A Near-Optimal Control Policy in Cloud Systems with Renewable Sources and Time-dependent Energy Price

Jiashang Liu*, Joohyun Lee†, Ness B. Shroff*‡, Prasun Sinha‡ and Sinong Wang*
* Department of Electrical and Computer Engineering, The Ohio State University
† Department of Electronics and Communications Engineering, Hanyang University
‡ Department of Computer Science and Engineering, The Ohio State University

***Abstract*—The cost of energy usage is of significant concern in cloud/data center systems that support a large number of servers. A simple way to reduce energy consumption and the electricity bill is to turn some of the servers off during periods of under utilization. However, turning a server back on typically consumes a lot of energy. Another way to reduce the energy cost is to equip cloud systems with renewable resources and batteries. Most works in the literature have focused on one or the other approach. In this work, we propose a joint server on-off and energy control policy, which determines the servers' on-off status, as well as the energy purchasing behavior, by taking electricity price, renewable resources, possible future tasks and turn-on cost into account. The server on-off control component is proved to be optimal in terms of energy consumption minimization. The joint policy is shown to be arbitrarily close to the optimal solution in terms of electricity bill minimization, in the case where the battery has infinite capacity with stable energy level. Simulation results show that even under reasonable battery size, a significant electricity cost reduction is achieved with the proposed policy.**

I. Introduction

Recent years have witnessed an explosive growth in cloud systems, due to its ability to free devices from heavy computation and assist in synchronization of multiple end devices. However, as these cloud systems have grown their energy consumption and costs have also grown. In fact, the energy bill for a cloud system provider could easily be in millions of dollars per year [1]. Hence, the question of how to optimally manage its resources to minimize consumption is becoming increasingly significant and already receiving a large amount of attention.

A promising approach for reducing energy consumption is to equip cloud systems with renewable resources and batteries, leading to the concept of the "green cloud system" [2]–[4]. However, because of the unpredictable nature of renewable energy sources, to ensure stable operation, cloud systems typically choose to buy electricity from power grids. These power grids have time-varying energy prices because they use a combination of energy sources and supply/demand control. We also notice that cloud systems are usually deployed to satisfy the peak demand of input tasks, which means that during off-peak periods, turning off idle servers can potentially reduce the energy consumption [5]. However, switching a server from an off-state to an on-state generally incurs a heavy turn-on cost [6], which introduces additional energy consumption.

Many prior studies have focused on reducing the energy consumption and hence the electricity bill. For instance, to dynamically determine how many servers are to be kept on, by taking turn-on cost of servers into consideration, the authors in [6] proposed an online algorithm that can achieve a constant competitive ratio. [7] continued to model jobs to be delay-tolerant with different deadlines, and converted the offline optimization problem to a generalized assignment problem. On the other hand, to reduce electricity bill, [1], [8]–[10] proposed energy management schemes which utilize batteries as buffers to store energy, to both minimize the peak energy consumption and the electricity cost. However, none of these works has considered both on-off costs of the servers, and the renewable sources together. The key question here is: *how should the control algorithm jointly decide on the on-off sequences of the number of servers and usage of batteries, when both operation-cost and turn-on cost of servers together with the renewable resources and electricity price are considered, in order to minimize the electricity bill of cloud systems.*

In this work, we consider a cloud system equipped with batteries, which can be recharged with renewable resources and a power grid, with time-varying electricity price. We focus on the problem of determining the status of servers, as well as the energy purchasing behavior, by considering the operation-cost of servers, the turn-on cost of servers, the electricity price and the state of batteries. The objective is to reduce the electricity bill of the cloud system, under the constraint that the task demands are served in each time-slot.

The proposed policy is a two-step scheme, which first determines the number of servers to be on, and then determines the amount of energy to be purchased from a power grid. To determine the server on-off state, we rely on the Markov decision process framework that considers the effect of the future tasks, which balances the future turn-on energy cost and the current energy cost. We then use the stochastic Lyapunov optimization technique to determine the amount of energy to be purchased. We show that our proposed policy achieves asymptotically optimal performance, and performs well under reasonable battery sizes through numerical simulations.

Our paper is organized as follows. In Section II we discuss the task, battery and energy model. In Section III we formulate our optimization problem. The proposed policy and its theoretical performance analysis are discussed in Section IV. Detailed simulation results are shown in Section V. We conclude our paper in Section VI.

II. System Model

We consider a cloud system with $N_{\max}$ servers equipped with batteries, which can be recharged by renewable sources

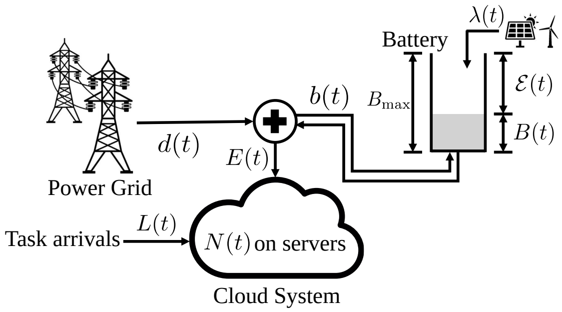


Fig. 1: System Model

or a power grid. Each server has the same computing power. The system is assumed to be time-slotted, where, during one time-slot, the on-off status of servers is fixed. Let $N(t)$ denote the number of servers that are on in time-slot t, with $N(t) \geq 1$, which is one of our control variables. We depict our system model in Fig. 1.

A. Task Model

Let $L(t)$ denote the number of tasks arriving at the beginning of time-slot t. Each task takes one time-slot to execute whether it runs on one or multiple servers in parallel. The tasks are delay-sensitive, and need to be served in that time-slot. We assume that $L(t)$ is a known integer at the beginning of time-slot t whose lower and upper bounds are L_{min} and L_{max}, respectively. We also assume that the number of tasks in the next time-slot, only depends on the number of tasks in the current time-slot, with a known conditional probability distribution. In other words, $\Pr(L(t+1)|L(t))$ is known apriori. This is a reasonable assumption, since with sufficient history, the number of future tasks can be accurately predicted.

