This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2767043, IEEE Transactions on Cloud Computing

A Carbon-Aware Incentive Mechanism for Greening Colocation Data Centers

Mohammad A. Islam, Hasan Mahmud, Shaolei Ren, Xiaorui Wang

Abstract—The massive energy consumption of data centers worldwide has resulted in a large carbon footprint, raising serious concerns to sustainable IT initiatives and attracting a great amount of research attention. Nonetheless, the current efforts to date, despite encouraging, have been primarily centered around owner-operated data centers (e.g., Google data center), leaving out another major segment of data center industry — colocation data centers — much less explored. As a major hindrance to carbon efficiency desired by the operator, colocation suffers from "split incentive": tenants may not be willing to manage their servers for carbon efficiency. In this paper, we aim at minimizing the carbon footprint of geo-distributed colocation data centers, while ensuring that the operator's cost meets a long-term budget constraint. We overcome the "split incentive" hurdle by devising a novel online carbon-aware incentive mechanism, called GreenColo, in which tenants voluntarily bid for energy reduction at self-determined prices and will receive financial rewards if their bids are accepted at runtime. Using trace based simulation we show that GreenColo results in a carbon footprint fairly close (23% vs 18%) to the optimal offline solution with future information, while being able to satisfy the colocation operator's long-term budget constraint. We demonstrate the effectiveness of GreenColo in practical scenarios via both simulation studies and scaled-down prototype experiments. Our results show that GreenColo can reduce the carbon footprint by up to 24% without incurring any additional cost for the colocation operator (compared to the no-incentive baseline case), while tenants receive financial rewards for "free" without violating service level agreement.

Index Terms—Carbon reduction, Colocation, Cost budgeting, Data center energy management

1 Introduction

To support the exploding IT demands across all sectors, data centers are growing in both numbers and sizes, thereby consuming a tremendous amount of electricity and raising serious environmental concerns [1]. Despite the recent encouraging progress in reducing data center carbon footprint (referred to as "greenness" in this paper) [2]–[6], the existing efforts have been primarily focused on owner-operated data centers (e.g., Google and Amazon), while leaving a critical segment of data center industry — multi-tenant colocation data centers – much less explored.

Colocation data centers, often simply referred to as "colocation" or "colo", provide reliable power and cooling to multiple tenants who individually manage their own servers in the shared space. While it varies among different operators, a widely-adopted pricing model is based on power subscription that charges tenants for how much peak power they reserve [7], [8]. Some colocations also charge tenants based on their metered energy usage as additional costs. Nonetheless, the current pricing models do not reflect the time-varying carbon efficiency resulting from temporal changes in fuel mixes or motivate tenants to save energy at the appropriate times (e.g., when electricity is carbon-intensive), thus resulting in an unnecessarily high carbon footprint for colocations and hindering the sustainability of the data center industry.

Why does colocation need attention? Colocations are an important segment of the global data center industry, consuming nearly as five times energy as Google-type data centers all combined together [9]. It provides an appealing alternative for companies that do not want to build self-owned data centers or completely outsource their computing needs to public cloud providers [10]. Colocations also serve as physical homes for many private clouds serving individual enterprises, and public cloud services offered by many medium-scale cloud providers (e.g., Salesforce, Box) that are not "large" enough to build megascale data centers on their own. Even the largest IT companies house some of their servers in colocations: Microsoft and Google have recently leased large capacities in colocations for service expansion [11], while 25% of Apple's servers are housed in colocations [12]. By one estimate [13], there are more than 1,400 colocation data centers in the U.S., and the combined peak power demand by such facilities in New York areas are estimated to exceed 400MW (comparable to Google's global data center power demand). Importantly, the recent surge of edge computing has placed an increasingly high demand on data centers in the *edge* of the

- *Mohammad A. Islam and Shaolei Ren are with University of California, Riverside. Email: misla006@ucr.edu and sren@ece.ucr.edu*
- *Hasan Mahmud is with Amazon. E-mail: asmhasan@amazon.com*
- *Xiaorui Wang is with The Ohio State University. E-mail: xwang@ece.osu.edu*

The authors would like to thank Haven Wang and Joseph Scott for assisting with the initial testbed development.
This work was supported in part by the U.S. NSF under grants CNS-1551661, CNS-1565474, and ECCS-1610471.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2767043, IEEE Transactions on Cloud Computing

Internet, where service providers house a distributed set of servers in third-party colocations [14].

Why is greening colocations important? Despite its critical role in the data center industry and fast-growing pace, colocations have been lagging far behind owner-operated data centers in terms of sustainability. Greenpeace, a global environmental organization, has included large colocations in its latest data center sustainability report "Clicking Clean" released in 2017, indicating very poor energy and carbon efficiencies of colocations [10]. Indeed, because of the massive global footprint, colocations bear a tremendous impact on building a greener Internet, even greater impact than today's sustainability leaders like Google [10]. On the other hand, colocation operators are well motivated to reduce dirty energy consumption and carbon footprints, in voluntary pursuit of utility incentives and green certifications (e.g., LEED program offering tax benefits and brightening public image [15]) and/or compelled by pro-sustainability tenants such as Apple and Akamai [16], [17]. Thus, it is at a critical point for colocation data centers to get on board to build a green digital economy.

In this paper, we address the urgent problem of reducing carbon footprints in geo-distributed colocations. While the research problem at hand is clear, it poses the following unique challenges.

• First, while many power management techniques exist and are proven to be carbon-efficient for owner-operated data centers [3], [4], they cannot be directly applied to colocations due to the operator's lack of control over tenants' servers. On the other hand, tenants may not be willing to manage their servers for the best carbon efficiency, especially under the widely-adopted power subscription-based pricing that provides no incentives for tenants to save energy.[1] Hence, colocations suffer from "split incentive": the operator desires carbon efficiency but has no control over tenants' servers, whereas tenants manage the servers but have no incentives for carbon efficiency.

• Second, colocation operator needs to keep its long-term (e.g., yearly or monthly) operation cost under budget, and hence cannot always offer arbitrarily high financial rewards to tenants for energy reduction. That is, the total budget needs to be carefully allocated to different time periods (e.g., offers more incentives during carbon inefficient time periods), but the optimal budgeting requires complete information (e.g., future carbon emission rate) which is unknown in practice, thereby necessitating an efficient online approach.

We take the position that greening colocation data centers require joint efforts by both tenants and the operator. To address the above challenges and overcome the "split incentive" hurdle, we propose a novel carbon-aware incentive framework, called GreenColo, which financially rewards the participating tenants for energy reduction while being able to satisfy the desired long-term budget constraint for the colocation operator. Based on supply bidding where energy reduction is treated as a resource supplied by the tenants to the colocation operator, GreenColo is implemented online and enables tenants to dynamically bid for energy reduction while requesting monetary benefits. After receiving the bids, the colocation operator determines the winning bids with the goal of minimizing carbon footprint while meeting the budget constraint. To address the lack of complete offline information (e.g., tenants' future bids, carbon emission rate), we leverage the recently-developed Lyapunov technique [18] and employ a cost tracking mechanism with the intuition that more weight is given to cost saving during the process of selecting winning bids if the cost thus far has deviated much from the desired budget constraint. We also jointly optimize the colocation operator's own server management to further reduce carbon footprint.

We first demonstrate the effectiveness of GreenColo via simulations, showing that carbon footprint can be reduced by 18% without any additional cost (compared to the baseline case in which no incentive is provided), while tenants may save up to 28% of their colocation cost by participating in GreenColo. We also implement GreenColo in a scaled-down prototype to corroborate the simulations, demonstrating that GreenColo can reduce carbon footprint by 24% with no additional cost to the colocation operator while tenants receive financial rewards for "free" without violating their Service Level Agreement (SLA).

2 Overview of GreenColo

GreenColo is an online carbon-aware incentive mechanism based on supply bidding [19], with the following sequence of actions during each execution.

• **Bidding:** At the beginning of a time slot, each participating tenant submits a set of bids. Each bid contains energy reduction that the tenant is willing to carry out, along with the corresponding incentive payment he wants. Participation in GreenColo is voluntarily, and there is no restriction on the number of bids in the bidding set as well.

• **Deciding winning bids:** The colocation operator inputs the received bids into an online optimizer (as detailed in Section 4), whose output specifies one winning bid from each bidding set (hence for each tenant one winning bid).

• **Energy reduction and reward:** The bidding results are then sent back by the colocation operator to corresponding participating tenants. Finally, the tenants carry out the energy reduction as committed in the winning bids and receive the corresponding rewards.

1. As shown in our simulations, even directly passing energy bills to tenants based on energy usage cannot lead to the best carbon efficiency, because carbon emission rate varies over time and is not reflected by the utility pricing.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2767043, IEEE Transactions on Cloud Computing

TABLE 1
List of key notations.

