GreenColo: Incentivizing Tenants for Reducing Carbon Footprint in Colocation Data Centers

Mohammad A. Islam, Hasan Mahmud, Shaolei Ren, Xiaorui Wang, Haven Wang, Joseph Scott

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, colocation suffers from "split incentive": tenants are often charged based on their peak power subscription regardless of their actual energy consumption, and hence they 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 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 resource management, Energy

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], [3], [4], [5], [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 the colocation pricing model varies among different operators as well as geographical locations, a widely-adopted pricing model is based on peak power subscription: tenants are billed for how much peak power they reserve, regardless the actual energy consumption [7], [8].[1] On top of the charge for power reservation, other fees, such as space and network bandwidth charges, may also be applied.

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 of cloud 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 smaller-scale cloud providers (e.g., Salesforce, Box) that are not "large" enough to build megascale data centers on their own. Furthermore, content delivery networks (CDN), that will be processing half of the global Internet traffic by 2018, house their servers in global colocations in proximity to user bases for latency minimization [11]. Colocations therefore provide the indispensable physical support for Internet traffic, and have been keeping a strong momentum to grow. By one estimate [12], there are more than 1,400 colocation data centers in the U.S., and the now U.S.$ 25billion global colocation market

- *This work was supported in part by the NSF under Grant No. CNS-1263124, CNS-1143607, CNS-1423137, and CNS-1551661.*
- *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 Florida International University. E-mail: amahm008@fiu.edu*
- *Xiaorui Wang is with The Ohio State University. E-mail: xwang@ece.osu.edu*
- *Haven Wang was with Cornell University while contributing to this paper. E-mail: hw389@cornell.edu*
- *Joseph Scott was with Ohio University while contributing to this paper. E-mail: js901410@ohio.edu*

1. The power-based pricing is partially due to that the data center power capacity is a deciding factor for sizing the facility infrastructure, e.g., cooling and power supply systems, which constitutes a major fraction of the operator's total cost of ownership.

is expected to grow to U.S.$ 43billion by 2018.

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 for the first time large colocations in its latest data center sustainability report "Clicking Clean" released in 2014, 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 industry 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 [13]) and/or compelled by pro-sustainability tenants such as Apple and Akamai [11], [14]. 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 focus on critical yet long-neglected colocations and 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 [4], [3], they cannot be directly applied to colocations due to the colocation operator's control over tenants' servers. On the other hands, tenants may not be willing to manage their servers for carbon efficiency, due to the widely-adopted pricing model based on peak power subscription that provides no incentives for tenants to save energy.[2]

• 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 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. Working in a "bidding" manner, 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 [15] 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).

The rest of this paper is organized as follows. In Section 2 and 3 we describe the work flow of GreenColo and build the model. In Section 4, we present the problem formulation and develop our online algorithm GreenColo. Section 5 and 6 provides our simulation and experimental results to support our analysis. Related work is reviewed in Section 7 and finally, concluding remarks are offered in Section 8.

2 Overview of GreenColo

We describe the sequence of actions taken by the tenants and the colocation operator for a one-time execution of GreenColo.

• **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:** After receiving the bidding sets from the tenants, the colocation operator inputs these 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

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

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
M_{ij}	No. of server of tenant τ_{ij}
m_{ij}	No. of servers turned off by tenant τ_{ij}
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

to corresponding participating tenants. Finally, the tenants carry out the energy reduction as committed in the winning bids and receive the corresponding rewards.

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. SFB eliminates the need of predicting information on how much demand suppliers can fulfill (as would otherwise be required by pricing-based mechanisms [16], [17]). Some variants of SFB have been applied in various contexts (e.g., hotel and air ticket bidding on Priceline, and power markets [18], [19]).

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 [11], [14].

