Paying to Save: Reducing Cost of Colocation Data Center via Rewards

Mohammad A. Islam Hasan Mahmud Shaolei Ren Xiaorui Wang†
Florida International University †The Ohio State University

***Abstract*—Power-hungry data centers face an urgent pressure on reducing the energy cost. The existing efforts, despite being numerous, have primarily centered around owner-operated data centers (e.g., Google), leaving another critical data center segment — colocation data center (e.g., Equinix) which rents out physical space to multiple tenants for housing their own servers — much less explored. Colocations have a major barrier to achieve cost efficiency: server power management by individual tenants is uncoordinated. This paper proposes RECO (REward for COst reduction), which shifts tenants' power management from uncoordinated to coordinated, using financial reward as a lever. RECO pays (voluntarily participating) tenants for energy reduction such that the colocation operator's overall cost is minimized. RECO incorporates the time-varying operation environment (e.g., cooling efficiency, intermittent renewables), addresses the peak power demand charge, and also proactively learns tenants' unknown responses to the offered reward. RECO includes a new feedback-based online algorithm to optimize the reward without far future offline information. We evaluate RECO using both scaled-down prototype experiments and simulations. Our results show that RECO is "win-win" and can successfully reduce the colocation operator's overall cost, by up to 27% compared to the no-incentive baseline case. Further, tenants receive financial rewards (up to 15% of their colocation costs) for "free" without violating Service Level Agreements.**

I. Introduction

Power-hungry data centers have been quickly growing to satiate the exploding information technology (IT) demands. In the U.S., electricity consumption by data centers in 2013 reached 91 billion kilo-watt hours (kWh) [34]. The rising electricity price has undeniably placed an urgent pressure on optimizing data center power management. The existing efforts, despite numerous, have centered around owner-operated data centers (e.g., Google), leaving another data center segment — colocation data center (e.g., Equinix) — much less explored.

Colocation data center, simply called "colocation" or "colo", rents out physical space to multiple tenants for housing their own physical servers in a shared building, while the colocation operator is mainly responsible for facility support (e.g., power, cooling). Thus, colocation significantly differs from owner-operated data centers where operators fully manage both IT resources and data center facilities.

Colocation offers a unique data center solution to a wide range of tenants (as shown in Fig. 1), including financial industries, medium cloud providers (e.g., Salesforce) [12], [33], top-brand websites (e.g., Wikipedia) [1], content delivery providers (e.g., Akamai) [33], and even gigantic firms such as Amazon [8]. The U.S. alone has over 1,200 colocations, and many more are being constructed [13]. According to a Google study [10], "*most large data centers* are built to host servers from multiple companies (often called colocation, or 'colos')."

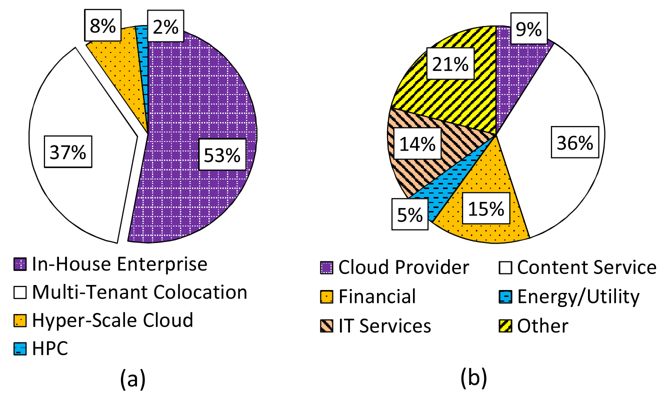


Fig. 1: **(a)** Estimated electricity usage by U.S. data centers in 2011 (excluding small server closets and rooms) [34]. **(b)** Colocation revenue by vertical market [14].

The global colocation market, currently worth U.S.$25 billion, is projected to U.S.$43 billion by 2018 [3]. Excluding tiny-scale server rooms/closets, colocation consumes 37% of the electricity by all data centers in the U.S. (see Fig. 1), much more than hyper-scale cloud data centers (e.g., Amazon) which only take up less than 8%. Hence, it is at a critical point to make colocations more energy-efficient and also reduce their electricity costs. Towards this end, however, there is a barrier as identified below: "uncoordinated" power management.

A vast majority of the existing power management techniques (e.g., [28], [47]) require that data center operators have full control over IT computing resources. However, colocation operator lacks control over tenants' servers; instead, tenants individually manage their own servers and workloads, without coordination with others. Furthermore, the current pricing models that colocation operator uses to charge tenants (e.g., based on power subscription [15], [35]) fail to align the tenants' interests towards reducing the colocation's overall cost. We will provide more details in Section II-B. Consequently, colocation operator incurs a high energy consumption as well as electricity cost.

In this paper, we study a problem that has been long neglected by the research community: "*how to reduce the colocation's operational expense (OpEx)?*" Throughout the paper, we also use "cost" to refer to OpEx wherever applicable. Because of the uniqueness of multi-tenant colocation, our work is distinctly different from a majority of the prior research that concentrates on owner-operated data centers (e.g., Google). We propose RECO (REward for COst reduction), using financial reward as a lever to shift power management in a colocation from uncoordinated to coordinated. RECO pays participating tenants for energy saving at a time-varying reward rate ($ per kWh reduction) such that the colocation operator's overall cost (including electricity cost and rewards to tenants) is minimized. RECO has a voluntary nature, without enforcing tenants to participate. Next, we highlight key challenges for optimizing the reward rate offered to tenants.

Time-varying operation environment. Outside air temperature changes over time, resulting in varying cooling efficiency. Further, on-site solar energy, if applicable, is also highly intermittent, thus calling for a dynamic reward rate to best reflect the time-varying operation environment.

Peak power demand charge. Peak power demand charge, varied widely across utilities (e.g., the maximum power demand measured over each 15-minute interval), may even take over 40% of colocation operator's total electricity bill [31], [45], [49]. Nonetheless, peak power demand charge cannot be perfectly known until the end of a billing cycle, whereas the colocation operator needs to dynamically optimize reward rate without complete offline information.

Tenants' unknown responses to rewards. Optimizing the reward rate offered to incentivize tenants' energy reduction requires the colocation operator to know how tenants would respond. However, tenants' response information is absent in practice and also time-varying.

RECO addresses the above challenges. It models time-varying cooling efficiency based on outside ambient temperature and predicts solar energy generation at runtime. To tame the peak power demand charge, RECO employs a feedback-based online optimization by dynamically updating and keeping track of the maximum power demand as a runtime state value. If the new (predicted) power demand exceeds the current state value, then additional peak power demand charge would be incurred, and the colocation operator may need to offer a higher reward rate to incentivize more energy reduction by tenants. RECO also encapsulates a learning module that uses a parametric learning method to dynamically predict how tenants respond to colocation operator's reward.

We evaluate RECO using both scaled-down prototype experiments and simulations. Capturing a small-scale multi-tenant environment, our prototype experiment demonstrates that RECO is "win-win" and reduces the cost by over 10% compared to no-incentive baseline, while tenants receive financial rewards for "free" without violating their respective Service Level Agreements (SLA). Complementing the experiment, our simulation shows that RECO can reduce the colocation operator's overall cost by up to 27% compared to the no-incentive baseline case. Moreover, using RECO, tenants can reduce their costs by up to 15%. We also subject RECO to a varying environment, showing that RECO can robustly adapt to changes in tenants' responses.

