: There exists at least one sequence of workload distribution policy over slots t = 0, …, $S-1$ that satisfies **P1**'s constraints.
- **Periodicity assumption**: There exits θ number of continuous slots, $\theta << S$, during which if carbon footprint is minimized (i.e., $\forall k \in \theta : \eta(k)=\eta_{min}(k)$, where $\eta_{min}(k)$ is the minimum possible carbon footprint for slot k which can be achieved by any workload distribution policy) then the average carbon footprint over θ becomes lower than that of average carbon cap ($\bar{\eta}$). The parameter θ depends on the cycle variation of the cloud carbon footprint as well as the tightness of $\bar{\eta}$, i.e., the proximity of the $\bar{\eta}$ to the minimum feasible average carbon footprint. Consider an extreme case where $\bar{\eta}\geq\eta_{max}$, then θ equals to one. If $\bar{\eta}$ is very tight, then θ becomes close to the cloud cycle variation. Note, given the weekly and daily variation of data center system parameters (e.g., workload, electricity price, carbon emission), we have that $\theta<<S$ even for the case where $\bar{\eta}$ is very tight.

III. Online Algorithm: OnlineCC

Carbon capping constraint (4) in **P1** couples the data center decisions across different time slots. Eliminating (4) from **P1** leads to an online problem, however we need a technique for managing the carbon capping requirement. While there are well-studied online problems such as Metrical Task System, we leverage Lyapunov optimization which enables online control of optimization problems with coupling property on the constraints [3], [4]. In accordance with this technique, we construct a (virtual) queue with occupancy

Algorithm 1 OnlineCC Algorithm

Initialize the virtual queue X.
for every slot $t = 1 \ldots S$ (beginning of the slot) **do**
  Predict the system parameters over slot t.
  Solve the following problem, namely **P2**:
  Minimize:

$$cost_{OnlineCC} = V \sum_i g_i(t)\alpha_i(t) + X(t) \sum_i \eta_i(t). \quad (6)$$

  Subject to: (1), and (2).
  Update the virtual queue X using (7).
end for

$X(t)$ to include the total excess carbon footprint beyond the average carbon footprint until the time slot t. Using $X(0) = 0$, we propagate $X(t)$ values over slots as follows:

$$X(t+1) = max[X(t) - \bar{\eta}, 0] + \sum_i \eta_i(t). \quad (7)$$

We design OnlineCC as given in Alg. 1 to solve the cost minimization in an online way. OnlineCC, solving the optimization problem **P2** in Alg. 1, requires only one slot ahead information as the inputs (i.e., $\lambda_j(t)$, $r_i(t)$, $\alpha_i(t)$, $\varepsilon_i^g(t)$, and $\varepsilon_i^r(t)$), since the problem **P2** removes the coupling property of **P1** (i.e., removing the constraint (4)). OnlineCC uses the control parameter V (see Alg. 1) to adjust the cost minimization and carbon capping tradeoff, for which we provide an analytically supported guideline for its adjustment.

A. OnlineCC performance analysis

We prove Lemma 1 which helps to prove the worst-case carbon capping violation of OnlineCC. It can be seen that OnlineCC minimizes the weighted sum of the electricity cost and the carbon footprint, weighted by V and $X(t)$, respectively. Lemma 1 presents a condition under which the second term of (6) outweighs its first term such that minimizing carbon footprint yields lesser value for OnlineCC objective function. The condition in Lemma 1 is related to the parameter X_{lim} which is a bound of the electricity cost difference over the carbon footprint difference across data centers for the entire time period. Mathematically, it is represented as $X_{lim}=\frac{c_{max}-c_{min}}{b}$, where, $c_{max}=\max_{i,t}(\frac{p_i}{\mu_i}\alpha_i(t))$, $c_{min}=\min_{i,t}(\frac{p_i}{\mu_i}\alpha_i(t))$, $b=min_{i,k,t,i\neq k}(\frac{p_i}{\mu_i}\varepsilon_i^g(t)-\frac{p_k}{\mu_k}\varepsilon_k^g(t)|\frac{p_i}{\mu_i}\varepsilon_i^g(t) \neq \frac{p_k}{\mu_k}\varepsilon_k^g(t))$.

Lemma 1. *Suppose data centers require non-zero active servers, and for a given slot t, $X(t)\geq VX_{lim}$, then OnlineCC minimizes carbon footprint for the current slot.*

Proof: See Appendix A for the proof. ■

Building upon Lyapunov optimization [4], Theorem 1 presents the performance analysis of OnlineCC.

Theorem 1. *(Performance Bound Analysis): Suppose $X(0)=0$, and that power demands and input workloads of data centers are bounded. Then given any fixed control parameter $V > 0$, OnlineCC achieves the following:*

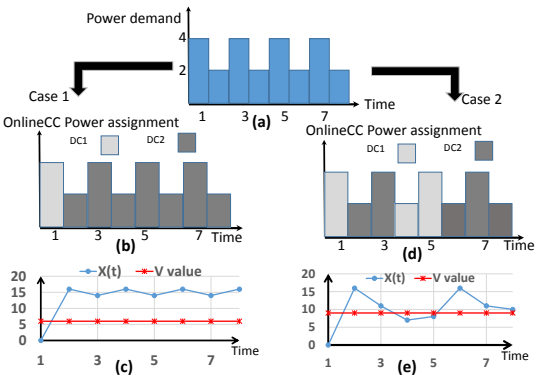


Figure 1. Examples to illustrate OnlineCC and V adjustment solutions.