3 MODELING

In this section, we build formal models to represent 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 $j = 1, 2, \cdots N_i$. Tenant τ_{ij} has M_{ij} servers [3] in data center i. While tenants may use various control knobs (e.g., scaling down CPU frequencies) for energy saving, our study adopts the widely-studied approach "turning unused servers off" as an example [2], [20] to illustrate GreenColo. A variant of this approach, called "Autoscale", has already been used in Facebook's production system for energy saving [21].

Energy consumption and electricity cost. If tenant τ_{ij} turns off m_{ij} servers and each of the servers have a static/idle energy consumption of p_{ij}^s and computing/dynamic energy consumption of p_{ij}^c, we can write the total energy consumption of tenant τ_{ij} as $e_{ij} = (M_{ij} - m_{ij}) \cdot (p_{ij}^s + p_{ij}^c \cdot u_{ij})$, where u_{ij} is the average server utilization of tenants τ_{ij} when it turns of m_{ij} servers. Thus, the total IT energy consumption at data center i can be expressed as

$$P_i^{IT} = \sum_{j=1}^{N_i} (M_{ij} - m_{ij}) \cdot (p_{ij}^s + p_{ij}^c \cdot u_{ij}). \tag{1}$$

The power model in (1) has been widely considered in prior works [2], [3] and shown to be fairly accurate in practical systems [22], [23]. Thus, 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

$$P_i = \left[\gamma_i \cdot P_i^{IT} - r_i\right]^+, \tag{2}$$

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 at data center i, the total electricity cost for the colocation operator can be derived as

$$e = \sum_{i=1}^{L} w_i \cdot P_i = \sum_{i=1}^{L} w_i \cdot P_i = \left[\gamma_i \cdot P_i^{IT} - r_i\right]^+. \tag{3}$$

3. Tenants with heterogeneous servers can be captured in this model by treating them as group of smaller tenants, each with homogeneous servers.

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 [24]. Due to different carbon efficiencies associated with different fuel types, the carbon emission rate of the grid power changes with the fuel mix. We use the following formula to derive the average carbon efficiency (with a unit of g/kWh) [4] at data center location i

$$\phi_i = \frac{\sum[\phi_f \cdot b_{fi}]}{\sum b_{fi}}, \tag{4}$$

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 plant serving data center i. As shown in Fig. 1(c), the fuel mix of power grids exhibits a temporal diversity, as electricity generation from different fuels are continuously regulated in the power grid to maintain balance between supply and demand. Thus, considering ϕ_{r_i} as the carbon emission rate of the onsite renewable energy, data center i's carbon emission is

$$c_i = \phi_i \cdot P_i + \phi_{r_i} \cdot r_i \tag{5}$$

The backup generator used in data centers during grid power failure also have significant carbon footprint because of their use of gasoline. Carbon emission by the backup generators that are not considered in this study can also be captured by (4).

3.2 Tenant

In GreenColo, tenants can voluntarily bid for energy reduction and specify the corresponding incentives they want. As described in the previous section, there is no restriction on how the tenants devise their bids. However, to facilitate the analysis of GreenColo, we provide a specific model that guides tenants to specify their bids. Specifically, we consider that the tenants request their incentives based on the incurred "costs" during energy reduction: *inconvenience cost* and *delay cost*, as detailed below.

Inconvenience cost. We use inconvenience cost to collectively represent 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 cost 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 cost. The tenants turn off servers by consolidating workloads into fewer servers, which may result into delay performance degradation for applications and causing "cost" to tenants [2]. We represent the delay cost of tenant τ_{ij} by d_{ij}, which intuitively increases with number of turned off servers m_{ij} and the tenant' workload arrival rate λ_{ij}. As a concrete example, we employ a widely-applied queuing-theoretic model by considering a M/M/1 queue at each active server [2], [3], [25]. Considering that the total workload is equally distributed among all the active servers, we express the delay cost of tenant τ_{ij} as

$$d_{ij}(m_{ij}, \lambda_{ij}) = \beta_{ij} \cdot \lambda_{ij} \cdot \left(\frac{1}{\mu_{ij} - \frac{\lambda_{ij}}{M_{ij} - m_{ij}}} - d_{ij}^{th} \right)^+, \tag{6}$$

where 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 cost, 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,

$$\frac{M_{ij} - m_{ij}}{\mu_{ij}(M_{ij} - m_{ij}) - \lambda_{ij}} \leq \bar{d}_{ij}^{\max}. \tag{7}$$

The maximum delay constraint in (7) 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 (6) provides a tenable estimation of the resulting delay performance and hence is widely used for performance analysis[2], [25].