To sum up, our key contribution is uniquely identifying and formulating the problem of reducing colocation's operational cost, which has not been well addressed by prior research. We also propose RECO as a lever to overcome uncoordinated power management and effectively reduce colocation's overall cost, as demonstrated using prototype experiments and simulations.

Data Center Location	Power Utility (Rate Schedule)	Demand Charge	Energy Charge	Demand Charge (% of total)
Phoneix, AZ	APS (E-35)	186,400	253,325	42.39%
Ashburn, VA	Dominion (GS-4)	153,800	207,360	42.59%
Chicago, IL	ComED (BESH)	110,000	276,480	28.46%
San Jose, CA	PG&E (E-20)	138,100	332,398	29.35%
New York, NY	ConEd (SC9-R2)	314,400	1,099,008	22.24%

TABLE I: A 10MW data center's electricity cost for selected locations (in U.S. dollars).

II. Preliminaries

A. Peak power demand charge

As a large electricity customer, colocation operator is charged by power utilities not only based on energy consumption, but also based on peak power demand during a billing cycle, and such peak power demand charge is widely existing (e.g., all the states in the U.S.) [31], [40], [45], [49]. Peak power demand charge is imposed to help power utilities recover their huge investment/costs to build and maintain enough grid capacities for balancing supply and demand at any time instant. The specific charge for peak power demand varies by power utilities. For example, some are based on the maximum power measured over each 15-minute interval, while others are based on two "peak power demands" (one during peak hours, and the other one during non-peak hours).

Next, we consider a data center with a peak power demand of 10MW and an almost "flat" power usage pattern (by scaling UPS-level measurements at Verizon Terremark's NAP data center during September, 2013, shown in Fig. 2). Table I shows the data center's monthly cost for selected U.S. data center markets. It can be seen that, as corroborated by prior studies [31], [45], [49], peak power demand charge can take up over 40% of the total energy bill, highlighting the importance of reducing the peak power demand for cost saving.

B. Limitations of colocation's current pricing models

There are three major pricing models in colocations [15], [17], as shown below. Pricing for bandwidths and other applicable add-on services is not included.

Space-based. Some colocations charge tenants based on their occupied room space, although space-based pricing is getting less popular due to increasing power costs [15], [17].

Power-based. A widely-adopted pricing model is based on power subscription regardless of actual energy usage (i.e., the amount of power reserved from the colocation operator before tenants set up their server racks, not the actually metered peak power). In the U.S., a fair market rate is around 150-200$/kW per month [11], [17], [44].

Energy-based. Energy-based pricing charges tenants based on their actual energy usage and is indeed being adopted in some colocations [15], [17]. This pricing model is more common in "wholesale" colocations serving large tenants, typically each having a power demand in the order of megawatts. In addition to energy usage, tenants are also charged based on power subscription (but usually at a lower rate than pure power-based pricing), because colocation operator needs to provision expensive facility support (e.g., cooling capacity,

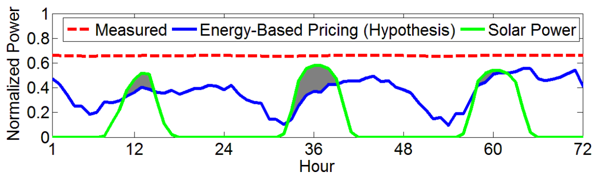


Fig. 2: Normalized power consumption of Verizon Terremark's colocation in Miami, FL, measured at UPS output from September 15–17, 2013.

power distribution) based on tenants' power reservation to ensure a high reliability.

Clearly, under both space-based and power-based pricing, tenants have little incentive to save energy. We show in Fig. 2 the power consumption of Verizon Terremark's colocation in Miami, FL, measured at the UPS output (excluding cooling energy) from September 15–17, 2013, and further normalized with respect to the peak IT power to mask real values. Verizon Terremark adopts a power-based pricing [44]. It can be seen that the measured power is rather flat, because of two reasons: (1) tenants' servers are always "on", taking up to 60% of the peak power even when idle [10]; and (2) the average server utilization is very low, only around 10-15%, as consistent with other studies [21], [29], [34].

Even under energy-based pricing, tenants still have no incentives to coordinate their power management for reducing colocation operator's electricity cost. For example, with intermittent solar energy generation available on-site (which is becoming widely popular [9], [18]), the colocation operator desires that tenants defer/schedule more workloads to times with more solar energy (i.e., "follow the renewables") for maximizing the utilization of renewables and reducing the cost, but tenants have no incentives to do so. For *illustration* purposes, we consider a *hypothesis* scenario by supposing that Verizon Terremark employs energy-based pricing. We extract the variations in measured power usage, and then scale the variations to demonstrate the situation that tenants are saving their energy costs via energy reduction (e.g., "AutoScale" used in Facebook to dynamically scale computing resource provisioning [47]). Fig. 2 shows that the intermittent solar power may be wasted (shown as shaded area), because of the mis-match between solar availability and tenants' power demand. Further, tenants do not have incentives to avoid coinciding their own peak power usage with others, potentially resulting in a high colocation-level peak power usage.

Note that it is not plausible to simply adopt a utility-type pricing model, i.e., tenants are charged based on "energy usage" and "metered peak power" (not the pre-determined power subscription). While this pricing model encourages certain tenants to reduce energy and also flatten their *own* power consumption over time, some tenants (e.g., CDN provider Akamai) have time-varying delay-sensitive workloads that cannot be flattened. Further, time-varying cooling efficiency and intermittent solar energy, if applicable, desire a power consumption profile (e.g., "follow the renewables") that may not be consistent with this pricing model.

To sum up, to minimize colocation operator's total cost, we need to overcome the limitations associated with the current pricing models in colocations and dynamically coordinate power management among individual tenants.

III. Mechanism and Problem Formulation

This section presents RECO, using reward as a lever for coordinating tenants' power consumption. We first describe the mechanism and then formalize the cost minimizing problem.

A. Mechanism

Widely-studied dynamic pricing (e.g., in smart grid [32]) *enforces* all tenants to accept time-varying prices and hence may not be suitable for colocations where tenants sign long-term contracts [51]. Here, we advocate a *reward*-based mechanism: **first**, colocation operator proactively offers a reward rate of r \$/kWh for tenants' energy reduction; **then**, tenants *voluntarily* decide whether or not to reduce energy; **last**, participating tenants receive rewards for energy reduction (upon verification using power meters), while non-participating tenants are not affected.

When offered a reward, participating tenants can apply various energy saving techniques as studied by prior research [19], [28], [47]. For example, a tenant can estimate its incoming workloads and then dynamically switch on/off servers subject to delay performance requirement. This technique has been implemented in real systems (e.g., Facebook's AutoScale [47]) and is readily available for tenants' server power management.

B. Problem formulation

We consider a discrete-time model by dividing a billing cycle (e.g., one month) into T time slots, as indexed by $t = \{0, 1, \cdots, T-1\}$. We set the length of each time slot to match the interval length that the power utility uses to calculate peak power demand (e.g., typically 15 minutes) [5], [31], [45], [49]. At the beginning of each time slot, colocation operator updates the reward rate $r(t)$ for energy reduction (with a unit of dollars/kWh). Then, tenants voluntarily decide if they would like to reduce energy for rewards. As discussed in Section IV-C, the amount of energy reduction by a tenant during a time slot is measured by comparing with a pre-set reference value for that tenant.