B. Battery Model

The cloud system is equipped with a rechargeable battery. We use B_{max} to denote the maximum battery capacity and $B(t)$ to denote the battery level at the beginning of time-slot t. We define $\lambda(t)$ to be the amount of harvested energy in time-slot t. In time-slot t, $b(t)$ represents the amount of energy drawn from or charged to the battery. $b(t)$ can be positive or negative, with a positive value representing that the energy is drawn from the battery, and a negative value representing that the battery is charged. Based on the definitions above, we have the following equation for the battery level evolution:

$$B(t+1) = \min\{B_{\max}, B(t) + \lambda(t) - b(t)\}.$$

Since the energy drawn from the battery cannot exceed the current battery level, we must have:

$$b(t) \leq B(t).$$

Also, because of the finite charging rate of the battery, we assume that in any time-slot, the amount of charged energy from the power grid to the battery cannot be larger than a constant value, which we denote as $b_{\max}$, i.e., $b(t) \geq -b_{\max}$.

C. Energy and Price Model

In this paper, we consider three types of energy costs: *on cost, turn-on cost*, and *service cost*. For any server, if it is in the on-state, we assume that it incurs C_{on} unit(s) of energy consumption, which is also referred to as *on cost*. We also consider the *turn-on cost* $C_{\text{t/o}}$ of the server, which is the energy consumption to switch a server from the off-state to the on-state. We do not consider the turn-off cost as it is negligible in cloud servers [11]. Lastly, we consider the computational energy required to serve tasks in one time-slot, which we refer to as the *service cost*. We assume that a server consumes $f(L)$ unit(s) of energy to serve L tasks in one time-slot, where $f(L)$ is convex. For example, $f(L) \propto L^3$ for computing systems according to [11].

In time-slot t, if $N(t)$ servers are turned on, they consume $C_{\text{on}}N(t)$ unit(s) *on cost*. If at the beginning of time-slot t, $N(t-1)$ servers are in the on-state, the number of servers that need to be turned on is $\max\{0, N(t)-N(t-1)\}$. Also, assume that fractional tasks can be assigned to a server. Because of the convexity of function f, it is easy to verify that each on-server should be assigned with the same number of tasks in order to minimize the overall energy consumption. Let $E(t)$ denote the energy consumption of the cloud system in time-slot t. $E(t)$ can be expressed as follows:

$$\begin{aligned} E(t) = C_{\text{on}}N(t) + C_{\text{t/o}}\left[N(t) - N(t-1)\right]^+ \\ + N(t) \cdot f(L(t)/N(t)), \end{aligned} \tag{1}$$

where $x^+ \triangleq \max\{x, 0\}$.

Let $d(t)$ denote the amount of energy purchased from the power grid in time-slot t. Recall that $E(t)$ is the amount of energy that the system requires, and $b(t)$ is the amount of energy drawn from or charged to the battery. Then, we must have $E(t) = b(t) + d(t)$. In our model, selling energy to the power grid is not allowed, i.e., $d(t) \geq 0$, so we have:

$$b(t) \leq E(t).$$

Given $E(t)$, if one of $b(t)$ and $d(t)$ is determined, the other one will be fixed. Thus, the control algorithm needs to determine either one of $b(t)$ and $d(t)$ in each time-slot. In the rest of this paper, we refer to $b(t)$ as the energy control decision.

Let $P(t)$ denote the electricity price per unit energy during time-slot t, which is assumed to be known at the beginning of each time-slot. $P(t)$ is assumed to be fixed in time-slot t, but can be different across different time-slots due to the pricing strategy of energy providers. The cost of purchasing energy from the power grid in time-slot t is

$$P(t)d(t) = P(t)\left[E(t) - b(t)\right]. \tag{2}$$

In this paper, we assume that all the processes in the system are ergodic and have bounded values. We let $P_{\max}$, $\lambda_{\max}$ denote the maximum value of energy price and of the renewable energy in one time-slot. Also we denote $E_{\max}$ as the possible maximum energy consumption of the cloud system, i.e.,

$$E_{\max} \triangleq \max_{\substack{1 \leq N(t-1) \leq N_{\max}; \\ 1 \leq N(t) \leq N_{\max}; \\ L_{\min} \leq L(t) \leq L_{\max}}} E(t). \tag{3}$$

III. Problem Formulation

From Eqs. (1) and (2), we know that in each time-slot, the cost of purchasing energy from the power grid is determined by (1) the number of servers to be on in that time-slot (i.e., instantaneous energy consumption from the cloud system), and (2) the amount of energy drawn from or charged to the battery. Our goal is to design an online control algorithm for these two decisions in each time-slot, which achieves the minimum long-term average cost of purchasing energy from the power grid. We formally state the problem below as **P1**:

(Problem **P1**)

$$\min_{\mathbf{N},\mathbf{b}} \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[P(t)\,(E(t)-b(t))]$$

$$\text{s.t. } N(0)=1 \tag{4}$$
$$B(0)=B_{\max} \tag{5}$$
$$b(t)\geq -b_{\max},\ \forall t \tag{6}$$
$$b(t)\leq E(t),\ \forall t \tag{7}$$
$$b(t)\leq B(t),\ \forall t \tag{8}$$
$$B(t+1)=\min\{B_{\max}, B(t)+\lambda(t)-b(t)\},\ \forall t \tag{9}$$

Here, the expectation for time-slot t takes over all the randomness of the energy price, tasks arrivals, renewable arrivals, and control action from time-slot 1 to time-slot T.