By combining both delay and inconvenience costs, the total cost of turning off $m_{i,j}$ servers for tenant τ_{ij} is expressed as

$$h_{ij}(m_{ij}) = \eta_{ij} \cdot m_{i,j} + d_{ij}(m_{ij}, \lambda_{ij}). \tag{8}$$

Hence, we can express bidding set of tenant τ_{ij}

$$\mathcal{B}_{ij} \in \{(m_{ij}, h_{ij}(m_{ij})) | \text{ (7) is satisfied}\}. \tag{9}$$

Note that tenants have the freedom to choose very high payment bids which, however, are more likely to be rejected by the colocatino operator, and consequently, tenants receive no rewards. This is analogous to the case of Priceline, where very low bids on hotels/flights submitted by over-greedy users will be turned down. If m_{ij} is the number of tenant τ_{ij}'s servers turned off as decided by GreenColo, the total incentive payout by the colocation operator is

$$h = \sum_{i=1}^{L} \sum_{j=1}^{N_i} h(m_{ij}), \tag{10}$$

which constitutes part of the colocation operator's cost (in addition to electricity cost).

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. int.

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:

$$\textbf{P-1}: \quad \min_{\text{bids}} \ \bar{c} = \frac{1}{K}\sum_{k=0}^{K-1}\sum_{i=1}^{L} c_i(k) \tag{11}$$

$$\text{s.t.} \quad \sum_{k=0}^{K-1}[e(k)+h(k)] \le Z, \tag{12}$$

$$[m_{ij}(k), h_{ij}(m_{ij}(k))] \in \mathcal{B}_{ij}(k), \ \forall\, i,j,k. \tag{13}$$

where the objective is to minimize the long-term average carbon footprint, the constraint (12) is the long-term operational cost which consists of the electricity cost and total incentive paid to tenants. The second constraint (13) 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 (12) 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 solution with future information. Next, we present GreenColo and its operation principle.

4.2 GreenColo

Based the extended Lyapunov technique [15], 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 (12), 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

$$q(k+1) = \left[q(k) + e(k) + h(k) - \frac{Z}{K}\right]^{+}, \tag{14}$$

Algorithm 1 GreenColo

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

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

  subject to constraint (13)
  Update $q(k+1)$ according to (14)
  if $k \mod \vartheta = 0$ **then**
    $z_k = \frac{Z}{K} - \frac{1}{k}\sum_{i=0}^{k}[e(i) + h(i)]$
    $V = \max\{V + \alpha \cdot z_k, V_{min}\}$
  end if
end for

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 $V \ge V_{min} > 0$ plus the operational cost multiplied by the budget deficit queue shown in (15). 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 chose/find the appropriate V in practice without the complete offline information [15]. 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 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

$$e_i^{IT} = \sum_{j=1}^{N_i} (M_{ij} - m_{ij}) \cdot (p_{ij}^s + p_{ij}^c \cdot u_{ij}) + (M_{io} - m_{io}) \cdot (p_{io}^s + p_{io}^c \cdot u_{io}) \quad (15)$$

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

$$d_{io}^{avg}(k) \leq d_{io}^{\max}, \forall i, k \quad (16)$$

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.