We consider a colocation data center with N tenants. At any time slot t, for a reward rate of $r(t)$, we denote the total energy reduction by tenants as $\Delta E(r(t), t)$, where the parameter t in $\Delta E(\cdot, t)$ indicates that the tenants' response to offered reward is time-varying (due to tenants' changing workloads, etc.). We denote the reference energy consumption by tenant i as $e_i^o(t)$. Thus, the total energy consumption by tenants' servers at time t can be written as

$$E(r(t), t) = \sum_{i=1}^{N} e_i^o(t) - \Delta E(r(t), t). \tag{1}$$

Considering electricity price of $u(t)$, power usage effectiveness of $\gamma(t)$ (PUE, ratio of total data center energy to

IT energy) and solar energy generation of $s(t)$, colocation operator's electricity cost and reward cost at time slot t are

$$C_{energy}(r(t),t) = u(t) \cdot [\gamma(t) \cdot E(r(t),t) - s(t)]^+, \quad (2)$$
$$C_{reward}(r(t),t) = r(t) \cdot \Delta E(r(t),t), \quad (3)$$

where $[\ .\ ]^+ = \max\{\ .\ ,0\}$ indicates that no grid power will be drawn if solar energy is already sufficient. Following the literature [31], we consider a zero-cost for generating solar energy, but (2) is easily extensible to non-zero generation cost.

The colocation pays for its peak energy demand during a billing cycle. Power utilities may impose multiple peak demand charges, depending on time of occurrence. For J types of peak demand charges, we use A_j for $j = \{1, 2, \cdots, J\}$ to denote the set of time slots during a day that falls under time intervals related to the j-th type of demand charge. Utilities measure peak demand by taking the highest of the average power demand during pre-defined intervals (usually 15 minutes) over the entire billing period. We write the peak demand charge as follows

$$C_{demand} = \sum_{j=1}^{J} d_j \cdot \frac{\max_{t \in A_j} [\gamma(t) \cdot E(r(t),t) - s(t)]^+}{\Delta t}, \quad (4)$$

where $E(r(t),t)$ is servers' energy consumption given in (1), Δt is the duration of each time slot, and d_j is the charge for type-j peak demand (e.g., $\sim$ 10$ per kW [31], [40], [49]).

Next, we present the colocation operator's cost minimizing problem (denoted as **P-1**) as follows

$$\min \sum_{t=0}^{T-1} [C_{energy}(r(t),t) + C_{reward}(r(t),t)] + C_{demand}.$$

Solving **P-1** and obtaining the optimal reward rate $r(t)$ faces unknown and uncertain offline information. First, cooling efficiency (which significantly affects PUE) and solar energy generation both vary with the outside environment. Second, the total cost contains demand charge which is only determined after a billing cycle. Last but not least, tenants' response to reward (i.e., how much energy tenants reduce) is unknown. The next section will address these challenges.

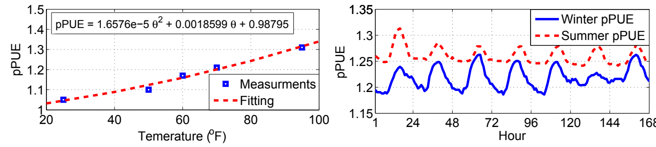
IV. RECO: Reducing Cost via Rewards

This section presents how RECO copes with three major technical challenges in optimizing the reward: time-varying environment, peak power demand charge that will not be perfectly known until the end of a billing cycle, and tenants' unknown responses. Then, we show the algorithm for executing RECO at runtime.

A. Modeling cooling efficiency and solar energy

Now, we provide details for modeling time-varying cooling efficiency and predicting on-site solar energy generation.

Cooling efficiency. Cooling energy is a non-trivial part of data center's total energy usage [48]. Data centers, including colocations, may improve cooling energy efficiency using air-side economizer (i.e., outside cold air for cooling).



(a) (b)

Fig. 3: **(a)** pPUE variation with outside ambient temperature [16], [48]. **(b)** Snapshot of weekly pPUE during Summer and Winter in San Francisco, CA, in 2013.

As a concrete example, we model the cooling energy efficiency based on a commercially-available cooling system manufactured by Emerson Network Power [16], [48]. This cooling system operates in three different modes: *pump*, *mixed* and *compressor*. Given a return air temperature of 85^oF, it runs in the pump mode for ambient temperature lower than 50^oF. It runs in the mixed mode for ambient temperature between 50^oF and 60^oF, and in the compressor mode when ambient temperature exceeds 60^oF. Based on manufacture-reported measurements, we model partial PUE (pPUE) as[1]

$$\text{pPUE} = 1.6576 \times 10^{-5}\theta^2 + 0.0018599\theta + 0.98795, \quad (5)$$

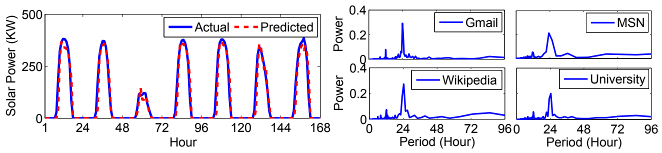
where θ is the ambient temperature in Fahrenheit [16], [48]. Then, runtime overall PUE $\gamma(t)$ can be calculated by including pPUE and the fraction of other non-IT power consumption (e.g., energy loss in power supply). The measured data points and fitted model are shown in Fig. 3a, while the pPUE calculated using (5) is shown in Fig. 3b for a snapshot of outside air temperature in San Francisco, CA.

Solar energy. On-site solar energy, a popular form of renewable energy, has been increasingly adopted by colocations (e.g., Equinix). Here, we consider that the colocation has photovoltaic (PV) panels to harvest solar energy on-site.

Solar energy generation is intermittent and depends on solar irradiance and weather conditions. Recent literature [24] shows that autoregressive moving average (ARMA) model based on historic data can predict solar generation with a reasonable accuracy.

We only require short-term solar energy prediction (as shown in Section IV-D). Thus, as an example, we use ARMA-based prediction method because of its lightweight implementation and good accuracy [24]. Specifically, our ARMA model is built with sum of weighted auto-regressive (AR) and moving-average (MA) terms. The predicted solar generation at time slot t using ARMA can be expressed as $s'(t) = \sum_{i=1}^{p} A_i \cdot s'(t-i) + \sum_{j=1}^{q} B_j \cdot \epsilon(t-j)$, where $s'(t)$ is the predicted solar energy, $\epsilon(t-j)$ is white noise with zero mean, p and q are the orders, and A_i and B_j are the weight parameters learned a priori. In Fig. 4a, we show predicted and actual solar generation of 7 days based on solar energy data from California ISO [4]. In the prediction, we have a Mean

[1] pPUE is defined as $\frac{Power_{IT}+Power_{Cooling}}{Power_{IT}}$, without including other non-IT power consumption such as losses in power supply system.



(a) (b)

Fig. 4: **(a)** Solar prediction with ARMA. Model parameters: $p = 2$, $q = 2$, $(A_1, A_2) = (1.5737, -0.6689)$ and $(B_1, B_2) = (0.3654, -0.1962)$. **(b)** Periodogram for different workloads using FFT.

Absolute Error (MAE) of 18kW, which is less than 2.5% of the considered peak generation of 750kW. More sophisticated models, e.g., incorporating weather forecast [39], can improve prediction and be plugged into RECO for areas where solar generation is not as regular as California.