1) *The carbon footprint capping constraint is approximately satisfied with a bounded deviation as follows:*

$$\sum_{t=0}^{S-1}\sum_i \eta_i(t) \leq \Sigma + VX_{lim} + \max(\theta-2,0)(\eta_{max}-\bar{\eta}) + \eta_{max} \quad (8)$$

2) *Assume data center parameters are i.i.d. over every slot, the time averaged cost under the online algorithm is within $\frac{B}{V}$ of the offline optimal time averaged cost value, $cost^*$:*

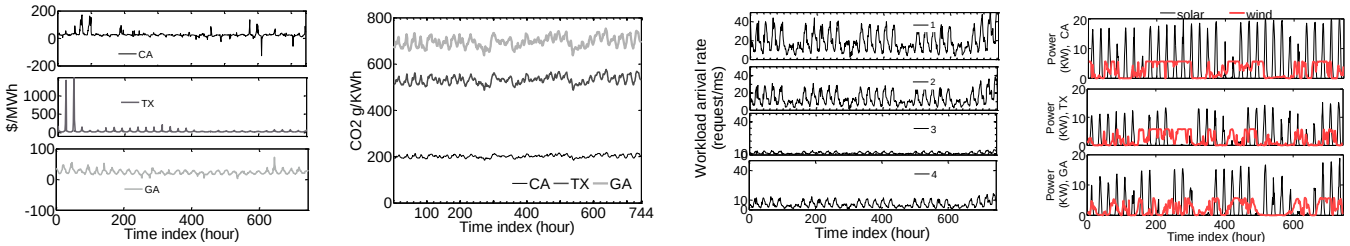
$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{cost_{OnlineCC}(\tau)\} \leq cost^* + \frac{B}{V}, \quad (9)$$

*where $B = \frac{1}{2}({\eta_{max}}^2+\bar{\eta}^2)$, $cost_{OnlineCC}$ refers to the value of (6) as shown in Alg. 1, and $cost^*$ is the optimal solution to **P1**.*

Proof: See Appendix B for the proof. ■

The results of Lemma 1 and (8) are important since they provide a deterministic bound on the maximum carbon capping violation of OnlineCC. The intuition behind (8) is that the carbon cap violation is bounded by the maximum value that X can get which is equal to the sum of VX_{lim} (the upper bound value of X to minimize carbon footprint) and the total carbon footprint backlog accumulated when minimizing carbon footprint in the worst case (i.e., over θ). These results can also be used to adjust the value of V.

1) How to choose V value?: OnlineCC uses a control parameter $V > 0$ that affects the distance from optimality. Particularly, according to Theorem 1, the algorithm achieves an average cost no more than $O(1/V)$ distance above the optimal average cost, while the large value of V comes at the expense of an $O(V)$ tradeoff in achieving the carbon cap. According to (7) the aggregated carbon violation until time t equals $\max(X(t)-\bar{\eta})$. Suppose we choose $V=V_{min}$, where $\bar{\eta}=V_{min}X_{lim}$, then according to Lemma 1, OnlineCC minimizes carbon footprint whenever either the carbon footprint over a slot exceeds $\bar{\eta}$ (due to peak workload) or the sum of backlog and the slot carbon footprint exceed $\bar{\eta}$. This means that choosing $V=V_{min}$, OnlineCC yields a value lower than that of $\bar{\eta}$ most of the time, since the offline solution does not



(a) (b) (c) (d)

Figure 2. Hourly traces for a month (Aug.): (a) electricity price, (b) carbon emission, (c) front-ends' workload, and (d) solar and wind power.

always minimizes the carbon footprint when the workload is at its peak (e.g., for loose $\bar{\eta}$). This is particularly true because VX_{lim} is an upper-bound value for OnlineCC to minimize the carbon footprint. Choosing the right value for V such that OnlineCC achieves near or very close to the carbon cap (Σ) depends on the variability of the electricity cost ($\alpha_i(t)$) and the carbon emission ($\epsilon_i^g(t)$) of data centers over time. Note, the right V value also depends on θ. Depending on the electricity cost, the optimal offline solution may violate the average carbon cap in some slots, which can be compensated in future slots (depending on θ). One heuristic solution to imitate this behavior, is to choose V such that on average around $\frac{\theta}{2}\bar{\eta}$ violations above the average carbon cap is allowed, i.e., V such that $\frac{\theta}{2}\bar{\eta}=VX_{lim}$ or $V=\frac{\theta}{2}V_{min}$. Further, for choosing V we should consider how tight the bound X_{lim} is, which can be approximately calculated using the data center input parameters (i.e., peak to mean ratio of the electricity cost and the carbon intensities).

2) Numerical Examples: Consider a cloud consisting of only one front-end and two data centers (DC1 and DC2), each having identical power consumption and service rate per server ($p=p_i$, and $\mu=\mu_i$). Suppose, both the electricity cost and the carbon footprint are constant over time, their magnitude are comparable (in the same range) and that the data center with the lower cost has the higher carbon footprint and vice versa: $\forall t, \alpha_1(t)$=2 \$/J, $\epsilon_1^g(t)$=4 CO_2/J, $\alpha_1(t)$=4 \$/J, $\epsilon_1^g(t)$=2 CO_2/J. Observe that DC1 and DC2 are optimal destination for power demand to minimize electricity cost and carbon footprint, respectively. Finally, consider T=8, and a cyclic power demand as given in Fig. 1(a). Observe that in this setting the minimum feasible average carbon footprint equals to eight and X_{lim}=1. To illustrate the proposed solutions for OnlineCC and V adjustment, consider two cases. First, suppose $\bar{\eta}$ is equal to the minimum feasible value, i.e., $\bar{\eta}$=8. According to the solution $V_{min}=\frac{\bar{\eta}}{X_{lim}}$=8. Note, the offline solution chooses to minimize the carbon cap over all slots to satisfy the carbon cap target (by assigning the entire power demand to DC2). Assume we choose $V=V_{min}$. For this setting, as shown in Fig. 1(b) except the first slot, OnlineCC assigns the power demand to DC2, since assigning power to DC2 causes to minimize the carbon footprint. This is because X value (see (7)) quickly exceeds V value (see Fig. 1(c)) resulting in the second term of OnlineCC objective function (see (6)) outweighing the first term such that minimizing the carbon footprint yields smaller value for OnlineCC objective function compared to minimizing the electricity cost.