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

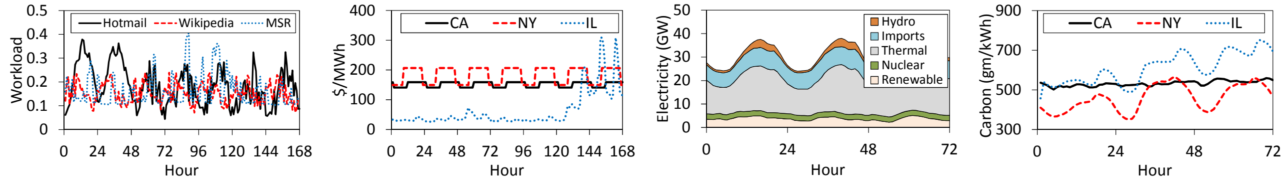
	Tenant 1	Tenant 2	Tenant 3
Delay cost β (¢/s/job)	0.015	0.01	0.005
η ($ per server-hour)	0.03	0.03	0.03
Power cost ($/kW/month)	145	145	145
Service rate (jobs/hour)	360,000	180,000	30
Soft threshold on avg. delay	10 ms	20 ms	150 s
Avg. delay constraint	20 ms	30 ms	250 s

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 results.

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 [12]. We consider that each data center houses three participating tenants, which have 2,000 servers each. The small number of tenants is applicable for 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



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

Fig. 1. Trace data.

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 [26], and 145 U.S.\$/kW/month in all of the three data centers, which is a fair market value [8], [27]. 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.

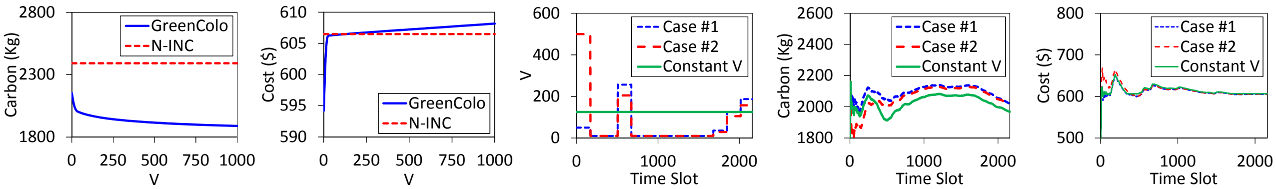
•**Workload.** We use three different workload traces for the tenants. The workload identified as "Hotmail" is taken from a 48-hour trace of 8 servers of Hotmail [28]. "Wikipedia" traces are taken from [29], which contain 10% of all user requests issued to Wikipedia from a 30-day period of September 2007, and "MSR" workload is a 1-week I/O trace of 6 RAID volumes at Microsoft Research Cambridge [28]. 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. These 3-month trace are then further randomized by up to 20% and used for totally 9 tenants in the three data centers. 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 [24] 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 [24], and use carbon efficiency for fuel types presented in [30] 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 [31], [4]. 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



(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.

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

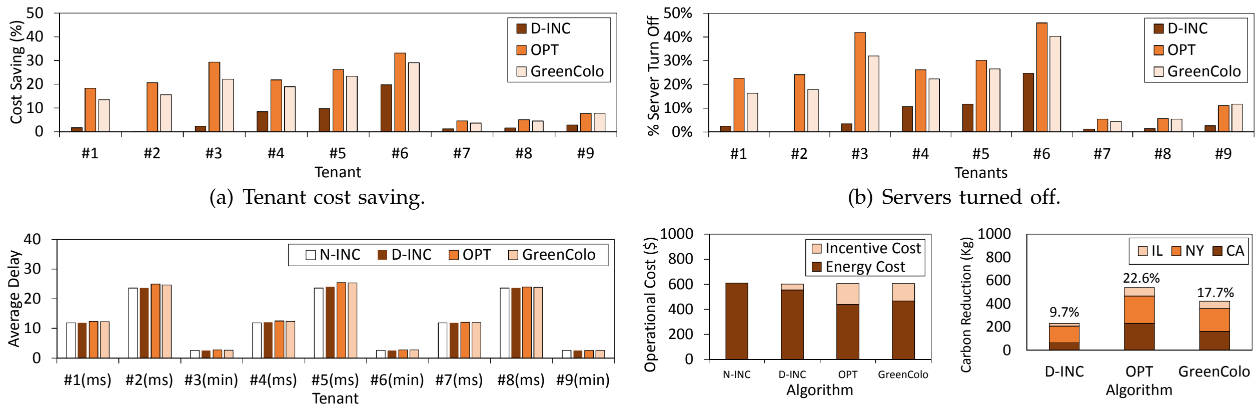
We first show the impact of control parameter V on the performance of GreenColo in Fig. 2(a) and Fig. 2(b). It can be seen that V governs the trade-off between carbon footprint reduction and budget constraint satisfaction: when V increases, GreenColo focuses more on reducing carbon footprint while caring 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.