B. Tracking peak power demand

The peak power demand is determined at the end of a billing cycle, and hence it cannot be perfectly known at runtime. To address this, we propose to keep track of the peak power demand value, denoted by $Q_j(t)$, which indicates the j-th type of peak power demand up to the beginning of time slot t. Intuitively, if the new power demand in the upcoming time slot is expected to exceed $Q_j(t)$, the colocation operator needs to offer a higher reward rate to better encourage tenants' energy saving for reducing demand charge.

The colocation operator updates $Q_j(t)$ online, if time t belongs to the time interval for type-j peak power demand, as follows

$$Q_j(t+1) = \max\left[\frac{[\gamma(t)\cdot E(r(t),t) - s(t)]^+}{\Delta t}, Q_j(t)\right], \quad (6)$$

where $\frac{[\gamma(t)*E(r(t),t)-s(t)]^+}{\Delta t}$ is the average power demand during time t. We initialize $Q_j(0)$ using an estimated peak power demand for the upcoming billing cycle (e.g., based on the peak demand of the previous cycle). The tracked peak power demand $Q_j(t)$ serves as a feedback value to determine whether it is necessary to offer a high reward rate to tame the peak power demand.

C. Learning tenants' response to reward

Naturally, optimizing the reward rate $r(t)$ requires the colocation operator to accurately predict how tenants would respond to the offered reward, but tenants' response information is absent in practice. To address this challenge, we propose a learning-based approach that predicts how tenants respond to the offered reward based on history data. We model the tenants' aggregate response (i.e., aggregate energy reduction) using a parameterized *response function* $\Delta E(r)$: if offered a reward rate of r \$/kWh, tenants will aggregately reduce servers' energy consumption by $\Delta E(r)$. We will explain the choice of $\Delta E(r)$ for tenants' response in Section VI-B.

Tenants' energy reduction naturally depend on their SLA constraints, and thus varies with workloads. However, IT workload exhibits diurnal patterns, which can be leveraged to greatly reduce the learning complexity. To validate this point, in Fig. 4b, we show the periodogram of time-series data of four different real-life workload traces (also used in our simulations) using Fast Fourier Transform (FFT). The peak at 24 hours indicates that workloads have a strong correlation over each 24 hours (i.e., daily repetition of workload). Thus, the colocation operator can just learn the *diurnal* response function: assume that the response functions for the same time slot of two different days are the same, and then update it incrementally at runtime. That is, if there are K time slots in a day, the colocation operator learns K different response functions, and we denote them as $\Delta E_k(r)$ where $k = \{0, 1, \cdots, K-1\}$.

We employ non-linear curve fitting based on least square errors to learn the response function. We use a sliding window with a predetermined number of previous observations (i.e., energy reduction and reward) to determine the unknown parameters in our parameterized response function. At the end of a time slot, the new observation replaces the oldest one, thus avoiding using too old information. RECO-LTR (RECO-Learning Tenants' Response) in Algorithm 1 presents our curve fitting algorithm to update the response function online. In our simulation, Fig. 13b demonstrates that the proposed learning-based method can reasonably accurately learn tenants' response over time.

Algorithm 1 RECO-LTR: Learning Tenants' Response

1: Input: Set of previous I observations $X' = \{(r'_i, y'_i)$: r'_i and y'_i are reward and energy reduction in observation $i\}$ for $i = 1, 2, \cdots I$ (larger index represents older data); new observation (r_0, y_0)
2: Set $X = \{(r_0, y_0), (r'_i, y'_i) : i = 1, 2, \cdots, I-1\}$
3: Update parameters for response function $\Delta E(r)$ to minimize $\sum_{i=0}^{I-1}(y_i - \Delta E(r_i))^2$

We next note that as in typical incentive-based approaches (e.g., utility incentive programs [43]), a reference usage for the no-reward case needs to be chosen in order to calculate each tenant's energy reduction. Thus, when the colocation operator announces reward rate r, it also notifies each participating tenant of its reference energy usage, such that tenants can determine on their own whether and how much energy to reduce. In our study, we can set reference usage based on tenants' energy consumption history (when no reward was offered) and/or calculate the diurnal reference energy usage based on the learnt response function evaluated at zero reward.

D. Feedback-based online optimization

We break down the original offline problem **P-1** into an online optimization problem (denoted as **P-2**). Specifically, we remove the total demand charge part and replace it with the cost increase associated with increase in peak power demand (hence demand charge). The new objective is to optimize

Algorithm 2 RECO

Initialize: For $t = 0, \forall j$ set $Q_j(0) = P_j^o$, where P_j^o is the estimated type-j peak power demand based on previous billing cycle or expectation.
while $t \leq T - 1$ **do**
  Input: Electricity price $u(t)$ and predicted solar generation $s'(t)$.
  Optimize: Solve **P-2** to find $r(t)$.
  Measurement: Measure energy reduction $\Delta E(r(t), t)$ (based on reference usage), and solar generation $s(t)$.
  Update peak power demand: For all $j \in A_j$, update $Q_j(t)$ according (6).
  Update tenants' response function: Using RECO-LTR (Algorithm 1), update $\Delta E_k(r)$ with $\{r(t), \Delta E(r(t), t)\}$, where $k = t$ mod K.
  $t = t + 1$
end while

reward rate $r(t)$ for minimizing

$$\textbf{P-2}: \quad C_{energy}(r(t), t) + C_{reward}(r(t), t) + \sum_j d_j \cdot \left[\frac{\gamma(t) \cdot E(r(t), t) - s'(t)}{\Delta t} - Q_j(t) \right]^+ \cdot \mathbb{I}_{t \in A_j}, \tag{7}$$

where $C_{energy}(r(t), t)$ and $C_{reward}(r(t), t)$ are the energy cost and reward cost given by (2) and (3), respectively, $\left[\frac{\gamma(t) \cdot E(r(t), t) - s'(t)}{\Delta t} - Q_j(t) \right]^+$ indicates whether the new (predicted) power demand during t will exceed the currently tracked value of $Q_j(t)$ for type-j demand charge, and $\mathbb{I}_{t \in A_j}$ is the indicator function equal to one if and only if time t falls into the time interval A_j for type-j demand charge (defined by the power utility).

We formally describe the feedback-based online optimization in Algorithm 2. At the beginning of each time slot, RECO takes the tracked peak power demand, electricity price $u(t)$, predicted tenants' response function and solar generation $s'(t)$ as inputs, and yields the reward rate $r(t)$ \$/kW by solving **P-2**. At the end of each time slot, RECO updates the peak demand queues Q_j using the actual power consumption. RECO also records the actual response of the tenants to the reward $\Delta E(r, t)$, and updates the corresponding response function using RECO-LTR with the new observation. The whole process is repeated until the end of a billing cycle.

We show the system diagram of implementing RECO in Fig. 5. On the colocation operator side, RECO can be implemented as a complementary/additional control module alongside any existing control systems (e.g., cooling control). Tenants, on the other hand, only need a very lightweight software to communicate with the operator for receiving the reward rate online. Upon receiving the reward information, tenants can decide at their own discretion whether and how to reduce energy subject to SLA for rewards.

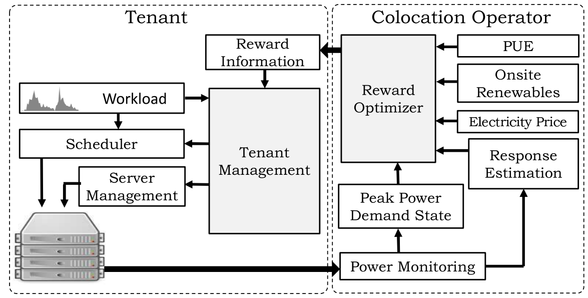


Fig. 5: System diagram of RECO.