Notation	Description
L	No. of data center locations
N_i	No. of tenants in data center i
τ_{ij}	Tenant j in data center i
λ_{ij}	Workload arrival rate of tenant τ_{ij}
μ_{ij}	Service rate of tenant τ_{ij}'s server
η_{ij}	Server power toggling cost for tenant τ_{ij}
γ_i	PUE of data center i
ϕ_i	Carbon efficiency at data center i
r_i	Onsite renewable at data center i
ϕ_{r_i}	Carbon efficiency of renewable at data center i
P_i	Total electricity usage of data center i
c_i	Total carbon emission of data center i
e	Operator's electricity cost
h	Operator's incentive payout
Z	Long-term cost constraint
q	Cost budget deficit queue

Conceptually, our bidding-based mechanism can be viewed as *supply function bidding* (SFB): power reduction is a product demanded by the data center operator and supplied by tenants (suppliers). In the language of SFB, suppliers inform the purchaser of how much demand they would like to fulfill and at what price, translating into "if given x dollars, I want to reduce y energy" in our context.

While SFB (or equivalently, demand function bidding in some contexts) does not enforce truthful bidding, it is the de facto mechanism used in many real markets, such as electricity market [20] and Amazon spot instance market (where users bid for virtual machines by proposing the prices they are willing to pay) [21]. SFB also has an advantage over direct pricing mechanism when there exist large uncertainties in the supply, as it eliminates the need of predicting how much demand suppliers can fulfill. In our context, tenants have time-varying workloads and hence, even given the same pricing signal, may respond differently over time, exhibiting a large degree of uncertainty to the colocation operator. For these reasons, we design GreenColo based on SFB so as to coordinate the tenants' energy reduction without predicting their responses *a priori*.

While tenants' participation in GreenColo is fully voluntary, we take the position that the mounting pressure from environmental groups (e.g., Greenpeace) to reduce carbon footprint [10], combined with the financial rewards and increasingly mature techniques for server power management, can incentivize (some of) the tenants to cooperate with the colocation operator in greening colocations. Our position is further corroborated by the recent commitments from large IT companies such as Akamai (which has a large colocation footprint worldwide) and Apple, which have pledged to become greener in their partnering colocations [16], [17].

3 MODELING

In this section, we formally model the colocation data center and tenants, while the key notations used are listed in Table 1. We first specify the data center's energy usage, electricity cost and carbon emission, and then present a model for guiding tenants to decide their bids. We divide the timescale of interest into K equal-length time slots indexed by $k = 0, 1, \cdots, K-1$. The duration of each time slot is decided based on how frequently the bidding process is executed. Time index is dropped, wherever applicable, to maintain the neatness of notations.

3.1 Data Center

We consider a colocation operator managing L colocation data centers, each having N_i tenants for $i = 1, 2, \cdots L$. The data centers are possibly located at different locations and connected to different power utilities, subject to different electricity prices. For notational convenience, we denote tenant j in data center i as τ_{ij} for $j = 1, 2, \cdots N_i$.

Energy consumption and electricity cost. As in any reward-based mechanisms [22], [23], in order to determine the financial rewards, the operator needs to know the reference energy usage e_{ij} by tenant τ_{ij} when GreenColo is not applied. In our case, the operator can either estimate the reference or pre-set a target value, which is announced to the tenants. Consequently, if the tenants use less energy than e_{ij}, they will be considered as contributing to energy *reduction* and hence credited/rewarded accordingly; otherwise, they are neither rewarded nor penalized, provided that they keep their power usage under the subscribed capacity. We denote $\hat{e}_{ij}$ as the energy reduction accounted for tenant τ_{ij}, while $\hat{e}_{ij} \leq 0$ means that tenant τ_{ij} is not considered as reducing energy.[2] Thus, the total IT energy consumption at data center i can be expressed as $P_i^{IT} = \sum_{j=1}^{N_i} (e_{ij} - \hat{e}_{ij})$.

Next, by capturing the non-IT energy consumption using power usage effectiveness (PUE, measuring the ratio of total energy to IT energy) and considering that an amount of r_i on-site intermittent renewable energy (e.g., solar panels) is available at data center i, we obtain the total electricity usage of data center i as $P_i = \left[\gamma_i \cdot P_i^{IT} - r_i\right]^+$, where γ_i is the PUE and $[\,\cdot\,]^+ = \max\{\cdot, 0\}$ indicates non-negative net electricity usage. Denoting w_i as the (possibly time-varying) utility-dependent electricity price, the electricity cost for the colocation operator at data center i can be derived as

$$E_i = w_i \cdot P_i = w_i \cdot \left[\gamma_i \cdot P_i^{IT} - r_i\right]^+. \qquad (1)$$

For large consumers like data centers, power utilities also impose a "demand charge" which accounts for

2. Tenants do not need to bid if $\hat{e}_{ij} \leq 0$.

the cost of power distribution infrastructure [24], [25]. It is charged based on the peak power usage during a billing cycle. The demand charge is fairly predictable, typically matching that of the previous billing cycle. While we do not explicitly incorporate it, demand charge can be handled by adding a penalty for exceeding the anticipated peak power in the optimization objective [25]. In addition, note that we consider the operator can accurately predict the renewable energy for the next time slot in (1). While it is orthogonal to our work, the operator can seamlessly integrate any suitable forecasting technique from the existing literature [26]. Nonetheless, we also show in our results that prediction errors in renewable energy do not significantly affect our proposed solution.

Carbon emission. Data centers indirectly contribute to carbon emission by consuming electricity from the power grid that has a significant carbon footprint due to heavy use of carbon-intensive fuels in electricity generation [27]. Since the source becomes indistinguishable when the electricity enters the power grid, we estimate the grid's carbon efficiency using a weighted sum of carbon efficiencies of different fuel types [4]. The weight of a fuel type is determined based on that fuel's relative contribution to the grid. We use the following formula to derive the average carbon efficiency (with a unit of g/kWh) at data center location i with F different types of fuels

$$\Phi_i = \sum_{f=1}^{F} \left[\phi_f \cdot \frac{b_{f_i}}{\sum_{f=1}^{F} b_{f_i}} \right], \tag{2}$$

where ϕ_f is the carbon efficiency of fuel type f and b_{fi} is the total electricity generation from fuel type f at the power grid serving data center i. Due to different carbon efficiencies associated with different fuel types, the carbon emission rate of the grid power changes with the fuel mix. Also, as shown in Fig. 1(c), the fuel mix of power grid exhibits a temporal diversity, as electricity generations from different fuels are continuously regulated in the power grid to maintain the balance between supply and demand. Hence, the grid's carbon efficiency also varies with time. Thus, considering ϕ_{r_i} as the carbon emission rate of onsite renewable, data center i's carbon emission is

$$c_i = \Phi_i \cdot P_i + \phi_{r_i} \cdot r_i. \tag{3}$$

3.2 Tenant

In GreenColo, tenants can voluntarily bid for energy reduction and specify the corresponding incentives they want. There is no restriction on how the tenants devise their bids, like in any SFB-based mechanisms (e.g., the one used in Amazon spot market [21]). Here, we express the bidding set of tenant τ_{ij} as

$$\mathcal{B}_{ij} \in \{(\hat{e}_{ij}^l, h_{ij}^l(\hat{e}_{ij}) | l = 1, 2, \cdots L_{ij}\}, \tag{4}$$

where $\hat{e}_{ij}$ and $h_{ij}(\hat{e}_{ij})$ represent the energy reduction and requested financial compensation, respectively, and L_{ij} is the total number of bidding tuples by tenant τ_{ij} (i.e., the number of different energy saving schemes). Although tenants have the freedom to choose very high payment bids, they are more likely to be rejected by the colocation operator and receive no rewards at all. This is analogous to the case of Amazon, where very low bids on virtual machines submitted by over-greedy users will be turned down.

4 ALGORITHM FOR GREENCOLO

In this section, we first present the problem formulation for GreenColo and then, in view of the lack of complete offline information, propose a provably-efficient online algorithm that can decide winning bids without foreseeing the far future information. We also extend GreenColo by including the colocation operator's self-managed servers to further reduce carbon footprint.

4.1 Problem Formulation

The focus of our study is to make colocations "greener": optimally decide the winning bids to minimize carbon footprint while ensuring that the colocation operator's long-term cost is kept under budget. We consider the operational cost rather than capital cost (e.g., building the data center). We formulate the problem as follows:

$$\textbf{P-1}: \quad \min_{\text{bids}} \ \bar{c} = \frac{1}{K} \sum_{k=0}^{K-1} \sum_{i=1}^{L} c_i(k) \tag{5}$$

$$\text{s.t.} \quad \sum_{k=0}^{K-1} [E(k) + h(k)] \le Z, \tag{6}$$

$$[\hat{e}_{ij}, h_{ij}(\hat{e}_{ij})] \in \mathcal{B}_{ij}(k), \ \forall\, i, j, k. \tag{7}$$

where the objective is to minimize the long-term average carbon footprint, the constraint (6) is the long-term operational cost which consists of the electricity cost $E(k) = \sum_{i=1}^{L} E_i(k)$ and total incentive paid to tenants $h(k) = \sum_i \sum_j h_{ij}(k)$. The second constraint (7) requires that only those bids voluntarily submitted by tenants can be chosen (i.e., colocation operator cannot *force* tenants to turn off certain number of servers against tenants' will).