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



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

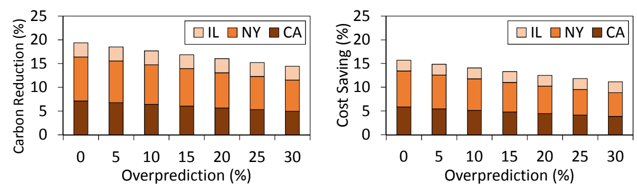
(c) Delay performance. (d) Operational cost. (e) Carbon reduction.

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%.

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

Insensitivity against tenants' workload overprediction. GreenColo relies on energy reduction bids submitted by tenants themselves. In order to determine their desired number of servers to be turned off while satisfying delay performance constraint, tenants need to estimate incoming workloads over the next hour. However, it may not be always possible to accurately predict the incoming workload arrival rates. To cope with unexpected possible traffic spikes, tenants can deliberately overestimate the workload arrival rate by a certain overprediction factor $\psi \geq 0$: the higher ψ, the more overprediction. Intuitively, when tenants are more conservative and tend more to overpredict workloads, fewer number of servers will be turned

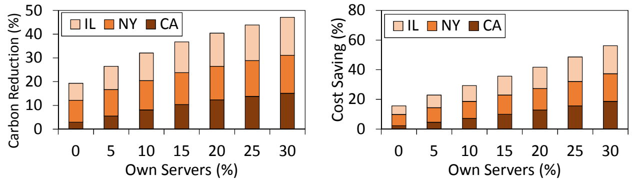


(a) Carbon reduction. (b) Tenant cost saving.

Fig. 4. Impact of tenants' workload overprediction. Carbon reduction and cost savings are not significantly impacted for even at 30% overprediction.

off. Fig. 4 shows that even when tenants overestimate the workloads (to leave a margin for traffic spikes) by 30%, which is already quite high [2], the carbon reduction and cost saving are not significantly compromised. The bars in Fig. 4(a) (as well as other carbon reduction figures in this section) represents the total carbon reduction of the three data center with the stacks representing contributions from different data centers.

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 along with the tenants. We vary the percentage of self-managed server in each data center from 0 to 30%. We also re-scale the number of tenant servers, such that the total number of servers (as well as the data center peak power) remains same. We consider that the self-managed servers process delay-sensitive jobs (like Tenants #1, #2 and #3), and use the Hotmail trace as workloads after adding 30% random variations to the original trace. 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. 5, we see that both carbon reduction and tenants' cost savings increase with increase in



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

Fig. 5. 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.

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.

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.

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

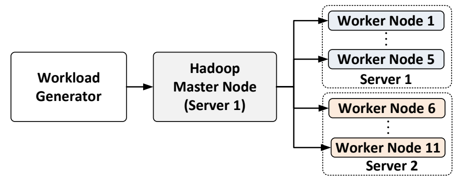


Fig. 6. Tenant #1 processing Hadoop workload. The Hadoop setup has one master node and eleven worker nodes in two servers.