Next, suppose $\bar{\eta}$ gets a larger feasible value i.e., $\bar{\eta}$=9. By choosing $V=V_{min}=\bar{\eta}$=9, it can be seen in Figs. 1(d) and (e) that as soon as either the slot carbon footprint exceeds $\bar{\eta}$ (slots 3 and 7), or the sum of the backlog and the slot carbon emission exceeds $\bar{\eta}$ (slots 2 and 6), X value exceeds V and that OnlineCC minimizes the carbon footprint by assigning the power demand to DC2. This results an average carbon footprint of 8.2, a value less than $\bar{\eta}$ (in agreement with §III-A1).

IV. Evaluation

We simulate a cloud consisting of three data centers at locations: CA, TX, and GA, namely DC1, DC2 and DC3, respectively. The data centers' power consumption and computing characteristics is considered from Table II(A). More accurate simulation of data centers consisting of simulating the cyber-physical interaction of servers and the cooling system in order to calculate the cooling power [10], is left for future work. We set the slot length to one hour, S to one month, and use realistic hourly traces of the electricity price, carbon intensity, renewable power, from data centers' locations. To ensure consistency, all traces are chosen from the month of July and August, since the workload traces were available for only these two months. Particularly, we use the hourly Locational Marginal Prices (LMP) of the aforementioned locations in August 2012, available at their corresponding RTO/ISO website (see Figure 2(a)). Further, we estimate the hourly carbon emission intensity of our three data centers by calculating the weighted average of carbon intensities of fuels in Table II(B) where the weights are taken from the available hourly electricity fuel mix of data center locations in August 2012 (see Figure 2(b)).

We consider four front-ends, corresponding to four time-zones in the U.S., and use two months (July and August) of NASA workload Internet trace [11]. The workload of each front-end is scaled proportionally to the number of Internet users and shifted according to the time zone for each front-end in the corresponding area, as shown in Fig. 2(c). Each data center has 380 servers, and the intensity of the workload is such that at the peak all servers of the entire cloud are required to be activated. We assume all data centers can

Table II
(A) DATA CENTERS' CHARACTERISTICS, AND (B) CARBON EMISSION OF WELL-KNOWN ELECTRICITY FUELS (CO_2/KWH).

(A)

DC	location	p	Y
DC1	Mountain View, CA.	210	380
DC2	Houston, TX	210	380
DC3	Atlanta, GA	210	380

(B)

coal	PL	NG	Nuclear energy	Wind	solar
986	890	440	15	22.5	18

receive workload of all the front-ends.

To capture the availability of wind and solar energy, we use the traces of [12] for three sites located in the aforementioned data center locations. We use the wind speed and the rated power to calculate the wind power, and Global Horizontal Irradiance (GHI) and the ambient temperature to calculate the solar power using models described in [13]. The renewable infrastructure capacity (i.e., PV cells and wind turbines) are considered to be equal for all three data centers (see Fig. 2(d)).

Prediction results: We use one month of training data (July traces) and build weekly and daily Seasonal Auto Regressive Integrated and Moving Average (SARIMA) prediction model to predict workload and solar energy, respectively. Further, we use ARMA prediction model for wind energy. The lag one (one hour-ahead) prediction error is 14%, 12% and 18% for workload, solar and wind energy respectively. The error goes up to 20%, 18% and 52% for 24 lag (24 hour ahead) prediction of workload, solar and wind energy respectively. Since wind and solar traces contain some values of zero or nearly zero, we report 95 percentile mean absolute percentage error of these two traces (e.g., lag one mean absolute error of the solar energy is 25%).

A. Experiments performed

The following algorithms are used to evaluate OnlineCC:

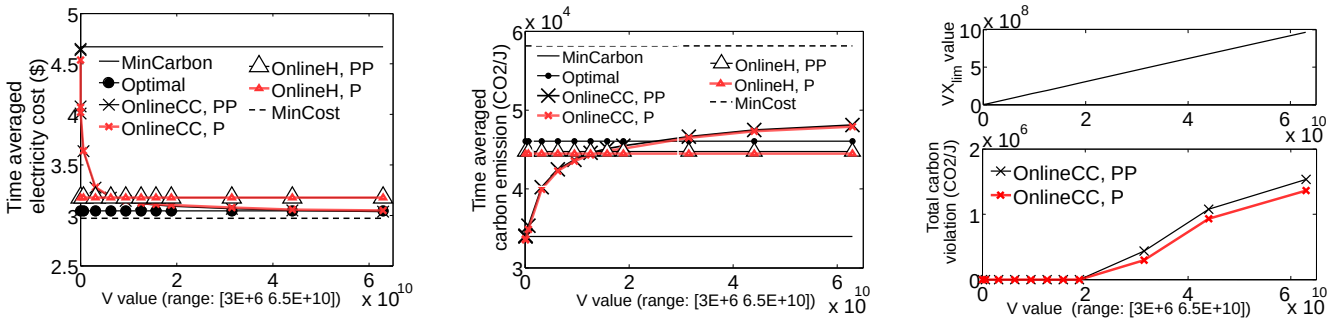
- **MinCost** (reference algorithm): performs workload distribution in the cloud to first minimize the electricity cost and then the carbon footprint.
- **MinCarbon** (reference algorithm): performs workload management to first minimize the carbon footprint and then the electricity cost in the cloud.
- **Optimal**: offline optimal solution to **P1**.
- **OnlineCC, PP**, and **OnlineCC, P**: Alg. 1 with Perfect Prediction (PP), and Predicted data (P).
- **OnlineH, PP**, and **OnlineH, P** (similar to heuristic solution of [5]): OnlineH divides the given carbon cap for a month (i.e., Σ) into chunks per day (i.e., T=24 hours/slots), where data center parameters can be predicted, and solves the problem **P1** over T. OnlineH satisfies the carbon cap in a best-effort manner, since the feasible carbon cap for a T-slot depends on the workload intensity, the availability of renewable energy and the carbon intensity on that T-slot. Similar to OnlineCC, OnlineH is evaluated either using a perfect prediction (PP), or our predicted scheme (P). Finally, we use the Receding Horizon Control (RHC) technique to improve the performance of OnlineH. Accordingly, the solution at time slot t is calculated by solving optimization problem of **P1** over the time frame T=24, given the solution at time $t-1$, and the predicted information over the entire T.

MinCost and MinCarbon can be viewed as representative of the previous schemes which solely focus on either cost minimization (e.g., [6], [8], [14]) or carbon footprint minimization (e.g., an algorithm in [2]). We use MATLAB and GNU Linear Programming Kit (GLPK) to solve all of the algorithms. In the experiments we justify Lemma 1, Theorem 1, and the solution of §III-A1. Further, we study the performance of OnlineCC versus OnlineH under various parameters i.e., V value (§ IV-B), carbon cap (Σ) (§ IV-C), and renewable energy and prediction error (§ IV-D).

B. Sensitivity to V value

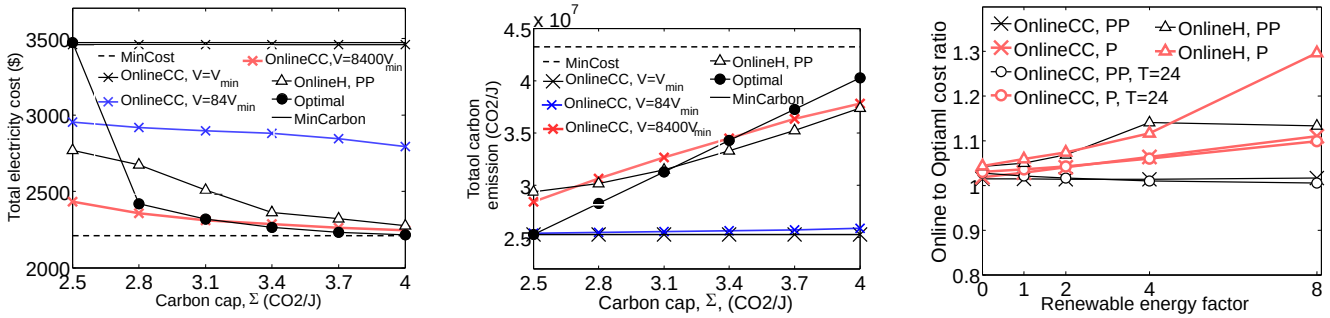
First, we run an experiment without on-site renewable energy for data centers. We set Σ to 34 CO_2 Mg/J ($\bar{\eta}$=46 CO_2 Kg/J), the mean total carbon footprint of MinCost and MinCarbon, and vary V starting from V=V_{min}=3×10^6, where X_{lim} equals to 0.015 in the data set. The results, shown in Figs. 3(a) and (b), being consistent with Theorem 1, clearly demonstrate the electricity cost and carbon reduction tradeoff which is managed by OnlineCC V parameter. Further, interestingly the prediction error has very negligible impact on the performance of the both online solutions i.e., OnlineCC and OnlineH. This is due to the relatively low prediction error of workload for both lag one and 24 hours. Comparing OnlineCC with OnlineH from Fig. 3(a) and (b), it can be seen that for $V>1\times10^{10}$, OnlineCC achieves a lower cost than OnlineH while satisfying the carbon footprint cap for $V<2\times10^{10}$. This indicates that OnlineCC surpasses OnlineH when V is appropriately adjusted. OnlineCC violates carbon cap for $V>2\times10^{10}$. However, as shown in Fig. 3(c) and being in agreement with Theorem 1, the carbon violation is much lower than the proven upper-bound. Particularly, as shown in the top graph of Fig. 3(c), for large V where OnlineCC violates carbon cap, VX_{lim} itself is much larger than the total carbon cap violation.

Figs. 3(a) and (b) show that there exists V for which OnlineCC achieves near one cost competitive ratio with respect to Optimal, while maintaining the carbon cap. In agreement with discussion in §III-A1, the same figures show that choosing V=V_{min}=3×10^6, OnlineCC yields an output almost equal to that of MinCarbon. Considering the weekly variation of the workload, the daily variation of the electricity price, and the carbon intensity as demonstrated in Fig. 2,



(a) (b) (c)

Figure 3. OnlineCC and OnlineH performance versus Optimal with and without prediction error: (a) average cost, (b) average carbon, (c) top graph: the value of VX_{lim} versus V value, and the bottom graph: total carbon violation from the cap, i.e., $\Sigma = 34E + 6$ CO_2 g/J. It can be seen that the violation in the given range of V value is up to 6% of the carbon cap target.