Eq. (4) indicates that initially one server is in an on-state. Eq. (5) indicates the initial state of the battery level. Eq. (6) limits the maximum charging rate of the battery from the power grid. Eqs. (7) and (8) imply that the amount of energy drawn from the battery cannot exceed the amount of required energy, and the current battery level, reflecting the energy demand constraint and the battery physical constraint. Eq (9) is the battery level evolution.

To aid our theoretical analysis, we define *emptiness of battery*, denoted as $\mathcal{E}(t)$, as:

$$\mathcal{E}(t)\triangleq B_{\max}-B(t). \tag{10}$$

And Eqs. (5) and (9) are equivalent to the following equations:

$$\mathcal{E}(0)=0 \tag{11}$$
$$\mathcal{E}(t)\leq B_{\max} \tag{12}$$
$$\mathcal{E}(t+1)=[\mathcal{E}(t)-\lambda(t)+b(t)]^{+} \tag{13}$$

In **P1**, $\mathcal{E}(t)$ has an upper bound $B_{\max}$ in each time-slot, and this makes it very difficult to find a feasible solution to **P1**. To make the problem tractable, we consider a relaxed constraint of Eq. (12). We show that Eq. (12) together with Eq. (13) implies the following inequality:

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} b(t)\leq\bar{\lambda} \tag{14}$$

where $\bar{\lambda}$ is the long-term average renewable energy arrival rate.

We show this argument using contradiction. Suppose Eq. (14) does not hold, then there exists $\delta>0$ such that $\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} b(t)\geq\bar{\lambda}+\delta$, and $\mathcal{E}(t)$ is divergent and it goes to infinity, which is contradicted by Eq. (12).

By replacing Eq. (12) with Eq. (14), we have a new optimization problem stated formally as follows:

(Problem **P2**)

$$\min_{\mathbf{N},\mathbf{b}} \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[P(t)\,(E(t)-b(t))]$$

s.t. In each time-slot t, Eqs. (4)(6)(7)(11)(13)(14) hold

P2 is a relaxed version of **P1**, because Eq. (14) is a relaxed constraint of Eq. (12), while all other constraints remain the same. In **P2**, $\mathcal{E}(t)$ is kept to be rate-stable [12] by any feasible solution, and $\mathcal{E}(t)$ is not necessarily bounded. In other words, **P2** is with infinite maximum battery capacity, while the emptiness of the battery is enforced for guaranteeing stability.

An online policy is called feasible in **P2** if the decisions under this policy satisfy the constraints in each time-slot. We denote the set of all feasible policies as Π. In Section IV, we focus on online policies in Π. We then propose a joint server on-off and energy control scheme, which is shown to be arbitrarily close to the optimal solution. Interestingly, by carefully choosing our control parameter, the proposed scheme can be adapted to become a feasible solution of **P1**, which is with a finite battery size constraint.

IV. Joint server on-off and energy control

In this section, we propose our control policy for the (1) server on-off decision, and (2) energy control decision by combining dynamic programming and Lyapunov technique. To solve **P2**, we notice that the available distribution information of future traffic enables us to rely on an infinite-horizon Markov decision process (MDP) to help develop a good online policy by considering both current and future possible system state. To better explain our proposed control policy, we first describe an overall state transition model and the infinite-horizon MDP for the cloud system part.

A. State transition model

Let $S=S(t)$ denote the system state at the beginning of time-slot t, which includes the number of on servers in the previous time-slot $N(t-1)$, the number of unit tasks in the current time-slot $L(t)$, the electricity price $P(t)$ and the emptiness of battery $\mathcal{E}(t)$, i.e., $S(t)=(N(t-1),L(t),P(t),\mathcal{E}(t))$. We omit t and use $S,N,L,P,\mathcal{E}$ when there is no confusion. We use $\bar{N}=N(t-1)$ to denote the number of on servers in the previous time-slot. Let $\mathcal{S}$ denote the set of all possible states. We focus on designing online policies, hence in time-slot t, the exact future information, i.e., $L(\tau),P(\tau),\lambda(\tau)$ for $\tau>t$, is unknown. We define a joint server on-off control and energy control policy (μ,β), with μ representing the server on-off control component, and β representing the energy control component, such that it maps the current and all previous system states, to both the server on-off decision and the energy control decision in the current time-slot, i.e., $(\mu,\beta):S(\tau)_{0\leq\tau\leq t-1}\times\mathcal{S}\to[1,2,\cdots,N_{\max}]\times[-b_{\max},E_{\max}]$.

For the rest of the paper, (μ^{M},β^{L}) is referred to as our proposed control policy, which includes the server on-off control component μ^M, and the energy control component β^L. Our proposed policy (μ^M, β^L) is a two-step policy. (1) The server on-off control component μ^M uses part of the system state information in the current time-slot, to make the server on-off decision, by modeling the sub-problem as an infinite-horizon Markov decision process. (2) After that, the energy control component β uses other system state information in the current time-slot, together with the server on-off decision, to decide the energy control. Before we describe our policy (μ^M, β^L) precisely, we introduce the MDP model in the following subsection.