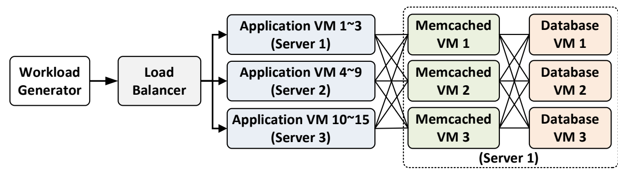
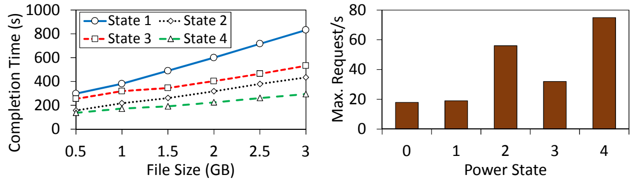


Fig. 7. Tenant #2 processing KV workload. The multi-tiered KV system consists of one front end load balancer, fifteen application nodes, three memory cache nodes and three database nodes in three servers.

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.

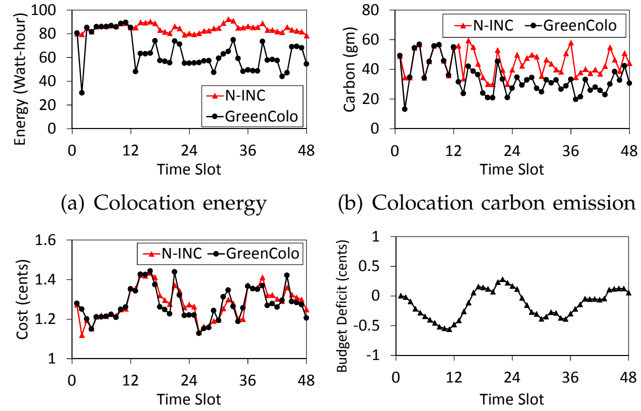
Tenant #2. Using the remaining three servers as illustrated in Fig. 7, tenant #2 processes key-value-store (KV) workload, which resembles a multi-tiered web service such as social networking [32]. Our implementation of Key-Value store has 4 tiers: front-end load balancer, application, memory cache, and 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



(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.



(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.

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.

6.2 Power management

Both tenants can adjust servers' power states and also scale the number VMs for energy saving, subject to SLA. In particular, tenant #1 running Hadoop jobs has a SLA of 10 minutes for the maximum job completion time, while tenant #2 running interactive jobs has a SLA requirement of 500ms on 95-percentile delay which is a reasonable setting as considered in prior research [32]. Tenant #1 can scale the number of worker VMs for its workloads on two servers, and tenant #2 can adjust the number of its application-tier VMs on three servers. We use three different power states for each server: High Performance (H), Low Performance (L), and Turned Off (Z). Tenants' processing capabilities under different combinations of power states are shown in Fig. 8. The power state and number of VMs can be adjusted in accordance to incoming workload arrival rates subject to SLA, as similarly adopted by Facebook's production systems [21]. 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 SAL 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.

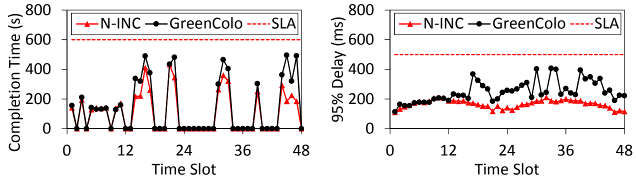
We first show the energy consumption and corresponding carbon emission in Figs. 9(a) and 9(b), respectively. We see that GreenColo can greatly reduce the energy consumption and carbon emission compared to N-INC, resulting in an average of 24% reduction. Fig. 9(c) shows the colocation operator's per-slot 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 [33], [2], [4], [3]. For example,



(a) (b)

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.

dynamically scaling server capacity provisioning to strike a balance between energy cost and performance loss has been the primary focus of several recent studies [33], [2]. Extending to a set of geo-distributed data centers, [22], [25] consider geographic load balancing to minimize the electricity cost and [4], [3], [5] leverage spatio-temporal carbon efficiency to make data centers greener. These studies, however, focus on owner-operated data centers in which servers' power management can be performed at the data center operator's discretion. Thus, while the technological advances made by these studies are appealing, they cannot be directly applied to colocation data centers unless tenants, which manage servers by themselves, are properly incentivized.