(a) (b) (c)

Figure 4. OnlineCC and OnlineH performance versus the magnitude of carbon cap (Σ): (a) total electricity cost, (b) total carbon emission. (c) OnlineCC to Optimal cost ratio versus the availability of renewables (e.g., renewable energy factor 8: 50% of power comes from renewable).

the parameter θ can be overestimated as the number of slots for a week, i.e., 168. According to §III-A1, for a sufficiently tight X_{lim} value, OnlineCC for $V{=}\frac{\theta}{2}V_{min}$ (i.e., $V{=}2.5\times10^8$) achieves a performance near to Optimal. However, in our dataset X_{lim} is not very tight, e.g., the value of $(\alpha_{max}{-}\alpha_{min})$ is around 100 times greater than the average electricity price differences in data centers (see Fig. 2 (a) and (b)). Therefore, OnlineCC for a V value around $2.5{\times}10^{10}$ achieves near Optimal solution performance. In general, due to the variability of input data parameters, the task of deciding the tightness of X_{lim} becomes tedious. This means that we need to run number of trials to find a right value for V. However, the heuristic solution of §III-A1 significantly shortens the search space for finding the right V value.

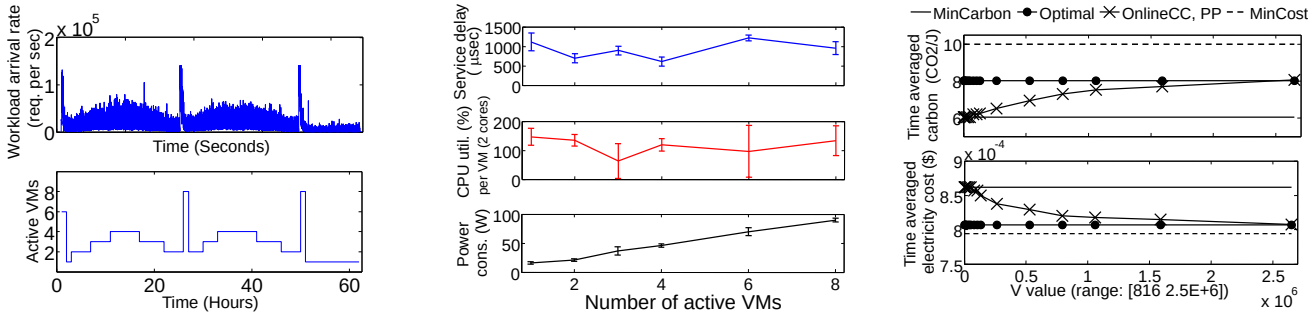
C. Sensitivity to carbon cap (Σ) value

We vary the carbon cap Σ from $\Sigma{=}\Sigma_{min}$, up to a value very close to Σ_{max}, where Σ_{min} and Σ_{max} denote the minimum and maximum cloud total carbon footprint achieved by MinCarbon and MinCost, respectively. Further, for a given Σ, we run OnlineCC for three values of V: $V{=}V_{min}$, $V{=}\frac{\theta}{2}V_{min}$, and $V{=}10^2\frac{\theta}{2}V_{min}$, where $\theta{\simeq}168$. The results shown in Fig. 4(a) and (b) indicates that when Σ is tight, i.e., Σ is close to Σ_{min}, OnlineH fails to satisfy the carbon cap (see Fig. 4(b)). However in the same situation, OnlineCC for $V{=}V_{min}$ and $V{=}\frac{\theta}{2}V_{min}$ satisfies the cap (see Fig. 4(b)). More interestingly, when Σ is tight, OnlineCC for $V{=}10^2\frac{\theta}{2}V_{min}$ yields a carbon footprint lower or very close to that of OnlineH, yet achieves 12% lesser cost than that of OnlineH (see Fig. 4(a)). Note that OnlineCC also violates cap for high V value, however, the violation never exceeds the upper bound defined in Theorem 1 (see Fig. 4(b)).

D. Sensitivity to the renewable energy and prediction error

This experiment assumes data centers have on-site renewable energy, where we scale the renewable energy by factors [0, 1.2, 4, 8]. As a result of this scaling, the portion of renewable energy is from 0% upto 50% of the total energy consumption by the cloud when using Optimal algorithm.

We calculate the online to Optimal cost ratio, where we run OnlineCC for appropriate V values for which OnlineCC, PP achieves a near optimal performance and that both OnlineCC and OnlineH approximately satisfy the carbon cap. Further, to figure out if the predictable future information improves the OnlineCC performance, we run OnlineCC when using $T{=}24$ lookahead information, in accordance of T-slot Lyapunov drift. In this way, the average carbon cap is calculated over T-slots ($\bar{\eta}{=}\frac{S}{T}\Sigma$), the virtual queue X is updated over every T-slots, and **P2** is solved over every T-slot. The results shown in Fig. 4(c) indicates that (i) with increasing renewable energy, the impact of prediction error on the performance of online algorithms worsens due to high



(a) (b) (c)

Figure 5. Experiment traces and results: (a) Microsoft Hotmail traces and number of active VMs in the experiment, (b) power and performance measurements, and (c) OnlineCC performance over experimental data.

prediction error of renewable energy even for lag one, (ii) there is no significant difference between the performance of OnlineCC with one hour ahead information compared to 24 hour ahead information, and (iii) for the condition that all algorithms almost satisfy the cap, OnlineH, P increases the cost by 30% compared to Optimal, whereas OnlineCC, P achieves only 10% more cost than Optimal.