B. Markov decision process for the cloud servers

We define a sub system state at the beginning of time-slot t, denoted as $S_1(t)$, which is a part of the system state $S(t)$, to be the number of task arrivals in time-slot t and the number of on servers in time-slot $t-1$, i.e., $S_1(t) = (L(t), N(t-1))$, and let $\mathcal{S}_1$ be the set of all sub system states S_1. Since the conditional probability of the number of unit tasks in the next time-slot is available, i.e., $\Pr(L(t+1)|L(t))$ is known apriori, given a sub system state $S_1 = (L, \bar{N})$ and a server on-off decision N, the transition probability to the next state $S_1^{'} = (L^{'}, N^{'})$ is

$$\Pr(S_1^{'}|S_1, N) = \begin{cases} \Pr(L^{'}|L) & \text{if } N^{'} = N \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

We then define the potential function $h(S_1)$ of a given sub system state $S_1 = (L, \bar{N})$ as follows:

$$\hat{h}(S_1, N) \triangleq \hat{E}(S_1, N) + \sum_{S_1^{'} \in \mathcal{S}_1} \Pr(S_1^{'}|S_1, N) h(S_1^{'}) \quad (16)$$

$$h(S_1) \triangleq \min_N \hat{h}(S_1, N) \quad (17)$$

where $\hat{h}(S_1, N)$ is a potential value for the state S_1 with the decision N, $\hat{E}(S_1, N)$ is the instantaneous energy consumption for the state S_1 and the decision N, which is the value of Eq. (1) with $L(t) = L, N(t-1) = \bar{N}$ and $N(t) = N$. We will explain how to compute the potential function later.

Given a sub system state S_1 and a decision N, Eq. (16) not only focuses on the instantaneous energy consumption, but also takes the future potential energy consumption into account, by computing the expectation of the potential value of the next possible state. Eq. (17) indicates that for a given sub system state, its potential function value is the minimum value attained by an optimal decision.

Intuitively, the potential value is the relative difference of the total expected energy consumption between S_1 and a reference state S_0 with $h(S_0) = 0$. To minimize the average expected energy consumption, the server on-off decision should minimize the potential value of the state-decision pair (S_1, N), which is exactly what the servers on-off control of our proposed policy μ^M does.

C. Near-Optimal Two-step Control Algorithm

On one hand, in order to minimize the cost of purchasing energy, our server on-off control component μ^M focuses on minimizing the energy consumption of the system, by utilizing the task statistics information to help determine the number of servers to be on in each time-slot. On the other hand, the energy control component β^L utilizes battery as a buffer and exploits the fluctuation of electricity price, by purchasing energy from the power grid to charge the battery when the electricity price $P(t)$ is relatively low, while using the energy stored in battery when $P(t)$ is relatively high. Our proposed policy (μ^M, β^L) is described in Algorithm 1, with a free control parameter K that will be discussed later.

Algorithm 1: Joint server on-off and energy control policy

Input: $N(t-1), L(t), P(t), \mathcal{E}(t)$.
Output: $N(t), b(t)$.
/* Server on-off control μ^M */
$N(t) = \arg\min_N \ \hat{h}((L(t), N(t-1)), N)$
Compute $E(t)$ by Eq. (1) with $L(t), N(t-1), N(t)$
/* Energy control β^L */
if $\mathcal{E}(t) \geq KP(t)$ **then**
  $b(t) = -b_{\max}$
else
  $b(t) = E(t)$

Server On-off Control Component μ^M**:** We rely on the MDP model mentioned in Section IV-B, and in time-slot t, μ^M chooses the decision $N(t)$ that minimizes Eq. (16) for current sub system state $S_1(t)$.

Intuitively, since switching a server from the off-state to the on-state incurs turn-on cost, in order to minimize the average energy consumption of the system, a smart policy should not only focus on the instantaneous energy consumption, but also consider the future possible consumption. For instance, if the number of current tasks is low, to minimize the instantaneous energy consumption, turning off some fraction of the servers is a good choice. However, if the number of future tasks were to become high with high probability, turning off servers is not a good option since we save instantaneous energy consumption but consume more energy by having to turn on servers in the future. With the known transition probability of tasks, we rely on our MDP model to predict the future energy consumption, which enables us to avoid the abovementioned situations and further minimize the overall energy consumption by taking future arrivals into consideration.

To solve Eqs. (16) and (17), we use the value iteration method [13] to compute the potential function values of all state decision pairs in the beginning, which is described in Algorithm 2.

Energy Control Component β^L**:** The energy control component has a threshold structure, with a free control parameter K. We will show in the performance analysis that K describes the gap between the performance of the proposed control policy and the optimal cost that can be achieved.

On one hand, when the emptiness of the battery is no less than the electricity price times the control parameter, it indicates that either the emptiness of battery is large, or the

Algorithm 2: Value iteration for finding the potential functions value of state decision pairs

Input: ϵ, $\Pr(S_1^{'}|S_1, N), \forall (S_1, N)$ s.t. $S_1 \in \mathcal{S}_1, N = 1, 2, \cdots, N_{\max}$.

Output: $h(S_1)$ for all $S_1 \in \mathcal{S}_1$.

/* Initial procedure: */

$h^0(S_1) = 0, \forall S_1 \in \mathcal{S}_1$

/* Iteration procedure: */

do

  Compute $\hat{E}(S_1, N)$ by Eq. (1)

  $\hat{h}^k(S_1, N) = \hat{E}(S_1, N) + \sum_{S_1^{'} \in \mathcal{S}_1} \Pr(S_1^{'}|S_1, N) h^{k-1}(S_1^{'})$

  $h^k(S_1) = \min_N \hat{h}^k(S_1, N)$

while $\max_{S_1 \in \mathcal{S}_1} h^k(S_1) - h^{k-1}(S_1) > \epsilon$;

electricity price is relatively low. In both cases, to keep the emptiness of battery stable, and to purchase energy from the power grid because of its low price, and store it in the battery for future use, the decision is to charge the battery at the maximum possible charging rate. On the other hand, if the emptiness of battery is relatively small, which indicates that the battery has sufficient energy stored in it, or if the electricity price is relatively high, the decision is to draw energy from the battery to fully support the energy consumption of the system in that time-slot. This avoids purchasing from the power grid when the price is high, thus saving the cost.