The long-term constraint (6) couples the winning bid decisions over all the time slots of the budgeting period, thereby requiring the complete offline information. In practice, however, it is not feasible to obtain all the future bids of the tenants over the entire budgeting period (e.g. month or year). The far future on-site renewable energy generation and power grid's carbon emission rate are also very difficult, if not impossible, to predict. To address this challenge, we propose an online algorithm GreenColo which solves **P-1** with a provable bound on the deviation from the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2767043, IEEE Transactions on Cloud Computing

Algorithm 1 GreenColo

1: Inputs: total cost budget Z, budgeting period K, V_{init}, V_{min}, V update interval $\vartheta \leq K$ and α
2: Initiate: $q(0) = 0$, $V = V_{init}$
3: **for** $k = 0$ to $K - 1$ **do**
4:     **for** $i = 1$ to L **do**
5:         Input: $\Phi_i(k)$, $w_i(k)$ and $r_i(k)$
6:         **for** $j = 1$ to N_i **do**
7:             Input: $\mathcal{B}_{ij}(k)$
8:         **end for**
9:     **end for**
10:     Decide winning bids to minimize

$$\textbf{P-2}: \; V \cdot \sum_{i=1}^{L} c_i(k) + q(k) \cdot \sum_{i=1}^{L} \left[E_i(k) + h_i(k)\right]$$

        subject to constraint (7)
11:     Update $q(k+1)$ according to (8)
12:     **if** $k \mod \vartheta = 0$ **then**
13:         $z_k = \frac{Z}{K} - \frac{1}{k} \sum_{t=0}^{k} [E(t) + h(t)]$
14:         $V = \max\{V + \alpha \cdot z_k, V_{min}\}$
15:     **end if**
16: **end for**

solution with future information. Next, we present GreenColo and its operation principle.

4.2 GreenColo

Based on the extended sample-path Lyapunov optimization technique [18], we propose an online algorithm, GreenColo, which eliminates the necessity of far future information to solve **P-1**. GreenColo decouples the long-term cost capping constraint (6), by constructing a virtual cost budget deficit queue that tracks the deviation from the budget. The cost budget deficit queue evolves over time as follows

$$q(k+1) = \left[q(k) + E(k) + h(k) - \frac{Z}{K}\right]^{+}, \tag{8}$$

where the queue length $q(k)$ indicates the colocation's operational cost surplus over the allocated budget thus far. Thus, a positive queue length implies that a larger budget deficit and hence the colocation operator needs to give more weight on cost saving to meet the long-term budget constraint. Leveraging this intuition and using the budget deficit queue as a guidance, we present the online algorithm in Algorithm 1.

4.2.1 *Working principle of GreenColo*

As shown in Algorithm 1, we construct a new optimization problem consisting of the original objective function scaled by a control parameter $V \geq V_{min} > 0$ plus the operational cost multiplied by the budget deficit queue shown in (9). The queue acts as the weighting parameter for cost saving relative to carbon reduction. If the colocation operator incurs a higher cost than the budgeted amount thus far, the queue length grows and biases the optimization in consecutive time slots to nullify the difference. As we do not impose any hard constraint on long term budget, using the budget deficit queue as guiding mechanism *approximately* satisfies the cost budget. However, as shown in Theorem 1 in Appendix, there is an analytical bound on the maximum deviation from the budget as well as on the average carbon footprint.

The impact of the queue length on the optimization outcome is regulated by V. A larger V causes the change in queue length to have a less impact on the optimization, and as a result, the deviation from long-term target needs to be mitigated over a greater number of time slots and hence the potential deviation from the budget constraint may be higher. A smaller V, on the other hand, indicates that the queue has a higher impact on the optimization result and the budget surplus (deviation) is therefore quickly rectified.

The parameter V essentially determines GreenColo's performance, and regulates the trade-off between meeting the long-term budget constraint and minimizing carbon emission. However, it is difficult to choose/find the appropriate V in practice without the complete offline information [18]. In Line 12 to 15 of Algorithm 1, we incorporate method to periodically (i.e., in every ϑ slots) update V at runtime in response to the operation need. Specifically, we begin with an initial value of $V = V_{init}$, and after every ϑ slots we calculate z_k (in Line 13) which if negative indicates the budget constraint is falling short (i.e., cost is more than the budgeted amount) and vice versa. For failing budget constraint V needs to be reduced and for surplussing budget V needs to be increased. This is integrated in Line 14 of Algorithm 1 where $\alpha > 0$ is a scaling parameter and $V_{\min} > 0$ is the smallest possible value of V. Using the proposed method, GreenColo can be applied based on online information and, with an initial input of V which does not need to be accurate, will automatically guide itself towards budget constraint satisfaction. We show the impact of initial V on GreenColo through simulations in subsequent sections.

On top of removing the requirement of far future information, GreenColo also lessens the computational complexity of **P-1** which involves constrained integer programming whose complexity grows exponentially with the number of participating tenants. Specifically, **P-2** in Algorithm 1 can be decomposed over the participating tenants and, as a result, the computational complexity only grows linearly $O(n)$.

4.3 Sizing Self-Managed Servers

In addition to providing facility support for multiple tenants, it is common that the colocation operator also

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2767043, IEEE Transactions on Cloud Computing

provides a variety of other services, e.g., cloud computing, using its self-managed physical servers. The benefits of energy-efficient operation of self-managed servers are two-folds. First, it naturally contributes to colocation's carbon footprint reduction. Second, some of the cost saving resulting from self-managed servers can be passed down to tenants such that tenants are more willing to reduce energy, further reducing the carbon footprint without violating the colocation operator's budget constraint. To formalize the idea, we denote the total number of self-managed servers at data center i by M_{io}, and consider that the colocation operator can turn off some servers to reduce energy consumption subject to constraints on quality of service. Denoting the number of self-managed servers turned off by m_{io} and the average utilization by u_{io}, the total server energy consumption at data center i now becomes

$$P_i^{IT} = \sum_{j=1}^{N_i} (e_{ij} - \hat{e}_{ij}) + (M_{io} - m_{io}) \cdot (p_{io}^s + p_{io}^c \cdot u_{io}) \quad (9)$$

where p_{io}^s and p_{io}^c are the static and computing power consumptions of each self-managed server respectively. By incorporating the energy consumption of self-managed servers into the model, the online algorithm can be developed in a similar way as Algorithm 1 and hence we omit the details for brevity. Note, however, that the following delay performance constraint needs to be satisfied $d_{io}^{avg}(k) \leq d_{io}^{\max}, \forall i, k$, where $d_{io}^{avg}(k) = \left(\mu_{io} - \frac{\lambda_{io}(k)}{M_{io} - m_{io}(k)}\right)^{-1}$, derived based on M/M/1 queueing model [2], specifies the average delay performance for the colocation operator's self-managed services at data center i, with μ_{io} and λ_{io} being the service rate and workload arrival rate, respectively.

4.4 Bidding Strategy

Thus far, the algorithm in GreenColo has focused on the colocation operator's decision — deciding the winning bids. To complete the design of GreenColo, an important issue is how participating tenants determine their bids. In general, tenants have the freedom to bid arbitrarily, although over-greedy bids are likely turned down by the colocation operator and hence do not necessarily benefit the tenants. Here, we provide an *example* bidding strategy that may be used by the tenants to specify their bids in GreenColo. This is only to facilitate a clearer understanding of how tenants may participate in GreenColo, while they can choose their bidding strategies completely at their own discretion.

In our setting, we consider a well-studied approach of "turning unused servers off" as tenants' energy saving technique [2], [28], while many other knobs (e.g., scaling down CPU frequencies) are also available for energy saving. A variant of this approach, called "Autoscale", has already been used in Facebook's production system for energy saving [29]. Formally, tenant τ_{ij} has M_{ij} servers in data center i, and turning off m_{ij} servers can save an energy of $\hat{e}_{ij} = m_{ij} p_{ij}^s$, where p_{ij}^s is the static energy consumption of a server per time slot.

Tenants may incur *costs* when shedding energy and hence, need to be financially compensated. Concretely, following the literature [2], we consider two types of rewards requested by tenants for energy reduction: *inconvenience reward* and *delay reward*, as detailed below.

Inconvenience reward. We use inconvenience reward to collectively compensate for the possible wear-and-tear caused by server power toggling, as well as the reduced processing capacity for the tenants to tackle sudden surge in workloads [2]. We model the inconvenience reward of tenants by an increasing function $\eta_{i,j} \cdot m_{i,j}$, where $\eta_{i,j} > 0$ is a scaling factor decided by the tenants.