V. Experimental evaluation

We implemented a small-scale experiment using real systems in order to validate the performance model (§II-A), as well as to validate the simulation results. We use two Intel(R) W2600 Pedestal server, 2 × Intel Quad 1.8 GHZ CPU (32 cores), and 32 GB RAM as the test servers. KVM hypervisor is used to create a virtualized environment with four virtual machines (VMs) in each system, and each VM is assigned two V-CPU and 1G of RAM. Each server emulates a data center and each VM emulates a physical server in our model. In other words, the control parameter in our experiment is the number of active VMs. Ubuntu Linux server 12.04 LTS 64-bit is installed as the VM operating system. One line of the future work is to extend the experimental study using BlueCenter Infrastructure [15], which offers small scale data center for experiments.

We developed a server-client program in C generating TCP-based requests on image files with size distribution following Pareto distribution and ranging from 0.3KB to 90KB, in accordance to a study on the file size distribution of web image content [16]. The server-side program performs image transcoding for each file, yielding in CPU-intensive operations. Each VM hosts the server-side program. The client-side programs run from a separate machine, where the workload arrival rate is taken from the two and half days of Microsoft Hotmail traces [17] (see Fig. 5 (a)) (the original Microsoft trace is a little bit modified by removing large spikes for the sake of do-ability in our testbed). The original trace gives the normalized average workload arrival (between 0 and 1). We first perform several trial runs with different workload arrival rates to model the servers' service rate (μ). Accordingly, we adjust the service rate μ to 6700 request per second and the reference average service delay d'^{ref} (including service time) to 0.0012 ms. Then we scale the traces, so that using (1) the total number of servers needed is eight with zero slack (see Fig. 5 (a)). With zero slack (y^{slack}=0) we expect our model to meet the reference average service delay d'^{ref}. Next, we choose some slots (in hour) of the trace, which represent all of the workload variation range (number of needed VMs vary from 1 to 8). We run 10 minutes of the actual Hotmail traces in all those slots (in second resolution) with appropriate number of VMs and log the servers power consumption, VMs' CPU utilization, and the turn around time of the requests. Considering that the servers and the workload generators are connected using an internal network, the turn around time of the requests are assumed to emulate data center service delay (including queuing delay and service time). Results in Fig. 5(b) indicates that the average turn around time satisfy the reference average service delay. Note that we subtract a certain amount of power from the measurements such that the total power consumption is roughly in proportion to the number of active VMs (i.e., enabling a new VM has a similar effect on power consumption as turning on a new physical server) as given in Fig. 5(b). For parameters that cannot be captured by our system (e.g., electricity price (α_i), carbon intensity (ε_i)), we use first two and half days of real-world trace data of DC1 and DC2 as presented in the simulation section.

Fig. 5(d) shows the results of running OnlineCC on the entire Hormail traces using the measurement data. We adjust Σ=480 CO_2 g/J (i.e., $\bar{\eta}$=8 for 60 slots), the mean carbon footprint achieved by MinCost and MinCarbon algorithms. We find that X_{lim}=0.01, and consequently V_{min}=816. The results are consistent with §III-A1 and the simulation study where OnlineCC performs close to MinCarbon algorithm for V=V_{min}. Having daily variation in the system parameters we have θ=24 hours. Therefore according to §III-A1, if X_{lim} is sufficiently tight by choosing $V=\frac{\theta}{2}V_{min}$ (i.e., V=1×10^3) OnlineCC achieves a performance near to that of Optimal solution. However, according to the results OnlineCC achieves near Optimal performance for V=1×10^6 (see Fig. 5(d)).

VI. Related work

There has been a growing interest in optimizing a geo-distributed data center's operational cost and carbon footprint through exploring temporal and spatial diversities among participating data centers [2], [5]–[8], [14], [18]–[20]. Our study complements the above works by taking into account the carbon capping requirements of the data centers, which is becoming increasingly important for data center operators.

Carbon capping in data centers has recently received attention in literature [2], [5], [18]–[20]. Le et al., devised a heuristic online global workload management to dynamically solve green and brown energy mix of data centers in a cloud in order to minimize the electricity cost while operating under carbon cap-and-trade policy. The online solution divides the given carbon cap (typically for a year) into chunks, i.e. one chunk per week, which is weighted by the service load predicted for the corresponding week. The workload management is then solved based on the predicted service load and the chunk for the following week. Our online reference algorithm, OnlineH, is motivated by this scheme. Xian et al., devised a request routing scheme for content distribution networks to minimize the weighted sum of the energy cost, the carbon footprint, and the delay violation cost [2]. However, the weights need to be carefully adjusted in order to achieve the desired optimization of the electricity cost and the carbon footprint, which is a very difficult task. While successfully framing the carbon capping problem for data centers, the above solutions are heuristics lacking competitive ratio compared to the offline optimal solution. There are also some recent works which used Lyapunov optimization to jointly optimize the electricity cost and the carbon footprint in data centers [18]–[20]. Ren et al., and Mahmud et al., focused on designing an online electricity cost aware workload management to achieve carbon neutrality for a single data center. The cost efficiency and carbon neutrality of the online solution is analytically proven compared to the offline solution with T future lookahead information. Our solution, however, works for geo-distributed data centers, for which we prove the upper bound of carbon capping violation compared to the optimal offline solution with entire future information. Finally, Zhou et al., leveraged the Lyapunov optimization technique to design a carbon-aware geographical load balancing, lacking a bound on the carbon footprint violation (beyond the carbon cap) of the online algorithm, the solution to tune the online algorithm control parameter (i.e., V), and the comparative study with respect to prediction based online solution.