Here, K is a control parameter that trade-offs monetary costs and the battery level. As K increases, the energy control component gives more importance to monetary costs than the battery level, resulting in energy purchase from the power grid only when energy price is small. It potentially achieves smaller cost for purchasing energy. However, we note that, when $\mathcal{E}(t)$ is greater than the value $KP(t)$, the battery is charged, which means that $\mathcal{E}(t)$ is never greater than $KP_{\max} + E_{\max}$. This implies that the emptiness of battery $\mathcal{E}(t)$ is bounded by a constant, and thus the proposed joint policy (μ^M, β^L) is feasible to **P2**. Meanwhile, as K increases, the threshold to charge the battery becomes large, which means that $\mathcal{E}(t)$ becomes large and potentially a larger battery size is required.

We note that the server on-off control component μ^M does not depend on any information of the energy price or the battery state, and once $N(t)$ and thus $E(t)$ are determined, the energy control component β^L makes the decision only based on the electricity price and the current battery state. Intuitively, a control policy in which these two components are jointly optimized, meaning that the server on-off decision also considers the battery state, the electricity price and the amount of energy drawn from the battery, while the energy control component takes the number of on servers that are decided to be turned on into account, can perform better than the proposed control policy. Surprisingly, we have found that this control policy (μ^M, β^L) performs arbitrarily close to any possible online policy, which means that it is asymptotically optimal, as shown in the following subsection.

D. Performance Analysis

Let $(N^{\mu,\beta}(t), b^{\mu,\beta}(t))$ denote the actual control by the policy (μ, β), and $G(\mu, \beta)$ denote the value of the objective function under the online policy (μ, β), i.e.,

$$G(\mu, \beta) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[P(t) \left(E^{\mu,\beta}(t) - b^{\mu,\beta}(t)\right)].$$

where $E^{\mu,\beta}(t)$ is computed with Eq (1) with decisions $N^{\mu,\beta}(t)$ and $N^{\mu,\beta}(t-1)$. We denote Π^M, Π^L as the set of feasible control policies in which the server on-off decision component or the energy control component is the same as the proposed server on-off decision component or the energy control component. In other words, $\mu = \mu^M$ if $(\mu, \beta) \in \Pi^M$, $\beta = \beta^L$ if $(\mu, \beta) \in \Pi^L$.

To obtain the performance of the proposed control policy (μ^M, β^L), we first prove the following two lemmas.

Lemma 1. *For any feasible control policy (μ, β) to* **P2***, there must exist a feasible control policy $(\mu, \beta^L) \in \Pi^L$ such that it satisfies the following inequality:*

$$G(\mu, \beta^L) \leq G(\mu, \beta) + C/K, \tag{18}$$

where $C \triangleq \lambda_{\max}^2 + \min\{b_{\max}, E_{\max}\}^2$.

Proof. We divide the proof into three steps. In step 1, we show that for any fixed server on-off control component μ, there exists an optimal stationary energy control component which minimizes the objective function. In step 2, we give a condition when Eq. (18) is satisfied. Finally we show that the proposed energy control component β^L satisfies the condition in the second step, and thus the proof is completed.

Step 1: *Claim:* Given any fixed server on-off control component μ, there exists an optimal stationary energy control component β^* which satisfies

$$\mathbb{E}\left[b^*(t) - \lambda(t)|\mathcal{E}(t)\right] = 0 \tag{19}$$

$$\mathbb{E}\left[P(t)(E(t) - b^*(t))\right] = G^*, \tag{20}$$

where $G^* = \min_\beta G(\mu, \beta)$.

The proof of this claim is similar to the Theorem 1 in [14] and thus we omit the proof for brevity here. In the following steps, we focus on β^*, and try to prove

$$G(\mu, \beta^L(t)) \leq G^* + C/K \tag{21}$$

Step 2: We define Lyapunov function $V(t)$ as $V(t) = \frac{1}{2}\mathcal{E}(t)^2$, and one-step conditional Lyapunov drift $\Delta V(t)$ as $\Delta V(t) = \mathbb{E}[V(t+1) - V(t)|\mathcal{E}(t)]$.

Claim: An energy control component β satisfies Eq. (21) if it satisfies

$$\Delta V(t) + K\mathbb{E}\left[P(t)\left(E(t) - b(t)\right)|\mathcal{E}(t)\right] \leq C + KG^* \tag{22}$$

Reason: Suppose Eq. (22) holds, we sum over it from $t = 1$ to $t = T$, and divide both sides by TK, then we have

$$\frac{\Delta L(T) - \Delta L(1)}{TK} + \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[P(t)(E(t) - b(t))|\mathcal{E}(t)\right] \leq G^* + C/K$$

By letting T go to infinity, then the first term on the left-hand side diminishes, and we obtain Eq. (21).

Step 3: *Claim:* The proposed energy control component β^L satisfies Eq. (22).

Reason: By the definition of $\Delta V(t)$, we have

$$\begin{aligned}\Delta V(t) = & \ \mathbb{E}[V(t+1) - V(t)|\mathcal{E}(t)] \\ \leq & \ \frac{1}{2}\mathbb{E}\left[(\mathcal{E}(t) + b(t) - \lambda(t))^2 - \mathcal{E}(t)^2|\mathcal{E}(t)\right] \\ \leq & \ \mathbb{E}\left[\mathcal{E}(t)(b(t) - \lambda(t)) + b^2(t) + \lambda^2(t)|\mathcal{E}(t)\right] \\ \leq & \ C + \mathbb{E}\left[\mathcal{E}(t)\left(b(t) - \lambda(t)\right)|\mathcal{E}(t)\right]\end{aligned} \tag{23}$$

Adding K times one time-slot cost on both sides of Eq. (23), we have

$$\begin{aligned}& \Delta V(t) + K\mathbb{E}\left[P(t)\left(E(t) - b(t)\right)|\mathcal{E}(t)\right] \\ \leq & \ C + \mathbb{E}\left[\mathcal{E}(t)\left(b(t) - \lambda(t)\right)|\mathcal{E}(t)\right] \\ & + K\mathbb{E}\left[P(t)\left(E(t) - b(t)\right)|\mathcal{E}(t)\right]\end{aligned} \tag{24}$$

