

# Workload Shaping to Mitigate Variability in Renewable Power Use by Data Centers

Muhammad Abdullah Adnan and Rajesh K. Gupta  
University of California San Diego

**Abstract**—This paper explores the opportunity for energy saving in data centers using the flexibility from the Service Level Agreements (SLAs) and proposes a novel approach for scheduling workload that incorporates use of renewable energy sources. We investigate how much renewable power to store and how much workload to delay for increasing renewable usage while meeting latency constraints. We present an LP formulation for mitigating variability in renewable generation by dynamic deferral and give two online algorithms to determine optimal balance of workload deferral and power use. We prove the feasibility of the online algorithms and show that their worst case performances are bounded by constant factors with respect to the offline formulation. We validate our algorithms by trace-driven simulation on MapReduce workload and collected and publicly available wind and solar power generation data. Results show that the algorithms give 20-30% energy-savings compared to the naïve ‘follow the workload’ policy.

## I. INTRODUCTION

Emergence of cloud computing drives the continued proliferation of data centers around the world accounting for an estimated 61 million MWh per year at a cost of \$4.5B [1]. Several measures have been taken to reduce operational and environmental costs. These include integration of renewable energy [2], [3], improved cooling [4], [5] and improved workload scheduling methods [5], [6]. Improved scheduling is attractive for its broad applicability across hardware platforms. Scheduling methods, of course, have to be constrained by the time it takes to complete the computing workload. Such constraints on workload completion latency are generally specified through Service Level Agreements (SLAs) between the customers and data center operators. To an operator, a SLA also provides a source of flexibility in workload scheduling to improve performance and efficiency [7].

Integration of renewable power generation – through solar or wind – presents another opportunity for flexibility in workload execution to meet a given SLA. Balanced against the necessary cost of energy storage, variability in renewable power generation directly affects the cost of electricity. To counter against fluctuations in electricity availability and prices, data center operators have used peak power shaving techniques to store excess energy during periods of high power availability [8], [9]. Since renewable power source is physically close to the data center, these can be integrated in a way to shape workload demands optimally across time and location. The goal of this paper is to align power generation variabilities against workload flexibility to jointly optimize workload deferral and renewable storage costs for non-renewable energy savings.

In this paper, we present analysis of the effectiveness of power usage in data centers under guarantees on average latency or deadline. We use the latency (deadline/average latency) information to defer some tasks so that we can reduce the total energy consumption for executing the workload when renewable energy is available. We determine the portion of the released workload to be executed at the current time and the portions to be deferred to be executed at later time slots without violating latency constraints. Due to the variation in workload and deviation in renewable power generation, workload shifting and peak power shaving techniques have

been proposed earlier to reduce the effect of demand-response pricing on the amount of work done [10]. But demand response pricing has been the only way to balance these two variations involving significant user interactions and prediction techniques. This paper takes a unified view on these two variations and develops algorithms to determine the optimal point of operation for data centers. We present a mathematical formulation for joint optimization of workload deferral and shaping renewable generation via dynamic storage for saving energy in data centers.

This paper makes three contributions. First, we present a linear programming formulation for workload deferral and renewable integration in data centers. The formulation not only determines the power usage but also determines the assignment of workload for each time slot. As a result the amount of power storage can be determined easily and resources can be allocated accordingly.

Second, we design two optimization based online algorithms depending on the nature of the latency requirements. For uniform requirement (e.g. all the jobs have same deadline), our algorithm named *Follow the Renewable with Deferral (FRD( $\delta$ ))*, looks ahead  $\delta$  slots to optimize the total energy consumption. The algorithm defers some workload to execute when renewable power generation is high. For nonuniform deadline, we design a *Follow the Renewable with Nonuniform Deferral (FRND)* algorithm that distributes the workloads in adjacent time slots according to renewable generation and thus reduces energy consumption. We prove the feasibility of the solutions and show that the performance of the online algorithms are bounded by a constant factor with respect to the offline formulation. To the best of our knowledge, this is the first algorithmic approach for renewable integration in data centers considering workload deferral with bounded latency.