V. Experiment

This section presents a prototype to demonstrate that RECO can effectively reduce colocation's cost by more than 10%. We show that the tenants can save their colocation rental cost without violating SLAs, while the colocation can save on both energy and demand charges. We first describe our colocation test bed, and then present the experiment results.

A. Colocation test bed

1) Hardware.: We build a scaled-down test bed with five Dell PowerEdge R720 rack servers. Each server has one Intel Xeon E5-2620 Processor with 6-cores, 32GB RAM and four 320 GB hard drives in RAID 0 configuration. One server (called "I/O Server") is equipped with a second processor and four additional hard disks, and used to host the database VMs. We use Xen Server 6 as the virtualization platform and Ubuntu Server 12.04.4 as the hosted operating system in each VM. As a rule of thumb, we allocate at least one physical core to each VM. We use a separate HP tower server to implement RECO and communicate with tenants using Java sockets. WattsUp Pro power meters are used to monitor power consumption of the tenants' Dell PowerEdge servers.

2) Tenants.: We have two tenants in our prototype, one running delay-tolerant Hadoop jobs and the other one processing key-value store (KVS) workload which resembles a realistic multi-tiered website such as social networking. The Hadoop system is built on 12 VMs hosted on 2 servers. We configure 11 worker nodes and 1 master node for the Hadoop system. A custom control module is used to consolidate and/or reconfigure the Hadoop servers to trade performance for energy. For Hadoop workload, we perform *sort* benchmark on randomly generated files of different sizes using *RandomTextWriter* (Hadoop's default).

Our implementation of KVS workloads has four tiers: front-end load balancer, application, memory cache, and database. The load balancer receives jobs from the generator and routes the requests to the application servers. The application tier processes the key and sends request to back-end database to get values. The back-end database is implemented in two tiers: replicated memory cache and database. We use three Memcached VMs and three database VMs, and put them in the I/O server. There are 15 application VMs in total (12 on two application servers and the other three on the I/O server). There are 100 million key-value entries in the database, and

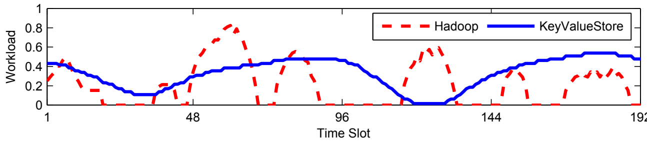
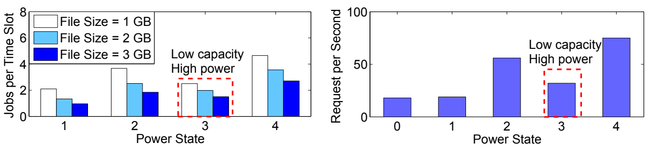


Fig. 6: Workload traces normalized to maximum capacity.



(a) Hadoop tenant (b) KVS tenant

Fig. 7: Processing capacity under different power states.

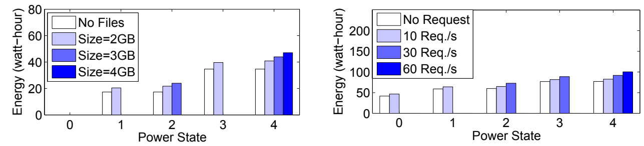
each key-value request returns multiple keys and the process repeats until the exit condition (e.g., number of iteration) is met. The KVS tenant can reconfigure the cluster and switch off up to two application servers (hosting 12 application VMs) to reduce energy.

3) Other settings.: We use the workload traces from Microsoft Research (MSR) as Hadoop workloads, and Gmail workload traces as KVS workloads [2], [41]. Fig. 6 shows the workload traces of the tenants normalized to their maximum processing capacity. Length of each time slot in our experiment is 15 minutes, and we run the experiment for 192 time slots (48 hours). We use the electricity price of PG&E [5]. Due to the relatively short experiment, we consider that RECO has already learned the response function before the experiment starts, but we will examine the learning capability of RECO via simulations. Due to prototype's limitation, we do not consider cooling efficiency or availability of solar energy, which will be incorporated in the simulation section.

B. Tenants' response

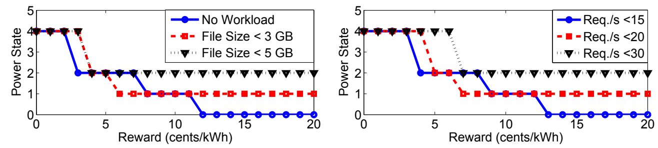
We consider that the Hadoop tenant has a SLA on job's maximum completion time of 15 minutes, while the KVS tenant has a SLA of 500 ms on the 95% delay (as similarly considered in prior research [19]). Each server is set to have three power states: high speed (H), low speed (L), and deep sleep/shut-down (Z). High and low speed settings correspond to all CPU cores running at 2 GHz and 1.2 GHz, respectively.

There are five combinations of power states for the Hadoop tenant with two servers, and we index the power states from 0 to 4: $(serverH1, serverH2) = \{(Z,Z),(L,Z),(H,Z),(L,L),(H,H)\}$. The KVS tenant with three servers also has five possible power states, because we keep the database server hosting the Memcached and database VMs unchanged. The server power state combinations are from the set $(serverK1, serverK2, serverK3) = \{(Z,Z,H),(Z,L,H),(Z,H,H),(L,L,H),(H,H,H)\}$. The first two servers are application servers and the last one is the I/O server. Note that, power state 0 corresponds to lowest speed and thus maximum energy reduction, while power state 4 means the system is running at its maximum capacity. Fig. 7 shows tenants' processing capacities subject to SLA



(a) Hadoop tenant (b) KVS tenant

Fig. 8: Energy consumption under different power states.



(a) Hadoop tenant (b) KVS tenant

Fig. 9: Response to reward under different workloads.

constraints under different power states. We see that power state 3 for both tenants has a lower processing capacity but consumes more power.

In Fig. 8, we show the energy consumption associated with each power state for different workload. If a certain workload cannot be processed given a power state, then its energy consumption at that power state is omitted from the figure. Fig. 8a shows the energy consumption of the Hadoop tenant during a time slot. We see that, the same file consumes more energy when processed in a higher power state, indicating a waste of energy when the system has a low workload. We also see that large files (e.g., 4GB) cannot be processed at low power states because of the SLA constraint. In Fig. 8b, we show the energy consumption by KVS tenant's servers for different request rates. Similar to that of Hadoop tenant, low request rates can be processed at a low power state with low energy consumption, while high request rates (e.g., 60 requests/second) require the use of higher power states and also more energy. The key observation in Fig. 8 is the energy saving opportunity for processing workloads subject to SLA.

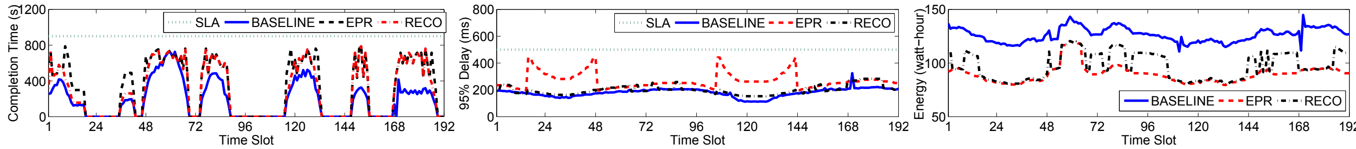
We consider the tenants' response to rewards in such a way that it resembles the response used in simulations (detailed in Section VI-B). Fig. 9 shows the tenants' response to different rewards under different workload conditions. Because of less capacity but more power/energy at power state 3, tenants do not use this state. We also see that because of SLAs, tenants cap their energy reduction given high incoming workloads and do not run their systems in very low power states (thus low capacity). The KVS tenant can use power state 0 for non-zero workloads, because it has three application VMs hosted on the I/O server that is always on.