Let $b(t)$ be $b^*(t)$, which is the optimal stationary energy control component, in Eq. (24). With Eq. (19) and Eq. (20), we know that there exists one energy control decision β^* that satisfies Eq. (22). If an energy control decision tries to minimize the right-hand side of Eq. (24) in each time-slot, it must also satisfy Eq. (22). We then obtain an optimization problem which is stated below:

$$\begin{aligned}\textbf{P3:} \ \min_{\mathbf{b}} \ & C + \mathbb{E}\left[\mathcal{E}(t)\left(b(t) - \lambda(t)\right)|\mathcal{E}(t)\right] \\ & + K\mathbb{E}\left[P(t)\left(E(t) - b(t)\right)|\mathcal{E}(t)\right]\end{aligned} \tag{25}$$

We can easily verify that the proposed energy control component β^L is the solution to **P3**, thus it satisfies Eq. (22) and therefore Eq (21). By the definition of G^*, we complete the proof. □

Lemma 2. *Let $d_{\max}$ denote the maximum amount of energy to be purchased from the grid in each time-slot, i.e., $d_{\max} = \sup_t d(t)$. If $d_{\max} \leq b_{\max}$, given any feasible online control policy (μ, β) to* **P2***, there must exist a feasible online control policy $(\mu^M, \beta^M) \in \Pi^M$, such that it achieves*

$$G(\mu^M, \beta^M) \leq G(\mu, \beta). \tag{26}$$

Proof. For the convenience, we use superscript M to denote the system state and decision variables of such feasible control policy that satisfies Eq. (26). Recall that in each time-slot t, given the system energy consumption $E^M(t)$, once $d^M(t)$ is determined, $b^M(t) = E^M(t) - d^M(t)$ is fixed. In this proof, $d^M(t)$ is referred to as the energy control decision. We prove this lemma by constructing a policy (μ^M, β^M) that satisfies Eq. (26).

In each time-slot t, the energy control policy β^M chooses the value $d^M(t)$ such that

$$d^M(t) \triangleq \min\{d(t), E^M(t) + \mathcal{E}^M(t), d_{\max} - E^M(t)\}. \tag{27}$$

From Eq. (27), we can see that, β^M modifies the given energy control policy β by keeping track of β and making it feasible in each time-slot. According to the definition of $d^M(t)$, we have $d^M(t) \leq d(t)$, which implies Eq. (26). The next step is to show that (μ^M, β^M) is feasible for **P2**. Before that, we make the following claim.

Claim: The expected time-average energy consumption $E^M(t)$ under online policy (μ^M, β^M), is no larger than the expected time-average energy consumption $E(t)$ under any given policy (μ, β), i.e.,

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E^M(t) \leq \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E(t) \tag{28}$$

Please see our technical report [15] for the detailed proof of Eq. (28). Essentially, Eq. (28) says that the proposed server on-off control μ^M achieves the smallest amount of average energy consumption among all online policies.

With the assumption $d_{\max} \leq b_{\max}$, we have $b(t) = E(t) - d(t) \geq -b_{\max}$, and Eq.(13) becomes:

$$\mathcal{E}(t+1) = \max\{0, \mathcal{E}(t) + [E(t) - d(t)] - \lambda(t)\}.$$

Since the given policy (μ, β) is feasible for **P2**, Eq. (14) holds, and we have:

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}[E(t) - d(t)] \leq \bar{\lambda}.$$

We then focus on the long-term average of $E^M(t) - d^M(t)$, and have:

$$\begin{aligned}& \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\left[E^M(t) - d^M(t)\right] \\ = & \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\left[E^M(t) - d(t)\right] \\ = & \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E^M(t) - \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} d(t) \\ \leq & \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E(t) - \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} d(t) \\ \leq & \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}[E(t) - d(t)] \leq \bar{\lambda}.\end{aligned}$$

Thus, Eq.(14) holds under the control policy (μ^M, β^M). We then conclude that (μ^M, β^M) is feasible for **P2**, and it satisfies Eq. (26). □

Using Lemmas 1 and 2, we are ready to obtain the following theorem.

Theorem 1. *Let $d_{\max}$ denote the maximum amount of energy to be purchased from the grid in each time-slot, i.e., $d_{\max} = \sup_t d(t)$. If $d_{\max} \leq b_{\max}$, given any feasible control policy (μ, β) to* **P2***, the proposed joint server on-off and energy control policy (μ^M, β^L) achieves:*

$$G(\mu^M, \beta^L) \leq G(\mu, \beta) + C/K.$$

Proof. With Lemma 2, we know that there exists a control policy $(\mu^M, \beta^M) \in \Pi^M$ which satisfies Eq. (26). Then we let μ be μ^M and β be β^M in Eq. (18), and together with Eq. (26) it completes the proof. □

Since Theorem 1 holds for any feasible control policy, we can easily obtain the following corollary.

Corollary 1.1. *Let $d_{\max}$ denote the maximum amount of energy to be purchased from the grid in each time-slot. When $d_{\max} \leq b_{\max}$, the proposed joint server on-off and energy control policy (μ^M, β^L) achieves a cost arbitrarily close to the optimal average cost, by increasing the control parameter K, with the gap shrinking in $\mathcal{O}(1/K)$.*

Complexity: To analyze the time complexity of the proposed algorithm (μ^M, β^L), we note that the server on-off decisions are computed offline using Algorithm 2, while in each iteration the time complexity is $\mathcal{O}(|\mathcal{S}_1|^2 n)$, with $n = N_{\max}$ and $|\mathcal{S}_1| = (L_{\max} - L_{\min})N_{\max}$. Although the time complexity to compute it offline is not low, once the potential function h is obtained with sufficiently small difference, to make the decision in each time-slot, μ^M maps the sub system state to the decision by looking at a decision table, with time complexity $\mathcal{O}(1)$. Since energy control policy β^L has a threshold structure with linear threshold value, the complexity of β^L is also $\mathcal{O}(1)$.