Third, we validate our algorithms using MapReduce traces (representative workload for data centers) and renewable generation traces (solar and wind) and evaluate cost savings achieved via dynamic deferral. We run simulations to explore a wide range of settings and show significant savings in each of them. We compare our methods against a baseline model of ‘follow the workload’ execution. Over a period of 24 hours, we find more than 30% total energy saving for FRND and around 20% total energy saving for FRD( $\delta$ ) even for small deadline requirements.

The rest of the paper is organized as follows. Section II presents the model that we use to formulate the optimization and gives the offline formulation considering hard deadline requirements for the jobs. In Section III, we present the FRD( $\delta$ ) algorithm for determining power usage and workload assignment dynamically when the deadline (latency requirement) is uniform. In Section IV, we illustrate the FRND algorithm with nonuniform deadline. Section V illustrates the simulation results. In Section VI, we describe the state of the art research related to energy efficient workload scheduling in data centers and Section VII concludes the paper.

## II. MODEL FORMULATION

In this section, we describe the model we use for workload scheduling to mitigate variability in renewable power generation via dynamic deferral.

### A. Workload Model

The workload model is over a time frame  $t \in \{0, 1, \dots, T\}$  where  $T$  can be arbitrarily large. In practice,  $T$  can be a year and the length of a time slot,  $\tau$ , can be as small as minutes (the granularity in which the renewable power data is available). Let  $L_t$  be the amount of workload released at time slot  $t$ . The workload  $L_t$  can contain short jobs and long jobs. If the length  $\ell$  of a job is greater than time slot length  $\tau$  then we decompose the job into small pieces ( $\leq \tau$ ) each of which is released after the execution of the preceding piece. Thus long jobs are decomposed into small jobs. Hence we do not distinguish each job, rather deal with the total amount of workload. Due to page limitation, we omit details of the length estimation and decomposition procedure in this paper; the details can be found in a technical report [11].

In our model, jobs have latency requirements specified in the SLAs. The latency requirements are specified in terms of hard/soft deadlines or average latency of completion. In the rest of this paper, we consider hard deadline requirements for the jobs. However, our model and algorithms can be extended for general latency requirements as discussed in Section VII. So, each job has a deadline  $D$  (in terms of number of slots) associated with it, where  $D$  is a nonnegative integer. A job released at time  $t$ , needs to be executed by time slot  $t + D$ . The value of  $D$  can be zero for interactive jobs and large for batch-like jobs. If the job is long and decomposed into smaller pieces, then we need to assign deadline to each individual piece. If the long job is preemptive then we assign deadline  $\lfloor D/\ell \rfloor - 1$  to each of the small pieces and for a non-preemptive job, we assign deadline of  $D - \ell$  to the first piece and deadlines of zeros to the other pieces. To simplify analysis, we first consider the case of uniform deadlines, that is, deadline is uniform for all the jobs, followed by non-uniform deadline case in Section IV. Since the deadline  $D$  is uniform for all the jobs, the total amount of work  $L_t$  must be executed by the end of time slot  $t + D$ . Since  $L_t$  varies over time, we refer to it as a *workload curve*.

A data center is a collection of servers that can execute jobs and consumes energy proportional to the amount of workload executed. In addition, a data center is equipped with local renewable power (solar/wind) generation plant. Let  $R_t$  be the amount of renewable power generated at time  $t$ . Some portion of this power  $R_t$  will be used to execute workload and some portion will be stored. We normalize  $R_t$  by the power requirement for the execution of a job at any time slot, i.e.,  $R_t$  denotes the number of jobs that can be executed at time  $t$ . Since  $R_t$  varies over time, we often refer to it as a *renewable curve*. Suppose,  $p_t$  is the amount of power used for workload execution at time  $t$  and  $x_t$  is the amount of workload execution at time  $t$ . Since  $p_t$  power will be used to execute the workload  $x_t$ , we have  $0 \leq x_t \leq p_t$  for all  $t$ . From the data center perspective, we focus on two important decisions during each time slot  $t$ : (i) determining  $p_t$ , the power usage, and (ii) determining  $x_t$ , workload execution at time  $t$ .