VII. Conclusions

In this paper we developed OnlineCC, a Lyapunov based online server and workload management for minimizing the electricity cost across geo-distributed data centers, while satisfying the cloud's carbon footprint cap. OnlineCC further extends Laypunov to guarantee the maximum carbon cap violation which OnlineCC yields in the worst case. The performance of OnlineCC heavily depends on the control parameter V, which manages the electricity cost-carbon reduction tradeoff. We derived the minimum V value to satisfy the carbon cap. We also designed a heuristic guideline to choose right V value such that OnlineCC achieves nearly optimal performance. Further, the performance of OnlineCC is evaluated with respect to a prediction-based scheme, namely OnlineH, and optimal solution with future information in a simulation study with real-world traces and a in a small scale experiment. The results complement the theoretical study and indicate that OnlineCC surpasses OnlineH in cost savings by 15-20%.

VIII. Acknowledgments

Thanks to Georgios Varsamopoulos and Christopher Faucon for help with experiment setup and editing, respectively.

References

[1] [Online]. Available: http://gigaom.com/2012/05/08/microsoft-pledges-to-be-carbon-neutral-by-the-summer/.

[2] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 211–222, 2012.

[3] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource allocation and cross-layer control in wireless networks*. Now Publishers Inc, 2006.

[4] M. J. Neely, "Energy optimal control for time-varying wireless networks," *Information Theory, IEEE Transactions on*, vol. 52, no. 7, pp. 2915–2934, 2006.

[5] K. Le, R. Bianchini, T. D. Nguyen, O. Bilgir, and M. Martonosi, "Capping the brown energy consumption of internet services at low cost," in *IEEE IGCC*, 2010.

[6] Z. Abbasi, T. Mukherjee, G. Varsamopoulos, and S. K. S. Gupta, "DAHM: A green and dynamic web application hosting manager across geographically distributed data centers," *ACM Journal on emerging technology (JETC)*, vol. 8, no. 4, pp. 34:1–34:22, Nov. 2012.

[7] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. H. Andrew, "Greening geographical load balancing," in *Proc. ACM SIGMETRICS*, june 2011, pp. 233–244.

[8] 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 *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[9] Z. Abbasi, G. Varsamopoulos, and S. K. S. Gupta, "TACOMA: Server and workload management in internet data centers considering cooling-computing power trade-off and energy proportionality," *ACM Trans. Archit. Code Optim.*, vol. 9, no. 2, pp. 11:1–11:37, Jun. 2012.

[10] S. K. Gupta, A. Banerjee, Z. Abbasi, G. Varsamopoulos, M. Jonas, J. Ferguson, R. R. Gilbert, and T. Mukherjee, "GDCSim - a simulator for green data center design and analysis," *ACM Transactions on Modeling and Computer Simulation*, 2014.

[11] [Online]. Available: http://ita.ee.lbl.gov/html/traces.html.

[12] [Online]. Available: http://www.nrel.gov/midc/

[13] Y. Zhang, Y. Wang, and X. Wang, "Greenware: greening cloud-scale data centers to maximize the use of renewable energy," *Middleware 2011*, pp. 143–164, 2011.

[14] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for Internet-scale systems," in *Proc. ACM SIGCOMM*, 2009, pp. 123–134.

[15] S. K. S. Gupta, G. Varsamopoulos, A. Haywood, P. E. Phelan, and T. Mukherjee, "BlueTool: Using a computing systems research infrastructure tool to design and test green and sustainable data centers," in *Handbook of Energy-Aware and Green Computing*, 1st ed., 2012.

[16] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, "An analysis of Internet content delivery systems," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 315–327, 2002.

[17] E. Thereska, A. Donnelly, and D. Narayanan, "Sierra: a power-proportional, distributed storage system," Microsoft Research, Tech. Rep., 2009.

[18] S. Ren and Y. He, "COCA: Online distributed resource management for cost minimization and carbon neutrality in data centers," in *Super Computing*, 2013.

[19] A. H. Mahmud and S. Ren, "Online capacity provisioning for carbon-neutral data center with demand-responsive electricity prices," *ACM SIGMETRICS Perf. Eval. Rev.*, vol. 41, no. 2, pp. 26–37, 2013.

[20] Z. Zhou, F. Liu, Y. Xu, R. Zou, H. Xu, J. C. Lui, and H. Jin, "Carbon-aware load balancing for geo-distributed cloud services," in *IEEE MASCOTS*, 2013.

Appendix A.
Preliminaries to prove Theorem 1

To prove Theorem 1, first in Lemma 1 we prove the conditions where OnlineCC minimizes the carbon footprint, i.e., when $X > VX_{lim}$. Next in Lemma 2 we prove how much farther X can grow beyond VX_{lim}. The upper bound of X specifies OnlineCC carbon violation (see (7)).