We note that the proposed control policy (μ^M, β^L) and its performance results presented here are for solving **P2**, which does not have finite battery capacity constraint. In reality, a battery, however, does have finite capacity. However, by carefully choosing the value of the control parameter K, the proposed control policy (μ^M, β^L) can be applied to the case with finite battery capacity.

Recall that when $\mathcal{E}(t)$ is greater than the value $KP(t)$, the battery is charged, which means that $\mathcal{E}(t)$ is never greater than $KP_{\max} + E_{\max}$. Therefore, given the maximum battery capacity $B_{\max}$, if the control parameter K is chosen such that $B_{\max} \geq E_{\max} + KP_{\max}$, the emptiness of battery $\mathcal{E}(t)$ is bounded by $B_{\max}$ under the policy (μ^M, β^L) with such K, which means that the proposed control policy is also feasible for **P1** with finite battery capacity.

Although the theoretical results of (μ^M, β^L) do not hold for **P1**, in Section V, we will show via simulations that the proposed policy (μ^M, β^L) actually achieves good performance with reasonably large battery capacity.

V. Numerical Results

A. Simulation Setup

We summarize the system parameter values in Table I. We conduct our simulation over a cloud system with 30 identical servers. Each time-slot is set to be 10 minutes, so that the wear-and-tear of power cycling matches that of operating cost [6]. In each time-slot, the number of task arrivals is uniformly distributed from 11 to 20, and the number of time-slots needed for each job follows a geometric distribution, with success probability p. To limit the number of tasks served in each time-slot, $L_{\min}$ and $L_{\max}$ are set to be 11 to 100 respectively. The amount of harvesting energy $\lambda(t)$ is uniformly distributed from $\lambda_{\min} = 10$ to $\lambda_{\max} = 100$. We carry out our simulation with the real market electricity price obtained from [16], as shown in Fig. 2.

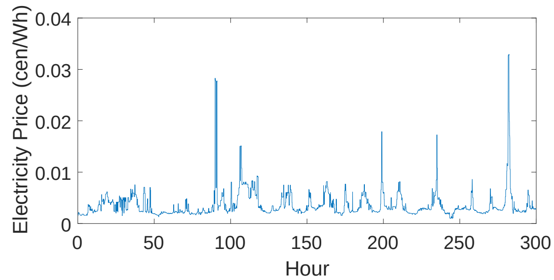


Fig. 2: Real-time Energy Price (10/5/2017-10/17/2017) in [16]

TABLE I: Simulation Parameter Settings

Parameters	Value
a time-slot	10 minutes
$\lambda(t)$	$U(10, 100)$ Wh/time-slot
on-cost C_{on}	10 Wh/time-slot
turn-on cost $C_{\text{t/o}}$	60, 120 Wh
service cost function	$f(L) = L^3$ Wh
maximum charging rate	$-b_{\max} = 3000$ Wh/time-slot
traffic departure probability p	0.25
$L_{\min}, L_{\max}$	$10, 100$

Benchmark control policies: We compare our control policy with the following three benchmark policies.

Baseline policy: This policy makes a myopic server on-off decision that minimizes the current energy consumption $E(t)$. The battery is only charged with the renewable energy, and the rest of required energy is purchased from the power grid.

Server on-off control only policy: In this control policy, the proposed server on-off control component μ^M is activated, while in each time-slot, the system first uses the energy stored in the battery, and if it is not sufficient, the rest of the energy demand is met by purchasing energy from the power grid.

Energy control only policy: In this control policy, the proposed energy control component β^L is activated, while in each time-slot, the system chooses to turn on the number of servers which minimizes the instantaneous energy consumption, without considering the future energy cost or utilizing the statistic information of tasks.

In both our proposed joint policy and energy control only policy, we choose K such that $KP_{\max} + E_{\max} = B_{\max}$, which exploits the maximum potential of the battery as mentioned in Section IV.

B. Key Results

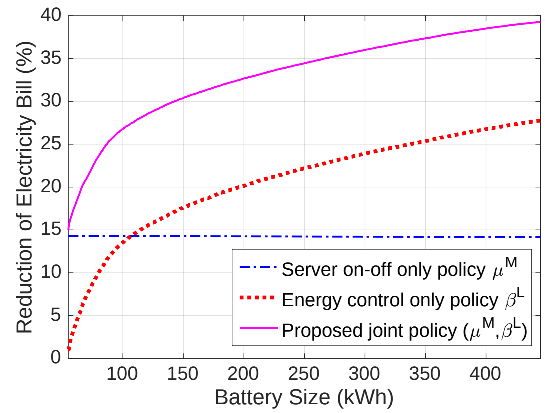
Fig. 3 shows the percentage reduction in electricity bill of three policies with respect to the battery capacity, compared with the baseline policy, with two different turn-on cost values. As shown in these figures, the gain of energy control only policy increases as the size of the battery increases, and reaches 20% under both cases with battery capacity 200 kWh [17][1]. This is because with larger battery capacity $B_{\max}$, the capability of the battery to smooth the electricity price, i.e., the capability to avoid purchasing expensive energy from the power grid, becomes stronger. The server on-off only policy,

[1]From [17], small cloud systems are usually equipped with 200 kWh battery.