C. Benchmarks

We consider two benchmarks to compare RECO with.

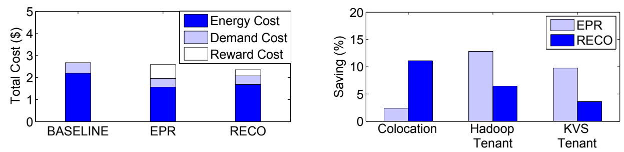
BASELINE. This is the baseline case where the colocation adopts a power-based pricing, without using any rewards. The tenants keep all their servers running.

EPR (Electricity Price-based Reward). In this case, the colocation directly offers electricity price as reward, without



(a) Job completion time for Hadoop tenant (b) 95% delay of KVS tenant (c) Energy consumption

Fig. 10: Comparison of different algorithms.



(a) Colocation's cost (b) Cost savings

Fig. 11: Cost and savings under different algorithms.

accounting for time-varying cooling efficiency or solar energy availability. This is equivalent to energy-based pricing.

D. Experiment result

We first compare the performance of the tenants' workloads in Fig. 10. We see that both tenants can reduce energy without SLA violation, showing the potential of RECO in real-life systems. In Fig. 10c, we show the energy consumption, demonstrating that RECO and EPR have a significantly lower energy consumption compared to BASELINE. In some time slots, EPR has lower energy consumption than RECO, because EPR provides a higher reward equal to electricity price.

Throughout the evaluation, we focus on the comparison of colocation operator's cost (including energy cost, peak power demand cost, and reward cost if applicable).[2] Fig. 11a shows the colocation's total cost for different algorithms. As we run the experiment for 48 hours, we scale down the monthly demand charge by PG&E to 48 hours based on a pro-rated charge. We see that RECO has the lowest total cost. BASELINE does not incur any reward cost, but has significantly higher energy and demand costs. EPR has the lowest energy and demand charges, but gives a significant portion of the cost saving as reward, thus resulting a total cost higher than RECO.

In Fig. 11b, we show the total cost savings of the colocation opeator and tenants by using RECO and EPR compared to BASELINE. RECO has more than 10% cost saving, and the Hadoop tenant and KVS tenant save 6.5% and 3.5% of their colocation rental cost[3], respectively. EPR only saves less than 3% of the total cost for the colocation operator, although both tenants save around 10% of their rental costs.

[2]We consider the commonly-used power-based pricing as the baseline case, and RECO is applied on top of this baseline. Hence, the colocation's revenue, i.e., tenants' power-based rent (excluding power-irrelevant bandwidth charges, etc.), is pre-determined and isolated from our study.

[3]The rental cost is calculated based on pro-rated for 48 hours with a rental rate of 147$/kW per month (a fair market rate for colocation service [11]), considering that Hadoop and KVS tenants have power subscriptions of 240W and 340W, respectively.

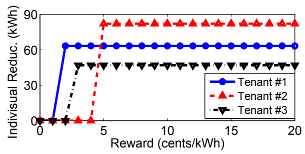
VI. Simulation

In this section, we present a trace-based simulation, complementing the prototype experiment. We show that using RECO, colocation operator can reduce the monthly cost by up to 27%, while the tenants can get as much as 16% of their monthly rent as reward. We first present our setup and then results.

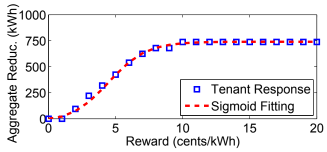
A. Setup

We consider a colocation located in San Francisco, California (a major market serving Silicon Valley) [13]. The colocation has 15 tenants, each having 2,000 servers and a peak power subscription of 500 kW. We collect the traces from Google, Microsoft, Wikipedia, Verizon Teremark and University (FIU) as the tenants' workload traces. In particular, we take the U.S. traffic data for Google services: Gmail, Search, Maps and Youtube from [2]. Microsoft traces are collected from [41], which consist of traces from Hotmail, Messenger and MSR. The Wikipedia traces are from [42], and contain traffic for Wikipedia (English). We collect the Verizon Teremark and University traces through our direct collaboration with them. Verizon Teremark traces are collected from multiple flywheel UPS measurements at one of their colocations, whereas the university trace contains HTTP requests to its website. The workloads are scaled to have a 15% average server utilization for each tenant, which is consistent with public disclosure [21], [34]. Note that tenants need to turn on more servers than the minimum-required capacity, such that workloads do not overload servers and can satisfy SLA. We predict the solar energy generation based on the traces collected from [4] using ARMA, and scale it to have a peak solar generation of 750 kW (10% of critical peak power of the colocation). The colocation is connected with PG&E, and registers as a large commercial customer under PG&E's electric schedule E-20 [5]. Besides monthly service charges, the colocation is subject to peak power demand charge and energy charge [5]. We collect the temperature data for San Francisco, CA, from [6] for 2013, and use it to determine the colocation's cooling efficiency using (5).

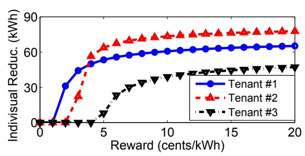
We use a discrete-time simulator, which is a common evaluation method for research. It simulates the colocation operator's decision and tenants' responses at runtime. The simulator for colocation operator takes renewable energy and temperature traces as inputs, executes RECO, and communicates with the tenant simulator using function calls. The tenant simulator uses workload traces and reward information as inputs, and outputs the servers' energy reduction. In each time



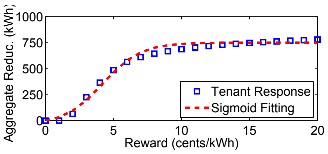
(a) Threshold-based response



(b) Fitting



(c) Profit-based response



(d) Fitting

Fig. 12: Tenant response and fitting.

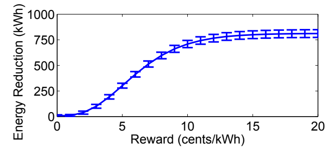
slot, all the decisions (e.g., reward, tenants' energy reduction) are logged.

B. Tenants' response

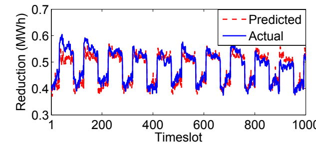
Upon receiving the reward information, as shown in Fig. 5, tenants can voluntarily choose their power management, depending on workloads and SLAs. Here, we consider that tenants will dynamically switch off servers (a variant of AutoScale being used in Facebook's production system [47]) while ensuring that their active servers' utilization will not exceed 50% for satisfying SLA.

We use Sigmoid function $f(r) = \frac{a}{1+c\cdot e^{-br}}$ for tenants' response, which exhibits two interesting properties: (1) given a low reward, tenants are reluctant to commit energy reduction; and (2) when the energy reduction approaches their maximum possible amount, tenants become less willing to reduce resource provisioning and energy.