A. Proof of Lemma 1

Proof: We prove the lemma for y^{slack}=0 for the sake of notation brevity. Let $p_i^{'tot}$, g_i', y_i', λ_i' and r_i' denote the value of parameters when minimizing carbon footprint. For OnlineCC to minimize the carbon footprint for any values of p_i^{tot}, g_i, y_i, λ_i, and r_i we should have the following:

$$\begin{aligned}&V\sum_i(p_i^{tot}(t)-r_i(t))\alpha_i(t)\\&+X(t)\sum_i(p_i^{tot}(t)-r_i(t))\varepsilon_i^g(t)+r_i(t)\varepsilon_i^r(t)\\&\geq V\sum_i(p_i^{'tot}(t)-r_i'(t))\alpha_i(t)\\&+X(t)\sum_i(p_i^{'tot}(t)-r_i'(t))\varepsilon_i^g(\tau)+r_i'(t)\varepsilon_i^r(t).\end{aligned}$$

The rest of the proof is about obtaining a bound for which the above inequality always holds. Since the carbon intensity of renewable power is much lower than that of grid where brown energy forms its significant energy source, $\varepsilon_i^r(t)<<\varepsilon_i^g(t)$, and that we consider zero cost for on-site renewable power, increasing the renewable energy favors both reducing the total electricity cost and carbon footprint i.e., $r_i(t)=r_i'(t)$. Therefore, given that $p_i^{tot}=y_ip_i$, to prove the above inequality it is sufficient to show:

$$X(t)\geq V\frac{\sum_i y_i'(t)p_i\alpha_i(t)-\sum_i y_i(t)p_i\alpha_i(t)}{\sum_i y_i(t)p_i\varepsilon_i^g(t)-\sum_i y_i'(t)p_i\varepsilon_i^g(t)}. \tag{10}$$

Using "service delay" constraint in (1) and the assumptions of y^{slack}=0 and $y_i>0$ we have that $y_i(t)=\frac{\sum_j\lambda_{i,j}(t)}{\mu_i}+\frac{1}{d\mu_i}$. Plugging this into (10), rearranging the terms, cancelling the term $\frac{p_i}{d\mu_i}$ from both numerator and denominator, and defining the parameters cost per flow, c as $c_i(t)=\frac{p_i\alpha_i(t)}{\mu_i}$, and carbon per flow, b as $b_i(t)=\frac{p_i\alpha_i(t)}{\mu_i}$, it is left to prove the following:

$$\begin{aligned}X(t)&\geq V\frac{\sum_i\sum_j\lambda_{i,j}'(t)c_i(t)-\sum_i\sum_j\lambda_{i,j}(t)c_i(t)}{\sum_i\sum_j\lambda_{i,j}(t)b_i(t)-\sum_i\sum_j\lambda_{i,j}'(t)b_i(t)}\\&\leq V\frac{\sum_i\sum_j\lambda_{i,j}'(t)c_{max}-\sum_i\sum_j\lambda_{i,j}(t)c_{min}}{\sum_i\sum_j\lambda_{i,j}(t)b_1-\sum_i\sum_j\lambda_{i,j}'(t)b_2}.\end{aligned} \tag{11}$$

where $c_{max}=\max_{i,t}(c_i(t))$, $c_{min}=\min_{i,t}(c_i(t))$, and b_1, and b_2 are such that $b_1-b_2=\min_{i,k,t,i\neq k}(b_i(t)-b_k(t)|b_i(t)\neq b_k(t))$. Note that for every $b_i(t)=b_k(t)$, we have $g_i(t)=g_i'(t)$, and $g_k(t)=g_k'(t)$ (this is because online algorithm first minimizes the carbon footprint and then the electricity cost) which means that such carbon factors do not affect the above inequality. Given $\sum_i\sum_j\lambda_{i,j}(t)=\sum_i\sum_j\lambda_{i,j}'(t)$, the lemma follows. ■

Lemma 2. *Suppose $X(0)$=0, then using OnlineCC, the virtual queue X is deterministically bounded as: $\forall t=0\ldots S-1, X(t)\leq VX_{lim}+\max(\theta-2,0)(\eta_{max}-\bar{\eta})+\eta_{max}$.*

Proof: We assume that $\eta_{max}>\bar{\eta}$, otherwise the proof is trivial. According to Lemma 1, once $X(t)$ exceeds VX_{lim}, OnlineCC minimizes carbon footprint. Thus the upperbound value of $X(t)$ as given in Lemma implies that there has been $\theta-1$ number of slots that OnlineCC has minimized carbon footprint, yet it has incurred carbon footprint of η_{max} on each of those $\theta-1$ slots (worst case scenario). According to definition of θ, the minimum carbon footprint in $t+1$, i.e., $\eta_{min}(t+1)$ must satisfy $\eta_{min}(t+1)<\bar{\eta}$ which follows that $X(t+1)=X(t)-\bar{\eta}+\eta_{min}(t+1)\leq X(t)-\bar{\eta}+\bar{\eta}\leq X(t)$. ■

Appendix B.
Proof of Theorem 1

Proof: First (8) immediately follows from Lemma 2, and (7), since (i) by (7), the total carbon footprint violation of OnlineCC from the carbon cap up to the end of slot t is equal to $\max(X(t)-\bar{\eta},0)$, and (ii) by Lemma 2, $X(S-1)$ never exceeds $VX_{lim}+\max(\theta-2,0)(\eta_{max}-\bar{\eta})+\eta_{max}$.

Next, the proof of (9) builds upon the recently-developed Lyapunov optimization technique [4]. By defining a quadratic Lyapunov function $L(t)$ that measures the aggregate carbon deficit in the system as: $L(t)=\frac{1}{2}X(t)^2$, and taking the same steps as [4, Lemma 1] the theorem can be proved. ■