Delay reward. The tenants turn off servers by consolidating workloads into fewer servers, which may result in delay performance degradation for applications and causing "cost" to tenants [2]. We represent the delay reward of tenant τ_{ij} by d_{ij}, which compensates tenant's performance loss and intuitively increases with the number of servers m_{ij} turned off and the tenant' workload arrival rate λ_{ij}. As a concrete example, we employ a widely-applied queuing-theoretic model by considering an M/M/1 queue at each active server [2], [3], [30]. Considering that the total traffic λ_{ij} is equally distributed among all the active servers, we express the delay cost of tenant τ_{ij} as

$$d_{ij}(m_{ij}, \lambda_{ij}) = \beta_{ij} \cdot \lambda_{ij} \cdot \left(\frac{1}{\mu_{ij} - \frac{\lambda_{ij}}{M_{ij} - m_{ij}}} - d_{ij}^{th} \right)^+, \quad (10)$$

where M_{ij} is tenant τ_{ij}'s total number of servers, μ_{ij} is the service rate of each server (measuring the amount of workloads that can be processed in a unit time), β_{ij} is a factor converting the experienced delay to an equivalent monetary value, the operator $(\,\cdot\,)^+ = \max\{\,\cdot\,, 0\}$, and $d_{i,j}^{th}$ is the *soft* average delay threshold (i.e., users are indifferent of the delay performance below this threshold). We also consider that each tenant has a maximum average delay constraint,

$$\frac{M_{ij} - m_{ij}}{\mu_{ij}(M_{ij} - m_{ij}) - \lambda_{ij}} \leq \bar{d}_{ij}^{\max}. \quad (11)$$

The maximum delay constraint in (11) essentially defines the upper limit on the number of servers that may be turned off, and hence bounds the maximum server utilization. Although not applicable for all application scenarios, the delay model used in (10) provides a tenable estimation of the resulting delay performance and hence is widely used for performance analysis [2], [30].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2767043, IEEE Transactions on Cloud Computing

TABLE 2
Simulation parameters (in U.S. currency).

	Tenant 1	Tenant 2	Tenant 3
Delay cost β (¢/s/job)	0.015	0.01	0.005
η ($ per server-hour)	0.03	0.03	0.03
Power cost ($/kW/month)	145	145	145
Service rate (jobs/hour)	360,000	180,000	30
Soft threshold on avg. delay	10 ms	20 ms	150 s
Avg. delay constraint	20 ms	30 ms	250 s

By combining both delay and inconvenience rewards, the total reward requested by tenant τ_{ij} is expressed as

$$h_{ij}(\hat{e}_{ij}) = h_{ij}(m_{ij}p_{ij}^s) = \eta_{ij} \cdot m_{i,j} + d_{ij}(m_{ij}, \lambda_{ij}). \quad (12)$$

We consider the above bidding strategy because different tenants may have different cost impacts for the same energy reduction. Naturally, a tenant with a lower performance degradation should reduce energy at a lower reward rate and vice versa. In addition, while in our example bidding strategy a tenant with a higher performance degradation may ask for more rewards, it is not over-rewarded as the colocation operator chooses the winning bids and a tenant asking for more rewards will be asked to reduce less energy (or even none at all). Further, the operator has a budget constraint which also limits its expenditure on rewards.

There also exist several alternative approaches for tenants to quantify their costs (e.g., [4], [31]) incurred for energy shedding and decide their bids accordingly. In any case, just like users on Amazon spot market [21], participating tenants can choose their preferred approaches to bid in GreenColo at their own discretion.

5 Simulation Study

In this section, we present a trace-based simulation to demonstrate the effectiveness of GreenColo, showing that GreenColo can reduce carbon emission by 18% and save tenants' cost by up to 28%, while incurring no additional operational cost for the colocation operator (compared to the no-incentive case). We first present our simulation setup and then show the evaluation results. While the actual results depend on particular simulation settings, our qualitative insight holds broadly: GreenColo *unifies the interests of both the colocation operator and tenants towards making colocations greener.*

5.1 Setup

We consider three colocation data centers located at Silicon Valley (CA), New York City (NY) and Chicago (IL), all of which are major colocation markets [13]. Each data center houses three participating tenants, which have 2,000 servers each. This is typical in wholesale colocation data centers, where each tenant subscribes a large capacity. The tenants of these three data centers are considered to have various delay performance requirements, with the first tenant running highly delay-sensitive, the second tenant running moderately delay-sensitive, and the third one running delay-tolerant workloads. The tenants of these data centers are numbered sequentially: tenants #1 to #3 are in CA data center, tenant #4 to #6 are in NY data center and tenant #7 to #9 are in IL data center. The modeling parameters for tenants are shown in Table 2.

— The parameter β converts delay performance to monetary value and quantifies the tenant's *average* cost (i.e., requested incentive payment) per job, if the resulting average delay exceeds the software threshold by one second. As shown in simulations, the values of β in Table 2 are already high enough to ensure that application performances are not noticeably affected. Similar model is also considered in prior work [3].

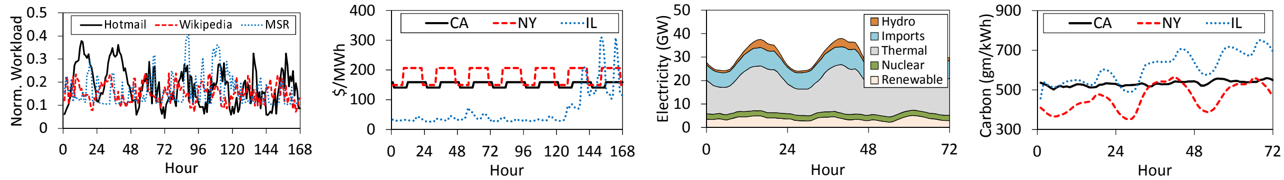
— The parameter η specifies the server unavailability cost for turning off each server for one hour. While there is no public disclosure of such data, we believe that $0.03 per server-hour is reasonable: with a 150W idle power for each server (in our setting), 3 cent/server/hour is already higher than the electricity cost saving achieved by turning off a server, had the tenants run servers in their own data centers (assuming a fair electricity price of 15 ¢/KWh). In other words, if tenants would like to turn off idle servers for cost saving in their own data centers (as extensively studied [2]), they should be more willing to do so in colocations.

— We consider the prevailing pricing model based on power subscription [32], and 145 U.S.$/kW/month in all of the three data centers, which is a fair market value [8], [33]. Service rates indicate the average number of jobs that can be processed by one server, the soft delay threshold indicates the average delay below which users are indifferent with the service quality, and the average delay constraint specifies the acceptable service quality.

We consider that each server has an idle power of 150W and peak power of 250W. The budgeting period of our simulation is considered to be the first quarter of 2014 (January to March) with each time slot equal to 1 hour. The default quarterly budget constraint is set to 1.3 million U.S. dollars, which is the total cost the colocation incurs when no incentive is provided and all servers are turned on as the status quo. The peak power of each of the 3 colocations is 2.4MW with PUE equal to 1.6, which is a fair value for colocations although some owner-operated data centers such as Google have reached a much lower PUE.

•**Workload.** We use three different workload traces for the tenants. The workload identified as "Hotmail" is taken from a 48-hour trace of 8 servers of Hotmail [34]. "Wikipedia" traces are taken from [35], which contain 10% of all user requests issued to Wikipedia from a 30-day period of September 2007, and "MSR"

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2767043, IEEE Transactions on Cloud Computing



(a) Normalized workload traces (b) Electricity price (c) Fuel mix data for CA (d) Carbon emission rate of power grid

Fig. 1. Trace data used in the simulations.

workload is a 1-week I/O trace of 6 RAID volumes at Microsoft Research Cambridge [34]. Due to the lack of available traces for the entire budgeting period, we add up to 30% random variations and extend the available traces to get the 3-month trace. We then take these 3-month traces and make three copies to have 9 traces. We add 20% randomness in each of these 9 traces to use as the workload trace for the 9 tenants. The workloads are normalized to corresponding tenant's maximum processing capacity and a snap shot of the traces is shown in Fig. 1(a).

• **Electricity price and on-site renewable energy.** We take the electricity price of non-residential customer from the utilities that serve the 3 data center locations. The electricity price trace of the first 48 hours is shown in Fig. 1(b). We collect the solar power generation data from [27] for California and use it as the trace for on-site renewable energy of the 3 data centers after adding up to 20% random variation. We re-scale the data so that the maximum on-site renewable energy is 10% of the maximum peak power of the considered data centers.

• **Carbon emission rate:** Due to lack of utility-level energy fuel mix data, we collect the fuel mix data from California ISO [27], and use carbon efficiency for fuel types presented in [36] to calculate carbon emission rate for CA data center. For the NY and IL data center, we estimate the hourly carbon emission rate from the annual average fuel mix and exploiting the fact that during peak load the carbon emission is higher because the peak load serving generators are typically run on oil and gas, and the daily average carbon emission rate is approximately 60% of the peak carbon emission [4], [37]. The energy fuel mix and carbon emission rates for the first 3 days are shown in Fig. 1(c) and 1(d).