To justify our choice of Sigmoid function, we consider two different cases of tenants' response to rewards, and show that the aggregate energy reductions can be approximated using Sigmoid functions. In the first case, we consider tenants' threshold-based binary response, where a tenant turns off the maximum number of servers subject to SLA when the reward rate is more than a cost threshold. Fig. 12a shows a sample of responses by three tenants (out of 15) who have different cost thresholds and SLA constraints. In the second case, we consider a profit-based response: turning off a server incurs a switching cost and also performance cost (due to possible performance degradation), and with the reward information, a tenant determines the optimal number of servers to turn off to maximize its net profit following a similar approach in [28]. We show a sample of profit-maximizing responses of three tenants in Fig. 12c. In both cases, tenants try to maximize their own net profits, consistent with prior studies that focus on energy cost saving [30]. From Fig. 12b and 12d, we see that in both cases, Sigmoid function can be used to estimate the aggregate response (i.e., total energy reduction by all tenants) with a high accuracy. Note that while we use Sigmoid function for evaluation, our methodology also applies to alternative response functions.

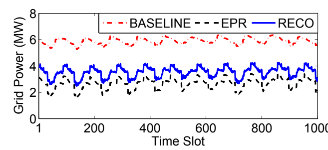


(a)

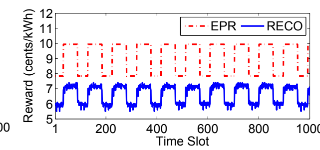


(b)

Fig. 13: **(a)** Response function for a day's first time slot. **(b)** Predicted and actual power reduction.



(a) Grid power



(b) Reward rates

Fig. 14: Grid power and reward rate w/ different algorithms.

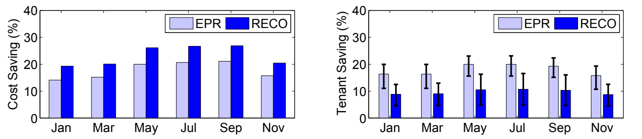
The colocation operator constructs a set of diurnal response functions, each corresponding to a different time slot of a day. Fig. 13a shows tenants' response to different reward rates for the first time slot of a day. The error bars represent the deviation of actual response from the learnt/predicted value. We consider that before simulation begins, the colocation operator already has the response functions based on one month's learning, which is sufficient as we will show later in Fig. 16 that the colocation can adapt to large changes in tenants' responses within one month.

C. Results

Below, we present our results based on the above settings. We examine the execution of RECO and show the performance comparison in terms of cost savings. Then, we demonstrate the applicability of RECO in different scenarios. The simulations are done for one year and each time slot is 15-minute, matching PG&E's peak power demand accounting [5].

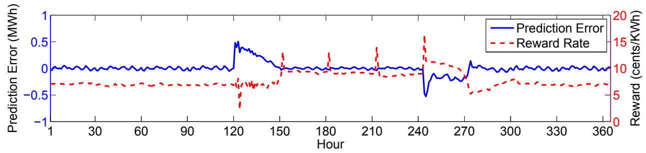
1) Performance comparison: The colocation operator minimizes the cost by optimally choosing the reward rate based on the response function and using Algorithm 2. Because of prediction error (as shown by error bar in Fig. 13a), the actual energy reduction may be different from the predicted value. However, Fig. 13b shows the actual and predicted energy reduction for a snapshot period, matching each other fairly well. The average deviation between the actual and predicted energy reduction for the whole year is less than 1%.

Fig. 14 shows a snapshot of colocation's grid power consumption and reward rates. In Fig. 14a, we see that BASELINE has the highest grid power consumption because tenants are charged based on power subscription and have no incentives to reduce any energy. RECO and EPR have much lower grid power consumption compared to BASELINE, saving 41% and 54% of average power consumption, respectively. Fig. 14b shows the reward rates provided to tenants. We see that RECO offers lower reward rates (average 7 ¢/kWh) compared to EPR (average 9.7 ¢/kWh), because RECO is optimizing the reward rate to minimize the colocation's cost and giving a higher

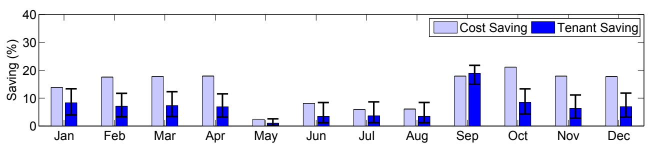


(a) Colocation's monthly cost saving (b) Tenants' average monthly saving

Fig. 15: Monthly cost savings for colocation and tenants.



(a) Prediction error in energy reduction and reward rate



(b) Colocation's cost and tenants' monthly saving

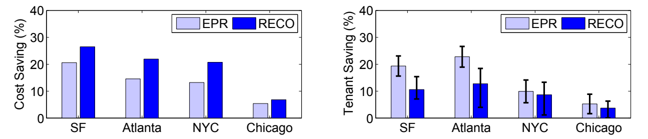
Fig. 16: Impact of changes in tenants' behaviors.

reward will increase energy reduction but the corresponding reward cost will increase the overall cost. Because of the lower reward rate offered by RECO, power consumption of RECO is higher than EPR, but the overall cost is reduced, which is the metric that RECO focuses on.

We show the cost savings of EPR and RECO in Fig. 15, compared to BASELINE that offers no reward. The error bars in Fig. 15a represent the range of tenants' cost savings. We see that RECO has a more than 19% cost saving compared to BASELINE, while reaching up to 27% during the summer months. The increased cost savings during summer months are because PG&E has higher energy and demand charges during summer, thus increasing the potential of cost saving via rewards. EPR has a cost saving of around 15% during winter and 20% during summer. While RECO saves more than EPR in terms of colocation's costs, it gives less reward to tenants and keeps some energy cost saving for the colocation operator. Nonetheless, tenants can still get back an average of more than 15% of their colocation costs.[4]

2) Adaptation of RECO*:* To demonstrate that RECO can adapt to large changes in tenants' power management, we increase the value of b in Sigmoid function for all the tenants' responses at the start of May, making the tenants less willing to reduce energy. We change back b to its initial value at the start of September. Fig. 16a shows the impact caused by the sudden changes in tenants' behaviors on the response function. We see the sudden spikes in energy reduction prediction errors when the changes occur, and then the error gradually goes down, showing the adaptability of RECO. The similar pattern occurs again when the response setting is changed back to its initial value. Positive prediction error indicates over-prediction of energy reduction, while negative prediction error indicates

[4]Based on a fair market rate of 147$/kW per month for colocation [11].



(a) Colocation's monthly cost saving (b) Tenants' average monthly saving

Fig. 17: Cost savings in different locations.

under-prediction. We also see that higher reward rate is offered when the tenants become less willing to participate in RECO. However, as shown in Fig. 16b, the tenants also have lower savings when they are less willing to reduce energy and correspondingly, cost saving for the colocation also decreases. We take the liberty that tenants will become more willing to shed energy (for rewards), as power management is being increasingly adopted and tenants (e.g., Apple and Akamai) are pressured by the public for energy efficiency [7], [9].

Finally, we show in Fig. 17 the cost savings by RECO and EPR, compared to BASELINE, in different U.S. colocation markets. The error bar indicates the range of different tenants' savings. The results are consistent with our above findings: by using RECO, colocation operator achieves the lowest cost, while tenants are also able to save some costs. The variations in cost savings across locations are mainly because of the location-specific electricity rates and colocation rental rates.