• *Incentive design and data center demand response:* Incentive design has been successfully applied in various engineering domains, such as time-dependant pricing in wireless networks [34], real-time pricing in smart grid [35], and rebate-based incentive in smart grid [36]. Economics theory has also been applied in computer science, such as bidding/auction in online sponsored advertisement market [37], auction-based Amazon Spot Instance market [38], and market-based scheduling in computer systems [39], [40], [41]. 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.

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 such as cloud computing and colocations [42], [43], [44], [45], [17], [46]. For example, [42] proposes a dynamic pricing-based method to recoup costs from cloud tenants, and among the handful of recent works on colocation, [44] proposes a randomized auction design for emergency demand response, while [43] 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. Furthermore, we use a different mechanism based on a new variant of SFB, whereas the prior research considers direct pricing [42], [17], auction [44], or a restricted family of parameterized SFB that considers a particular form of supply function [43]. Last but not least, our focus on carbon footprint minimization subject to long-term budget constraint has been rarely studied in the literature. Our study also advances [47] by: considering a geo-distributed data centers; providing a thorough evaluation and self-tuning approach to dynamically adjusting parameter V for budget constraint; exploiting colocation operator's self-managed servers to further reduce carbon footprint; prototype experiment to validate GreenColo.

In sum, this paper addresses a unique problem of reducing carbon footprint of colocation data centers for greenness, a long-neglected driving factor for sustainable computing. To our best knowledge, our work represents the first effort in greening geo-distributed colocations subject to long-term budget constraint, by breaking the split-incentive hurdle between the colocation operator and tenants.

8 Conclusion

In this paper we addressed a critically important problem of reducing carbon footprint of colocations for greenness by proposing an 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 validated 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 [15]. The proof of Theorem 1 is put online [48] to save space.

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 and $c_{\min}$ is the minimum carbon footprint per time slot that can be achieved by any feasible decisions.

Theorem 1 provides the theoretical performance bound on GreenColo. The considered T-slot look ahead algorithm represents a class of algorithms, parameterized by the information prediction window size $T \leq K$. We see from Δ_{cost} that, the deviation from the long term cost capping increases with increase in V, which is intuitive as large V means there is more focus on the original objective (minimizing carbon emission) than the cost capping. Δ_{carbon} specifies the performance gap between GreenColo and optimal offline solution in terms of carbon minimization, showing that the gap is decreases when V becomes larger. We see that, in GreenColo, there exists a tradeoff 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. This theoretical observation will be further substantiated in the simulation section. ■

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 *IEEE 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 *Super-Computing*, 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: How companies are creating the green internet," 2014.

[11] Akamai, "Environmental sustainability policy," http://www.akamai.com/html/sustainability/our_commitment.html.

[12] DatacenterMap, "Colocation USA," http://www.datacentermap.com/usa/.

[13] U.S. Green Building Council, "Leadership in energy & environmental design," http://www.usgbc.org/leed.

[14] Apple, "Environmental responsibility report," 2014, http://images.apple.com/environment/reports/docs/Apple_Environmental_Responsibility_Report_2014.pdf.

[15] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.

[16] 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 *eEnergy*, 2015.

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

[18] R. Johari and J. N. Tsitsiklis, "Parameterized supply function bidding: Equilibrium and efficiency," *Oper. Res.*, vol. 59, no. 5, pp. 1079–1089, Sep. 2011.

[19] N. Yu, T. Wei, and Q. Zhu, "From passive demand response to proactive demand participation," in *IEEE Intl. Conf. Auto. Sci. Eng. (CASE)*, 2015.

[20] 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.

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

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

[23] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *ISCA*, 2007.

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

[25] 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.