5.2 Results

We present our simulation results below. First, we introduce three baselines with which we compare GreenColo. Then, we examine the execution of GreenColo and show the performance comparison. Finally, we demonstrate the applicability of GreenColo in different scenarios. Unless otherwise stated, all the results are hourly values.

5.2.1 Baselines

We consider three baselines as below.

• **No Incentive (N-INC):** This is a baseline case in which no incentive is provided and the colocation is operated following the existing practice with no tenants' servers turned off.

• **Direct Incentive (D-INC):** In D-INC, the colocation operator directly forwards the current electricity price multiplied by the effective PUE (reflecting the additional facility energy saving) to the tenants as an incentive for energy saving. Tenants individually determine their energy reduction to maximize their own benefits (i.e., difference between incentive received and cost incurred).

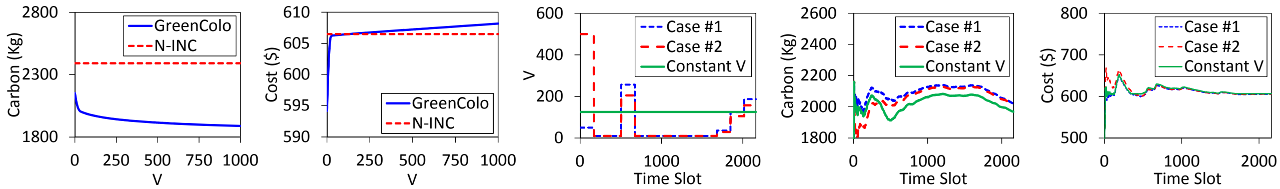
• **Optimal Offline (OPT):** This is the optimum algorithm which, with complete future information (e.g., future bids submitted by tenants), solves the offline problem **P-1** and minimizes the carbon footprint subject to long-term budget constraint. OPT is not feasible in practice, but provides a lower bound on the carbon footprint that can be possibly achieved by GreenColo.

5.2.2 Execution of GreenColo

We first show the impact of control parameter V on the performance of GreenColo in Fig. 2(a) and Fig. 2(b). It can be seen that V governs the trade-off between carbon footprint reduction and budget constraint satisfaction: when V increases, GreenColo focuses more on reducing carbon footprint while caring less about operational cost, and vice versa. When $V \approx 125$, the desired budget constraint is satisfied, while the carbon footprint is significantly reduced compared to N-INC (by 17.7%).

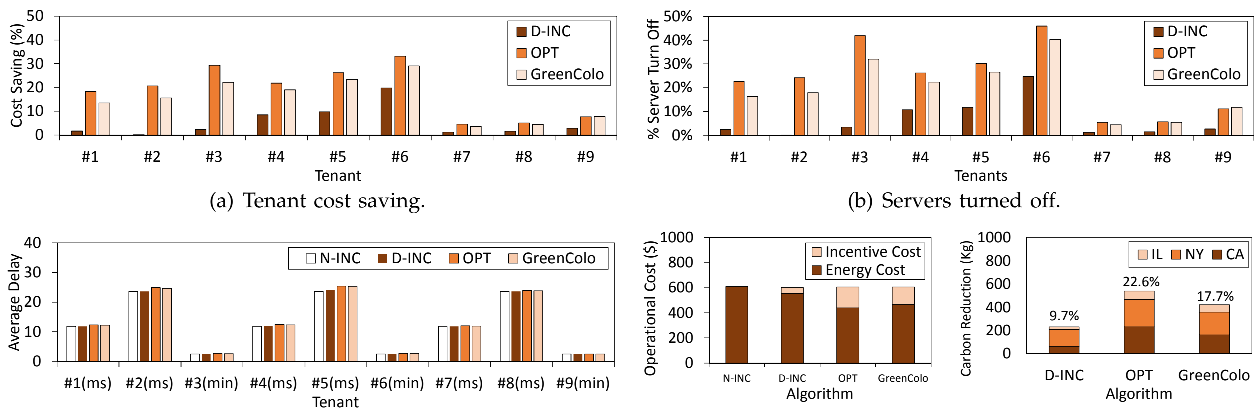
Next, Fig. 2(c) shows the dynamic change of V with different initial V. We see in Fig. 2(e) that regardless of the initial values, the operational costs are very close to the N-INC case, satisfying the long-term budget constraint. However, as shown in Fig. 2(d), because V is regulated progressively to meet the budget target, changing V dynamically results into a slightly higher carbon emission (still less than 3% deviation) compared to the case with a fixed V that is chosen in advance to satisfy the budget constraint. This demonstrates that the parameter V can be autonomously adjusted for satisfying budget constraint, which is important for applying GreenColo in practical systems since the optimal constant V cannot be predetermined without accurately.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2767043, IEEE Transactions on Cloud Computing



(a) Carbon footprint. (b) Operational cost. (c) Dynamic V. (d) Carbon footprint. (e) Operational cost.

Fig. 2. **(a) and (b)** Impact of V on carbon footprint and operational cost, where V governs the trade-off between carbon footprint and cost budget constraint. **(c), (d) and (e)** Dynamically updating V every 7 days. Case 1: initial $V = 50$. Case 2: initial $V = 500$. Constant $V = 125$. $\alpha = 50$ and $V_{\min} = 10$. Regardless of initial V, GreenColo satisfies the cost budget while having slightly higher carbon footprint compared to case with fixed V.



(a) Tenant cost saving. (b) Servers turned off.

(c) Delay performance. (d) Operational cost. (e) Carbon reduction.

Fig. 3. Performance comparison between GreenColo and baselines. Cost saving by GreenColo is close to the OPT and significantly higher than D-INC. There is as much as 29% cost savings by tenants without much impact on performance, whereas the data centers reduce carbon emission by more than 17%.

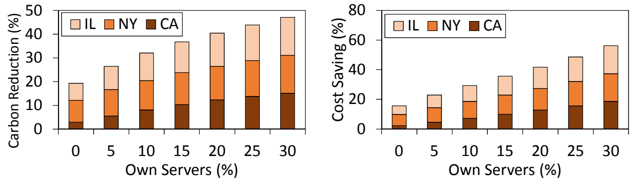
5.2.3 Performance comparison

In Fig. 3, we compare the performance of GreenColo with the baseline algorithms.

Reduce tenants' costs without noticeable performance degradation. First, we show the cost savings and delay performances of the tenants under different algorithms. In cost saving percentages, we only consider power subscription cost assuming that tenants carefully subscribe to power based on their peak server power; other costs, such as space and network connectivity cost, are often lower than power costs with a significant variation across tenants and hence excluded from our consideration [7]. Fig. 3(a) shows that using GreenColo, there is as much as 29% cost saving by the tenants. We see a general trend that the 3rd tenant in each data center (tenants #3, #6 and #9) enjoys higher cost savings than the other 2 tenants. This is because of their higher delay tolerance: they have a low delay cost and can reduce more energy for less incentive, consequently being favored by GreenColo when deciding the wining bids. We see in Fig. 3(b) that, these tenants turn off more servers than the other tenants in the same data center. The tenants at IL data center (tenant #7, #8 and #9) have the least cost savings because the operator cannot offer high incentive for energy reduction as the colocation cannot save much from energy reduction due to low electricity cost at IL. Also, cost saving for D-INC is significantly lower than GreenColo, because D-INC directly passes the electricity cost saving to tenants without considering the time-varying nature of carbon emission rates. In Fig. 3(c), we show the average delay of the tenants. We see that there is no significant increase in delay performance compared to N-INC for the tenants when they participate in GreenColo, reducing energy and saving cost. This is because tenants typically accept cost saving and green practices, only when application performance is not compromised.

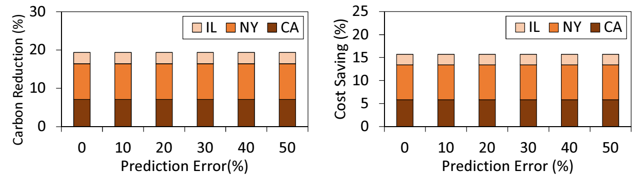
Reduce carbon footprint without increasing operational cost. We see from Fig. 3(d) that all the algorithms result in the same operational cost as N-INC, which we use as a reference case. Moreover, GreenColo provides a greater incentive payment to tenants than D-INC, because GreenColo is able to perform a joint optimization across all tenants and data centers by taking the advantage of heterogeneities among tenants and data centers. In Fig. 3(e), we show the average footprint reductions under different incentive mechanisms compared to N-INC. We see that GreenColo can reduce carbon emission by 17.7% compared to 9.7% reduction by D-INC. Naturally, as CA and NY tenants turn off more servers, these 2 data centers are dominant contributors towards the carbon footprint reduction. We also observe that, in terms of carbon footprint reduction, GreenColo is fairly

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2767043, IEEE Transactions on Cloud Computing



(a) Carbon reduction. (b) Tenant cost saving

Fig. 4. Impact of self-managed servers. Both carbon reduction and cost savings opportunity increases as the operator can pass on the savings from its own servers to further incentivize the tenants for energy reduction.