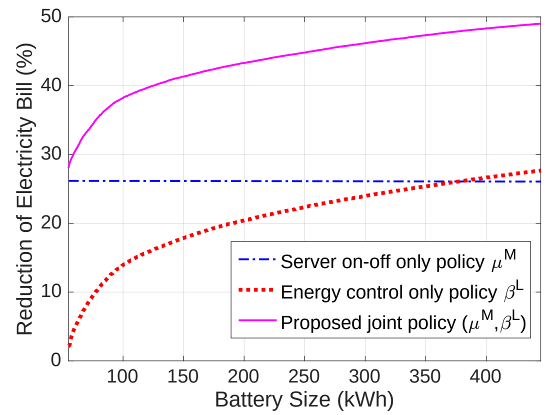
however, does not consider the battery as a buffer to take advantage of the fluctuation of the electricity price, and its performance remains the same regardless of the battery size.

Meanwhile, the server on-off only policy utilizes the history information of tasks to minimize the energy consumption. By comparing Figs. 3(a) and 3(b), we can see that, with larger turn-on energy cost, the gain obtained by the server on-off policy becomes larger. With the prediction of the number of unit tasks in the future, the server on-off only policy avoids to turn on servers in the future, by keeping more servers on in the current time-slot. It saves more turn-on energy consumption by sacrificing less instantaneous energy cost, which becomes more crucial with larger turn-on energy consumption. To obtain the same cost savings as the proposed policy, the energy only control policy needs a much larger battery capacity, which incurs higher costs. The simulation results also show that, the actual energy consumption of the policies with server on-off control under small and large turn-on cost cases are 504Wh and 525Wh per time-slot, which reduce 13% and 24% compared with the policies without server on-off control.

Our proposed joint policy, by both activating energy control and server on-off control, not only reduces the overall energy consumption, but also exploits the battery buffer, to store energy when the price is relatively low, and thus leads to a monetary cost savings of 33% and 43% with small and large turn-on costs respectively, under a reasonable battery capacity of 200 kWh.



(a) Percentage reduction in electricity cost compared to Baseline with small turn-on cost, $C_{t/o} = 60$Wh per time.



(b) Percentage reduction in electricity cost compared to Baseline with large turn-on cost, $C_{t/o} = 120$Wh per time.

Fig. 3: Performance v.s. Battery Capacity.

VI. Conclusion

In this work, we focus on cloud systems with renewable resources and batteries, with the aim of designing a joint server on-off and energy control algorithm to reduce the electricity cost of cloud systems. Our proposed online policy is proven to achieve electricity cost that is arbitrarily close to that of an optimal online algorithm when the battery capacity is unbounded. In addition, we show via simulations that the proposed server on-off control and energy control policy reduces both the electricity cost and energy consumption dramatically under reasonable battery capacity.

References

[1] Y. Shi, B. Xu, B. Zhang, and D. Wang, "Leveraging Energy Storage to Optimize Data Center Electricity Cost in Emerging Power Markets," in *Proc. of ACM E-energy*, 2016, p. 18.

[2] Z. Iverson, A. Achuthan, P. Marzocca, and D. Aidun, "Optimal Design of Hybrid Renewable Energy Systems (HRES) Using Hydrogen Storage Technology for Data Center Applications," *Renewable energy*, vol. 52, pp. 79–87, 2013.

[3] C. Li, A. Qouneh, and T. Li, "iSwitch: Coordinating and Optimizing Renewable Energy Powered Server Clusters," in *Proc. of IEEE ISCA*. IEEE, 2012, pp. 512–523.

[4] Y. Zhang, Y. Wang, and X. Wang, "Greenware: Greening Cloud-scale Data Centers to Maximize the Use of Renewable Energy," in *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer, 2011, pp. 143–164.

[5] L. A. Barroso and U. Hölzle, "The Case for Energy-Proportional Computing," *IEEE Computer*, vol. 40, no. 12, 2007.

[6] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic Right-sizing for Power-Proportional Data Centers," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 5, pp. 1378–1391, 2013.

[7] C. Hasan and Z. J. Haas, "Deadline-aware Energy Management in Data Centers," in *Proc. of IEEE CloudCom*, 2016, pp. 79–84.

[8] W. Zhang, Y. Wen, L. L. Lai, F. Liu, and R. Fan, "Electricity Cost Minimization for Interruptible Workload in Datacenter Servers," *IEEE Transactions on Services Computing*, 2017.

[9] H. Zhou, J. Yao, H. Guan, and X. Liu, "Comprehensive Understanding of operation Cost Reduction Using Energy Storage for IDCS," in *Proc. of IEEE INFOCOM*, 2015, pp. 2623–2631.

[10] Y. Guo and Y. Fang, "Electricity Cost Saving Strategy in Data Centers by Using Energy Storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1149–1160, 2013.

[11] M. Dayarathna, Y. Wen, and R. Fan, "Data Center Energy Consumption Modeling: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732–794, 2016.

[12] M. Bramson *et al.*, "Stability of Queueing Networks," *Probability Surveys*, vol. 5, pp. 169–345, 2008.

[13] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena scientific Belmont, MA, 1995, vol. 1, no. 2.

[14] M. J. Neely, "Energy Optimal Control for Time-Varying Wireless Networks," *IEEE transactions on Information Theory*, vol. 52, no. 7, pp. 2915–2934, 2006.

[15] J. Liu, J. Lee, N. B. Shroff, and P. Sinha. "A Near-Optimal Control Policy in Cloud Systems with Renewable Sources and Time-dependent Energy Price." Technical Report. [Online]. Available: https://www.dropbox.com/s/hxqfkgsmvfmx4g2/TechnicalReport.pdf?dl=0

[16] ISO New England Inc, "Real-Time Maps and Charts," *http://www.iso-ne.com/isoexpress/*, 2017.

[17] APC by Schneider Electric, "Data Center and Facility 3 Phase UPS," *http://www.apc.com/us/en/*, 2018.