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

[27] CBRE, "Q4 2013: National data center market update," 2013, http://www.cbre.us/services/office/data-center-solutions-group/.

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

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

[30] 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.

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

[32] 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.

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

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

[35] 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.

[36] 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.

[37] E. Even Dar, V. S. Mirrokni, S. Muthukrishnan, Y. Mansour, and U. Nadav, "Bid optimization for broad match ad auctions," in *WWW*, 2009.

[38] "Amazon EC2 Spot Instances," http://aws.amazon.com/ec2/spot-instances/.

[39] M. Taifi, "Banking on decoupling: Budget-driven sustainability for hpc applications on auction-based clouds," *SIGOPS Oper. Syst. Rev.*, vol. 47, no. 2, pp. 41–50, Jul. 2013.

[40] S. M. Zahedi and B. C. Lee, "Ref: Resource elasticity fairness with sharing incentives for multiprocessors," in *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS, 2014.

[41] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, "Dominant resource fairness: Fair allocation of multiple resource types," in *NSDI*, 2011.

[42] 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.

[43] N. Chen, X. Ren, S. Ren, and A. Wierman, "Greening multi-tenant data center demand response," in *IFIP Performance*, 2015.

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

[45] C. Wang, B. Urgaonkar, G. Kesidis, U. V. Shanbhag, and Q. Wang, "A case for virtualizing the electric utility in cloud data centers," in *HotCloud*, 2014.

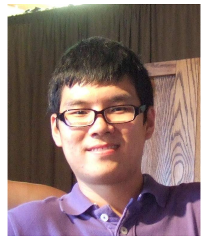
[46] S. Ren and M. A. Islam, "Colocation demand response: Why do I turn off my servers?" in *ICAC*, 2014.

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

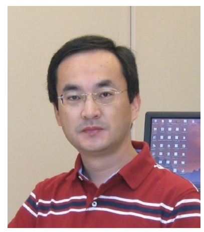
[48] "Online material," https://sites.google.com/site/onlinematerials1/.

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. He is a member of the Sustainable Computing Group (SCG) at UCR led by Dr. Shaolei Ren. His research interests include data center resource management, cloud computing, and sustainability for IT.

A. Hasan Mahmud is a Ph.D. student in the School of Computing and Information Sciences at Florida International University, Miami, Florida. He received his B.Sc. degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in January 2008. Before joining Florida International University, he spent 3.5 years in a leading software company in Bangladesh. He is currently working in the Sustainable Computing Group (SCG) at Florida International University, led by Dr. Shaolei Ren. 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.



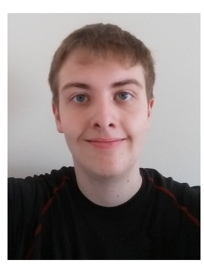
Shaolei Ren received his B.E., M.Phil. and Ph.D. degrees, all in electrical engineering, from Tsinghua University in 2006, Hong Kong University of Science and Technology in 2008, and University of California, Los Angeles, in 2012, respectively. From 2012 to 2015, he was with Florida International University as an Assistant Professor. Since July 2015, he has been an Assistant Professor at University of California, Riverside. His research interests include power-aware computing, data center resource management, and network economics.



Xiaorui Wang received the D.Sc. degree in computer science from Washington University in St. Louis in 2006. He is currently an associate 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.

PLACE PHOTO HERE

Haven Wang was with Cornel University. He participated in the National Science Foundation (NSF) sponsored REU program, and spent Summer'14 at Florida International University.



Joseph Scott received his bachelor degrees in mathematics and computer science from Ohio University in 2015. He is currently a graduate student at the University of Waterloo. While at Ohio University, he participated in National Science Foundation (NSF) sponsored REUs at Florida International University and Ohio University. At Ohio University, he received the Outstanding Junior of Computer Science award, and was then later recognized as the Most Outstanding Senior of the Russ College of Engineering. He is a member of ACM and IEEE.