(a) Carbon reduction. (b) Tenant cost saving

Fig. 5. Impact of renewable generation prediction error.

close to OPT (17.7% versus 22.6%), demonstrating the effectiveness of GreenColo even though only online information is available.

Energy efficiency of self-managed servers contributes to carbon footprint reduction. Here, we study the case where the colocation operator also hosts its self-managed servers in colocations.We vary the percentage of self-managed server in each data center from 0 to 30% while keeping the total number of servers same by rescaling tenant servers. We set the data center total operational cost, incurred when no servers (both tenants' and self-managed) are turned off, as the long-term budget constraint. In Fig. 4, we see that both carbon reduction and tenants' cost savings increase with increase in the percentage of self-managed servers in colocations, reaching up to 45% carbon reduction and 55% tenant cost savings for 30% self-managed servers. This is because the colocation operator can exploit the savings from its self-managed servers to further incentivize tenants and drive them into greater carbon reduction.

Insensitivity against renewable energy prediction error. Here, we study the impact of renewable generation prediction error on GreenColo. Toward this end, we consider that the operator inaccurately estimates the renewable generation in each time slot. We study the carbon reduction and tenants' cost saving by varying the degree of prediction errors from $\pm 0\%$ to $\pm 50\%$ and show the results in Fig. 5. We see that there is virtually "zero" impact on carbon emission and cost saving with prediction errors in the renewable generation. This is because the renewable generation acts as an offset to the electricity cost, and the prediction error is corrected with the actual renewable generation after every time slot when the budget deficit queue is updated.

6 Prototype Experiment

The previous section highlights the benefits of GreenColo over existing solutions in simulation environments. To corroborate the simulation and ensure that GreenColo is practically applicable, we subject GreenColo to a scaled-down prototype experiment and contrast it with the current no-incentive baseline approach. Next, we first describe the experiment setup, followed by how the tenants devise their bids, and then present the experiment results.

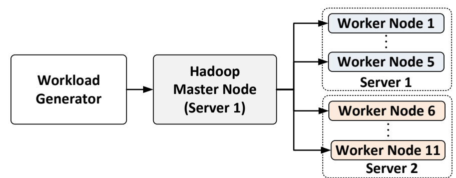


Fig. 6. Tenant #1 processing Hadoop workload. The Hadoop setup has one master node and eleven worker nodes in two servers.

6.1 Setup

Testbed. Due to hardware constraints, we implement a scaled-down colocation facility hosting two tenants on a testbed consisting of five Dell PowerEdge R720 rack servers. Four of these servers each have one 6-core Intel Xeon E-26XX Processor (210-ABVP), 32GB RAM and four 320 GB hard drives in RAID 0 configuration. The 5th server has two Intel Xeon CPUs and eight 320 GB hard drives in RAID 0. The 5th server has significantly higher I/O capability and hence is used to host the database VMS. Each server has six VMs, using Xen-Server 6.2 as the virtualization platform. The power consumption of each server is measured with WattsUpPro power meter. We implement GreenColo in a separate HP tower server with Core i7-3770 CPU and 16 GB of memory. This tower server acts as the colocation operator's control module for executing GreenColo and communicates with tenants.

Tenant #1. As illustrated in Fig. 6, tenant#1 processes delay-tolerant Hadoop workloads in two servers hosting 12 VMs in total. We configure 11 virtual machines as the worker node and one virtual machine as the master node of the Hadoop system. We implement a scaling module that can adjust and/or consolidate the number of worker nodes to trade for energy. Each Hadoop job consists of two parts: first, generate a distributed random file using *RandomTextWriter* (Hadoop's default) on HDFS (Hadoop Distributed File System); second, run *sort* benchmark on the randomly generated file. The completion time for each job is recorded as the performance metric of interest.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2767043, IEEE Transactions on Cloud Computing

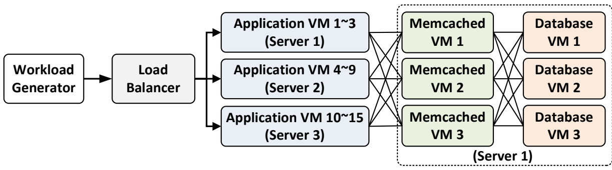
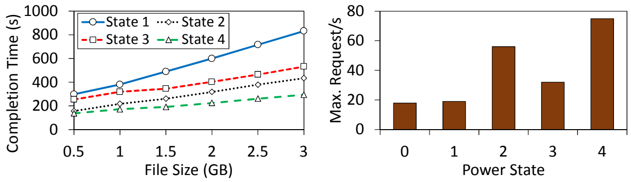


Fig. 7. Tenant #2 processing KV workload. The multi-tiered KV system consists of one front end load balancer, fifteen application nodes, three memory cache nodes and three database nodes in three servers.

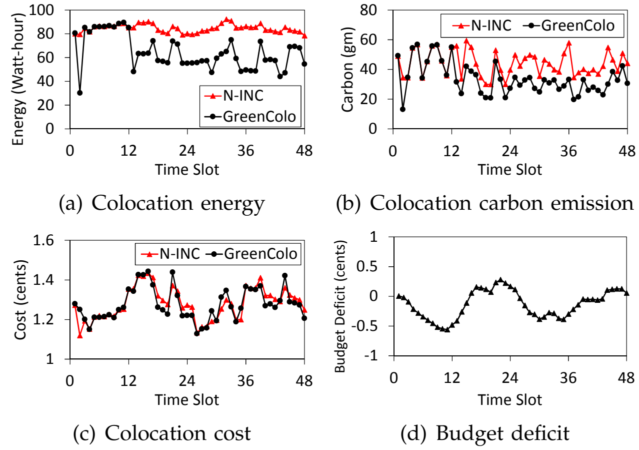


(a) (b)

Fig. 8. **(a)** Tenant 1: Hadoop job turn-around time. Power states are: State 1 (L, Z), State 2 (H, Z), State 3 (L, L), and State 4 (H, H). **(b)** Tenant 2: Maximum job request per second subject to SLA (i.e., 95% delay below 500ms). State 1 (H, L, Z), State 2 (H, H, Z), State 3 (H, L, L), and State 4 (H, H, H). Each letter corresponds to a server's power state.

Tenant #2. Using the remaining three servers as illustrated in Fig. 7, tenant #2 processes key-value-store (KV) workload, which resembles a multi-tiered web service such as social networking [38]. Our implementation of Key-Value store has 4 tiers: front-end load balancer, application, memory cache, and database. The load balancer is a Java program that receives jobs from the generator and routes the requests to the application servers. The application is implemented in PHP running on an Apache web server. We use Memcached, a distributed memory object caching system, in the mid-end for improving the database performance. We use MySQL as our database which contains 100 million key value pairs. The server with 2 CPUs and high I/O capacity hosts Memcached VMs, three replicated database VMs, three application VMs, and a VM for load balancer. The other two servers each host six application VMs. In a separate server, we implement the job generator which can send workloads of various job sizes following a Zipf distribution.

We use the "MSR" trace as the Hadoop workload and "Wikipedia" trace as the KV workload. The traces are appropriately scaled down to have a 20% average utilization. To avoid lengthy running, we scale down each time slot to 10 minutes and run the experiment for 48 time slots. We use the hourly electricity price and carbon emission rate of CA. The total cost budget for the 48 time slots (8 hours) is set to 61.16¢, which is the total cost without any incentive.



(a) Colocation energy (b) Colocation carbon emission

(c) Colocation cost (d) Budget deficit

Fig. 9. Performance comparison between GreenColo and N-INC. GreenColo have 24% carbon reduction, while meeting the budget constraint by closely following N-INC's cost.

6.2 Power Management

Both tenants can adjust servers' power states for energy saving. We use three different power states for each server: High Performance (H), Low Performance (L), and Turned Off (Z). Note that, we do not consider virtual machine migration and therefore our VM placement in the servers are fixed. Consequently, when a server is put to "turned off" or Z state, all the VMs hosted on that server are also turned off. The energy saving is done subject to SLA constraints. In particular, tenant #1 running Hadoop jobs has an SLA of 10 minutes for the maximum job completion time, while tenant #2 running interactive jobs has an SLA requirement of 500ms on 95-percentile delay which is a reasonable setting as considered in prior research [38]. Tenant #1 has two servers and Tenant #2 has three servers. Their processing capabilities under different combinations of power states are shown in Fig. 8. The power state can be adjusted in accordance with incoming workload arrival rates subject to SLA, as similarly adopted by Facebook's production systems [29]. We also consider a 10% margin on SLA requirement for both tenants (e.g., tenant #1 provisions resources to serve workloads within 9 minutes, whereas the actual SLA requirement is 10 minutes). Tenant #1 asks for ¢10 per kWh energy reduction and tenant #2 asks for ¢15 per kWh energy reduction. They also add ¢0.05 for each server turned off.