VII. Related Work

Data center power management has been explored by many prior studies. For example, power proportionality [22], [27], [28], has been well investigated and also applied in large systems (e.g., Facebook's AutoScale [47]). In geo-distributed data centers, several studies explore spatio-temporal diversity, e.g., electricity price [30], [36], [37], carbon efficiency [20], [26], and renewable availability [50]. Power over-subscription [46] and hardware heterogeneity [23] are also effective to reduce operator's total cost of ownership and improve performance. Recently, taming data center's peak power demand charge has been studied by using, e.g., workload scheduling [45], load shedding [49], and jointly optimizing IT and facility resources [31]. These studies are applicable for owner-operated data centers, but they cannot be used directly for colocations due to the operator's lack of control over tenants' servers.

In the context of colocations, [38] investigates colocation demand response to aid power grid stability, while [25] proposes a bidding-based mechanism to let tenants "compete" for a limited budget for minimizing carbon footprint (rather than cost). These studies, however, suffer from tenants' untruthfulness, i.e., if tenants report falsified values to gain benefits, the solutions will collapse. By contrast, in our study, the colocation operator proactively learns tenants' response without relying on tenants' self-reporting, thus avoiding tenants' cheating. Further, our study focuses on cost minimization and also differs from [25], [38] in: (1) we capture peak power demand charge; (2) we incorporate the effect of outside temperature and solar energy availability; and (3) we propose a new feedback-based online algorithm to optimize reward rate for cost minimization.

VIII. Conclusion

This paper focused on reducing operational cost for colocation and addressed the lack of coordination among tenants' power consumption, which has been neglected by prior research. We proposed RECO, which learns tenants' response to rewards and dynamically sets the reward rate to reduce colocation's cost using feedback-based online optimization. We evaluated RECO via a scaled-down prototype and also simulations, showing that RECO can save up to 27% of the operational cost while the tenants may save up to 15% of their colocation rent subject to SLA.

Acknowledgement

This work was supported in part by NSF CNS-1423137 and CNS-1143607.

References

[1] "Wikimedia's data center search ends with cyrusone," http://www.datacenterknowledge.com/archives/2014/05/05/wikimedias-data-center-search-ends-cyrusone/.
[2] "Google transparency report," http://www.google.com/transparencyreport/traffic/explorer.
[3] "Colocation market - worldwide market forecast and analysis (2013 - 2018)," http://www.marketsandmarkets.com.
[4] "California ISO open access same-time information system," http://www.oasis.caiso.com.
[5] http://www.pge.com/.
[6] "Historical Weather," http://www.wunderground.com/.
[7] Akamai, "Environmental sustainability policy," http://www.akamai.com.
[8] Amazon, https://aws.amazon.com/about-aws/globalinfrastructure/.
[9] Apple, "Environmental responsibility report," 2014, http://images.apple.com/environment/reports/docs/Apple_Environmental_Responsibility_Report_2014.pdf.
[10] L. A. Barroso, J. Clidaras, and U. Hoelzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan & Claypool, 2013.
[11] CBRE, "Q4 2013: National data center market update," 2013, http://www.cbre.us/services/office/data-center-solutions-group/.
[12] CyrusOne, "Colocation: the logical home for the cloud," 2012,.
[13] DatacenterMap, "Colocation USA," http://www.datacentermap.com/usa/.
[14] J. dePreaux, "Wholesale and retail data centers - North America and Europe - 2013," *IHS*, Jul. 2013, https://technology.ihs.com/api/binary/492570.
[15] R. A. Dines, "Build or buy? the economics of data center facilities," in *Forrester Research*, 2011, http://www.io.com/wp-content/uploads/2013/04/build-or-buy.pdf.
[16] Emerson Network Power, "Liebert DSE precision cooling system," http://www.emersonnetworkpower.com/documentation/en-us/products/precisioncooling/largeroomcooling/documents/sl-18927.pdf.
[17] Enaxis Consulting, "Pricing data center co-location services," 2009, http://enaxisconsulting.com.
[18] Equinix, www.equinix.com.
[19] 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.
[20] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 211–222, Aug. 2012.
[21] J. Glanz, "Power, pollution and the internet," in *The New York Times*, Sep. 22, 2012.
[22] B. Guenter, N. Jain, and C. Williams, "Managing cost, performance and reliability tradeoffs for energy-aware server provisioning," in *IEEE Infocom*, 2011.
[23] M. Guevara, B. Lubin, and B. C. Lee, "Navigating heterogeneous processors with market mechanisms," in *HPCA*, 2013.
[24] R. Huang, T. Huang, R. Gadh, and N. Li, "Solar generation prediction using the arma model in a laboratory-level micro-grid," in *SmartGridComm*, 2012, pp. 528–533.
[25] M. A. Islam, S. Ren, and X. Wang, "GreenColo: A novel incentive mechanism for minimizing carbon footprint in colocation data center," in *IGCC*, 2014.
[26] 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," ser. SuperComputing, 2011.
[27] C. Li, R. Zhou, and T. Li, "Enabling distributed generation powered sustainable high-performance data center," in *HPCA*, 2013.
[28] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," in *IEEE Infocom*, 2011.
[29] H. Liu, "A measurement study of server utilization in public clouds," in *DASC*, 2011.
[30] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Greening geographical load balancing," in *SIGMETRICS*, 2011.
[31] Z. Liu, A. Wierman, Y. Chen, B. Razon, and N. Chen, "Data center demand response: avoiding the coincident peak via workload shifting and local generation," in *SIGMETRICS*, 2013.
[32] 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.
[33] J. Novet, "Colocation providers, customers trade tips on energy savings," Nov. 2013, http://www.datacenterknowledge.com/.
[34] NRDC, "Scaling up energy efficiency across the data center industry: Evaluating key drivers and barriers," *Issue Paper*, Aug. 2014.
[35] 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.
[36] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *SIGCOMM*, 2009.
[37] 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 *IEEE Infocom*, 2010.
[38] S. Ren and M. A. Islam, "Colocation demand response: Why do I turn off my servers?" in *ICAC*, 2014.
[39] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting solar generation from weather forecasts using machine learning," in *SmartGridComm*, 2011, pp. 528–533.
[40] STEM, "Demand charges," 2014.
[41] E. Thereska, A. Donnelly, and D. Narayanan, "Sierra: a power-proportional, distributed storage system," *Tech. Rep. MSR-TR-2009-153*, 2009.
[42] G. Urdaneta, G. Pierre, and M. Van Steen, "Wikipedia workload analysis for decentralized hosting," *Computer Networks*, 2009.
[43] U.S. EPA, "Utility guide for designing incentive programs focused on data center efficiency measures," Nov. 2012.
[44] Verizon Terremark, "Pricing plan," http://www.terremark.com/landing-pages/colocation-promo-group3/.
[45] C. Wang, B. Urgaonkar, Q. Wang, and G. Kesidis, "A hierarchical demand response framework for data center power cost optimization under real-world electricity pricing," in *MASCOTS*, 2014.
[46] D. Wang, C. Ren, and A. Sivasubramaniam, "Virtualizing power distribution in datacenters," in *ISCA*, 2013.
[47] Q. Wu, "Making facebook's software infrastructure more energy efficient with autoscale," 2014.
[48] H. Xu, C. Feng, and B. Li, "Temperature aware workload management in geo-distributed datacenters," in *SIGMETRICS*, 2013.
[49] H. Xu and B. Li, "Reducing electricity demand charge for data centers with partial execution," in *e-Energy*, 2014.
[50] Y. Zhang, Y. Wang, and X. Wang, "Greenware: Greening cloud-scale data centers to maximize the use of renewable energy," in *Middleware*, 2011.
[51] 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.