6.3 Results

Here, we report the experimental results by comparing GreenColo against N-INC, while noting that D-INC has a similar behavior with the simulation result and omitted for brevity.

We first show the energy consumption and corresponding carbon emission in Figs. 9(a) and 9(b),

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2767043, IEEE Transactions on Cloud Computing

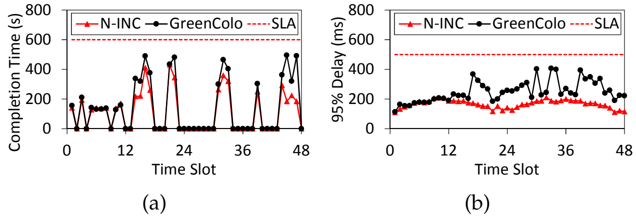


Fig. 10. **(a)** Job turn around time for tenant #1 running Hadoop jobs. **(b)** 95% delay of tenant #2 processing KV workloads. GreenColo reduces carbon footprint without SLA violation.

respectively. We see that GreenColo can greatly reduce the energy consumption and carbon emission compared to N-INC, resulting in an average of 24% reduction. Fig. 9(c) shows the colocation operator's per-slot cost, demonstrating that GreenColo closely follows N-INC to satisfy the budget constraint. This is better illustrated in Fig. 9(d), where we show the cumulative average budget deficit over time. It can be seen that there is (almost) zero budget deficit at the end of the experiment, confirming that GreenColo can successfully satisfy the long-term budget constraint without complete offline information. The experimental results show that GreenColo can be used in real life successfully to reduce carbon emission of the colocation data center without any extra cost to the colocation operator. We also perform an offline simulation with the same settings, models and traces used in the experiment. We observe that the maximum deviation of the experimental results from simulation is less than 4%, which further corroborates our observations and findings in the previous simulation studies.

We now show the tenants' workload performance during runtime in Fig. 10. We see that, using the readily-available power management techniques described above, both tenant #1 and tenant #2 can participate in GreenColo without violating their respective SLA. In other words, GreenColo delivers financial rewards to tenants for "free" under SLA, creating a win-win situation benefiting both colocation operator and tenants. Further, this embodies the great potential of GreenColo in real systems.

7 Related Work

In this section, we discuss the related work from the following perspectives.

• *Data center cost/carbon minimization:* Making data centers cost and/or carbon efficient has been studied by many prior studies [2]–[4], [39]. For example, dynamically scaling server capacity provisioning to strike a balance between energy cost and performance loss has been the primary focus of several recent studies [2], [39]. Extending to a set of geo-distributed data centers, [30], [40] consider geographic load balancing to minimize the electricity cost and [3]–[5] leverage spatio-temporal carbon efficiency to make data centers greener. In addition, data center demand response is also emerging to help stabilize the grid operation (e.g., data center frequency regulation [41], [42]). These studies, however, focus on owner-operated data centers in with operator's full control of server power management, and cannot be directly applied to colocation data centers unless tenants, which manage servers by themselves, are properly incentivized and coordinated.

• *Incentive design:* Incentive design has been successfully applied in various engineering domains, such as time-dependant pricing in wireless networks [43], real-time pricing in smart grid [44], and rebate-based incentive in smart grid [23]. Economics theory has also been applied in computer science, such as auction in Amazon Spot Instance market [21], and market-based scheduling in computer systems [45]. While these works all leverage incentive mechanisms for various purposes, none of them have considered the unique context of colocation whose operator has a natural long-term budgeting constraint and is striving for minimizing carbon footprint.

• *Data center demand response:* Our study can be also viewed as *demand response* within data centers (i.e., using economic incentives to reshaping tenants' demand), which has recently been studied for in various contexts [22], [46]–[51]. For example, [46] and [47] propose pricing-based methods to recoup costs from cloud tenants and fairly split the energy cost, respectively. Among the growing set of papers on colocation demand response, [52] proposes a randomized auction design for emergency demand response (i.e., colocation-wide energy shedding when requested by the utility for grid stability during emergencies), [51] studies fairly incentivizing tenants' energy reduction for emergency demand response, while [22] studies SFB for green-aware colocation demand response under both emergency and economic programs. However, these works often focuses on *one*-step optimization, which cannot be applied or trivially extended to satisfy the long-term budget constraint. While [50] considers online randomized auction for emergency demand response, it studies a very different problem setting of emergency demand response, and the randomized auction mechanism is tailored to the specific problem. Furthermore, we use a different mechanism based on a new variant of SFB, whereas the prior research considers direct pricing [46], [48], auction [50], [52], or a restricted family of parameterized SFB that considers a particular form of supply function [22], [31]. Last but not least, our focus on carbon footprint minimization subject to long-term budget constraint has not been studied in the literature.

Our study also advances [53] by: considering a geo-distributed data centers; providing a thorough evaluation and self-tuning approach; exploiting colocation operator's self-managed servers to further reduce carbon footprint; prototype experiment to validate

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2767043, IEEE Transactions on Cloud Computing

GreenColo.

8 Conclusion

In this paper, we address a critically important problem of reducing carbon footprint of colocations for greenness. We propose a carbon-aware incentive mechanism to break the split-incentive hurdle between colocation operator and tenants. We show that our proposed algorithm, GreenColo, can achieve 18% carbon reduction and save tenants' cost by up to 28%, while the colocation operator does not incur any additional cost. Finally, using a scaled-down testbed experiment, we validate the effectiveness of GreenColo in real life, showing that participating tenants can receive financial rewards without SLA violation and that the colocation can reduce carbon emission by 24% without incurring additional cost.

Appendix
Analytical bounds on GreenColo

We present the performance bound on GreenColo for a given V in Theorem 1, whose proof builds upon sample-path Lyapunov technique [18].

Theorem 1. *For any $T \in \mathbb{Z}^+$ and $H \in \mathbb{Z}^+$ such that $K = HT$, the following statements holds.*

a. *The long-term budget constraint is approximately satisfied with a bounded deviation:*

$$\frac{1}{K}\sum_{k=0}^{K-1}[E(k)+h(k)] \leq \frac{Z}{K}+ \frac{\sqrt{U+D(T-1)+\frac{V}{H}\sum_{h=0}^{H-1}(c_h^*-c_{\min})}}{\sqrt{K}},$$

b. *The average carbon footprint $\bar{c}$ of GreenColo satisfies:*

$$\bar{c} \leq \frac{1}{R}\sum_{h=0}^{H-1} c_h^* + \frac{U+D(T-1)}{V},$$

where U and D are certain finite constants, c_h^* is the minimum average carbon footprint achieved by the optimal algorithm with offline T-slot lookahead information over time slots $hT, hT+1, \cdots, (h+1)T-1$, for $h = 0, 1, \cdots, H-1$, and $c_{\min}$ is the minimum carbon footprint per time slot that can be achieved by any feasible decisions.

Proof of Theorem 1 Here we sketch the key steps for proving Theorem 1. We start with defining a quadratic Lyapunov function $L(q(t)) \triangleq \frac{1}{2}q^2(t)$, where $q(t)$ is our budget deficit queue tracking the backlog. We then derive the maximum Lyapunov drift, i.e., the change in Lyapunov function between time slots. Now, we will see that **P-2** in Algorithm 1 is minimizing the maximum Lyapunov drift plus the carbon emission, i.e., the drift-plus-penalty in Lyapunov optimization. Proceeding from here we can arrive at the two bounds on carbon emission (objective function) and cost budget violation (queue backlog) following similar steps in Chapter 3.2 (pages 37 to 43) and Chapter 4.1 (pages 45 to 51) of [18].

Theorem 1a shows that, the deviation from the long term cost capping increases with increase in V, while Theorem 1b shows that the performance gap between GreenColo and the offline optimum decreases when V becomes larger. We see that, in GreenColo, there exists a trade-off between carbon footprint and budget constraint satisfaction balanced by V. Increase in V reduces the gap in carbon emission between GreenColo and the offline algorithm, but may suffer from larger deviation from long-term budget target, and vice versa. ■

References

[1] J. G. Koomey, "Growth in data center electricity use 2005 to 2010," 2011.
[2] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," in *INFOCOM*, 2011.
[3] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Greening geographical load balancing," in *SIGMETRICS*, 2011.
[4] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," *SIGCOMM Comput. Commun. Rev.*, 2012.
[5] K. Le, R. Bianchini, J. Zhang, Y. Jaluria, J. Meng, and T. D. Nguyen, "Reducing electricity cost through virtual machine placement in high performance computing clouds," in *SuperComputing*, 2011.
[6] Y. Zhang, Y. Wang, and X. Wang, "Greenware: Greening cloud-scale data centers to maximize the use of renewable energy," in *Middleware*, 2011.
[7] D. S. Palasamudram, R. K. Sitaraman, B. Urgaonkar, and R. Urgaonkar, "Using batteries to reduce the power costs of internet-scale distributed networks," in *SoCC*, 2012.
[8] Verizon Terremark, "Pricing plan," http://www.terremark.com/landing-pages/colocation-promo-group3/.
[9] NRDC, "Scaling up energy efficiency across the data center industry: Evaluating key drivers and barriers," *Issue Paper*, Aug. 2014.
[10] Greenpeace, "Clicking clean: Who is winning the race to build a green internet?" 2017.
[11] Y. Sverdlik, "Google to build and lease data centers in big cloud expansion," in *DataCenterKnowledge*, April 2016.
[12] Apple, "Environmental responsibility report," 2016.
[13] DatacenterMap, "Colocation USA," http://www.datacentermap.com/usa/.
[14] Y. Sverdlik, "How edge data center providers are changing the internet's geography," in *DataCenterKnowledge*, August 2015.
[15] U.S. Green Building Council, "Leadership in energy & environmental design," http://www.usgbc.org/leed.
[16] Akamai, "Environmental sustainability policy," http://www.akamai.com/html/sustainability/our_commitment.html.
[17] Apple, "Environmental responsibility report," 2014, http://images.apple.com/environment/reports/docs/Apple_Environmental_Responsibility_Report_2014.pdf.
[18] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.
[19] R. Johari and J. N. Tsitsiklis, "Parameterized supply function bidding: Equilibrium and efficiency," *Oper. Res.*, vol. 59, no. 5, pp. 1079–1089, Sep. 2011.
[20] N. Li, L. Chen, and M. A. Dahleh, "Demand response using linear supply function bidding," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1827–1838, July 2015.
[21] Amazon, "EC2 spot instances," http://aws.amazon.com/ec2/spot-instances/.
[22] N. Chen, X. Ren, S. Ren, and A. Wierman, "Greening multi-tenant data center demand response," in *IFIP Performance*, 2015.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2767043, IEEE Transactions on Cloud Computing

[23] H. Zhong, L. Xie, and Q. Xia, "Coupon incentive-based demand response: Theory and case study," *IEEE Trans. Power Systems*, vol. 28, no. 2, pp. 1266–1276, May 2013.

[24] H. Xu and B. Li, "Reducing electricity demand charge for data centers with partial execution," in *e-Energy*, 2014.

[25] M. A. Islam, H. Mahmud, S. Ren, and X. Wang, "Paying to save: Reducing cost of colocation data center via rewards," in *High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on*. IEEE, 2015, pp. 235–245.

[26] R. H. Inman, H. T. Pedro, and C. F. Coimbra, "Solar forecasting methods for renewable energy integration," *Progress in energy and combustion science*, vol. 39, no. 6, pp. 535–576, 2013.

[27] "California ISO, http://www.caiso.com/."

[28] U. Hoelzle and L. A. Barroso, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan and Claypool Publishers, 2009.

[29] Q. Wu, "Making facebook's software infrastructure more energy efficient with autoscale," 2014.

[30] L. Rao, X. Liu, L. Xie, and W. Liu, "Reducing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment," in *INFOCOM*, 2010.

[31] M. A. Islam, X. Ren, S. Ren, A. Wierman, and X. Wang, "A market approach for handling power emergencies in multi-tenant data center," in *HPCA*, 2016.

[32] Enaxis Consulting, "Pricing data center co-location services," 2009, http://enaxisconsulting.com.

[33] CBRE, "Q4 2013: National data center market update," 2013.

[34] E. Thereska, A. Donnelly, and D. Narayanan, "Sierra: a power-proportional, distributed storage system," *Tech. Rep. MSR-TR-2009-153*, 2009.

[35] G. Urdaneta, G. Pierre, and M. Van Steen, "Wikipedia workload analysis for decentralized hosting," *Computer Networks*, 2009.

[36] J. V. Spadaro, L. Langlois, and B. Hamilton, "Greenhouse gas emissions of electricity generation chains: Assessing the difference," *IAEA bulletin*, vol. 42, no. 2, pp. 19–28, 2000.

[37] D. Bouley, "Estimating a data center's electrical carbon footprint," *Schneider Electric White Paper Library*, 2011.

[38] A. Gandhi, M. Harchol-Balter, R. Raghunathan, and M. A. Kozuch, "Autoscale: Dynamic, robust capacity management for multi-tier data centers," *ACM Trans. Comput. Syst.*, vol. 30, no. 4, pp. 14:1–14:26, Nov. 2012.

[39] B. Guenter, N. Jain, and C. Williams, "Managing cost, performance and reliability tradeoffs for energy-aware server provisioning," in *INFOCOM*, 2011.

[40] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *SIGCOMM*, 2009.

[41] Y. Shi, B. Xu, B. Zhang, and D. Wang, "Leveraging energy storage to optimize data center electricity cost in emerging power markets," in *e-Energy*, 2016.

[42] H. Chen, A. K. Coskun, and M. C. Caramanis, "Real-time power control of data centers for providing regulation service," in *CDC*, 2013.

[43] S. Ha, S. Sen, C. Joe-Wong, Y. Im, and M. Chiang, "Tube: time-dependent pricing for mobile data," in *SIGCOMM*, 2012.

[44] H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 320–331, Dec. 2010.

[45] S. M. Zahedi and B. C. Lee, "Ref: Resource elasticity fairness with sharing incentives for multiprocessors," in *ASPLOS*, 2014.

[46] C. Wang, N. Nasiriani, G. Kesidis, B. Urgaonkar, Q. Wang, L. Y. Chen, A. Gupta, and R. Birke, "Recouping energy costs from cloud tenants: Tenant demand response aware pricing design," in *e-Energy*, 2015.

[47] N. Nasiriani, C. Wang, G. Kesidis, B. Urgaonkar, L. Y. Chen, and R. Birke, "On fair attribution of costs under peak-based pricing to cloud tenants," in *MASCOTS*, 2015.

[48] M. A. Islam, H. Mahmud, S. Ren, and X. Wang, "Paying to save: Reducing cost of colocation data center via rewards," in *HPCA*, 2015.

[49] S. Ren and M. A. Islam, "Colocation demand response: Why do I turn off my servers?" in *ICAC*, 2014.

[50] Q. Sun, S. Ren, C. Wu, and Z. Li, "An online incentive mechanism for emergency demand response in geo-distributed colocation data centers," in *eEnergy*, 2016.

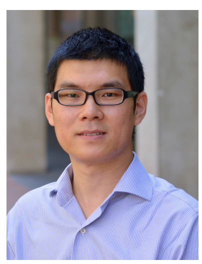
[51] Y. Guo, H. Li, and M. Pan, "Colocation data center demand response using nash bargaining theory," *IEEE Transactions on Smart Grid*, vol. pp, no. 99, p. 1, Januray 2017.

[52] L. Zhang, S. Ren, C. Wu, and Z. Li, "A truthful incentive mechanism for emergency demand response in colocation data centers," in *INFOCOM*, 2015.

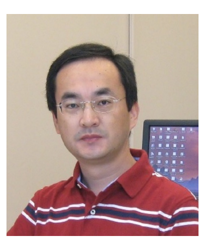
[53] M. A. Islam, S. Ren, and X. Wang, "GreenColo: A novel incentive mechanism for minimizing carbon footprint in colocation data center," in *IGCC*, 2014.

Mohammad A. Islam is doing his Ph.D. in Electrical Engineering at University of California, Riverside. He received his B.Sc. in Electrical and Electronics Engineering from Bangladesh University of Engineering and Technology in January 2008. His research interests include data center resource management, cloud computing, and sustainability for IT.

A. Hasan Mahmud is with Amazon. He received his Ph.D. degree in computer science from Florida International University in 2016. His current research interests include resource management in cloud computing, capacity provisioning and autoscaling of virtualized resources. He has received the Best Paper Award in 8th International Workshop in Feedback Computing, 2013.



Shaolei Ren is an Assistant Professor of Electrical and Computer Engineering at University of California, Riverside. He received his B.E. from Tsinghua University in 2006, M.Phil. from Hong Kong University of Science and Technology in 2008, and Ph.D. from University of California, Los Angeles in 2012, all in electrical and computer engineering. His research interests include cloud computing, data centers, and network economics. He was a recipient of the U.S. NSF Faculty Early Career Development (CAREER) Award in 2015.



Xiaorui Wang received the D.Sc. degree in computer science from Washington University in St. Louis in 2006. He is currently a professor in the Department of Electrical and Computer Engineering at The Ohio State University. He is the recipient of the US Office of Naval Research (ONR) Young Investigator (YIP) Award in 2011 and the US National Science Foundation (NSF) CAREER Award in 2009. He also received the Best Paper Award from the 29th IEEE Real-Time Systems Symposium (RTSS) in 2008. He is an author or coauthor of more than 100 refereed publications. His research interests include computer architecture and systems, data center power management, and cyber-physical systems. He is a member of the IEEE and the IEEE Computer Society.