### B. Cost Model

The goal of this paper is to maximize the renewable energy usage in data centers. The energy cost function consists of two parts: execution cost and power wastage cost. *Execution cost* is the cost for executing the workload which in our model is proportional to the assigned workload. We use the common

model for energy cost for typical servers which is an affine function:

$$C(x_t) = e_0 + e_1 x_t$$

where  $e_0$  and  $e_1$  are constants (e.g. see [12]) and  $x_t$  is the assigned workload (utilization) of the servers at a time slot. Although we use this general model for cost function, other models considering nonlinear parameters such as temperature, frequency can be adopted in the model which will make the optimization problem nonlinear. Our algorithms can be applied for such nonlinear models by using techniques for solving nonlinear optimizations as each optimization is considered as a single independent step in the algorithms.

*Power wastage cost* is the cost incurred for not using renewable energy at each time. This cost is analogous to the storage cost for renewable energy. Power wastage cost at time  $t$  is defined as follows:

$$W_t = \beta |R_t - p_t|$$

where  $\beta$  is a constant for the cost of energy wastage or storage.

### C. Optimization Problem

Given the models above, the goal of a data center is to determine the power usage  $p_t$  and the assignment  $x_t$  to minimize the total cost during  $[1, T]$ .

To formulate the constraint that no assignment violates any deadline we define delayed workload  $l_t$  with maximum deadline  $D$ .

$$l_t = \begin{cases} 0 & \text{if } t \leq D, \\ L_{t-D} & \text{otherwise.} \end{cases}$$

We call the delayed curve  $l_t$  for the workload as *deadline curve*. Thus we have two fundamental constraints on the assignment of workload for all  $t$ :

$$(C1) \text{ Deadline Constraint: } \sum_{j=1}^t l_j \leq \sum_{j=1}^t x_j$$

$$(C2) \text{ Release Constraint: } \sum_{j=1}^t x_j \leq \sum_{j=1}^t L_j$$

Condition (C1) says that all the jobs assigned up to time  $t$  cannot violate deadline and Condition (C2) says that the assigned workload up to time  $t$  cannot be greater than the total released workload up to time  $t$ . In addition to the constraints for workload execution, we have constraints on renewable energy usage. Due to limited battery size, there is a maximum limit  $S$  on the energy that can be stored.

$$(C3) \text{ Storage Constraint: } 0 \leq \sum_{j=1}^t R_j - \sum_{j=1}^t p_j \leq S$$

Using these constraints we can formulate the optimization by minimizing execution cost and maximizing renewable energy usage:

$$\begin{aligned} \min_{x_t, p_t} \quad & \sum_{t=1}^T C(x_t) + \beta \sum_{t=1}^T |R_t - p_t| \quad (1) \\ \text{subject to} \quad & \sum_{j=1}^t l_j \leq \sum_{j=1}^t x_j \leq \sum_{j=1}^t L_j \quad \forall t \\ & \sum_{j=1}^T x_j = \sum_{j=1}^T L_j \\ & 0 \leq \sum_{j=1}^t R_j - \sum_{j=1}^t p_j \leq S \quad \forall t \\ & 0 \leq x_t \leq p_t \quad \forall t \end{aligned}$$

Since the operating cost function  $C(\cdot)$  is an affine function, the objective function is linear as well as the constraints. Hence

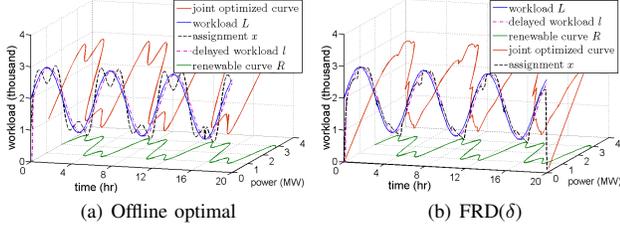


Fig. 1. Illustration of (a) offline optimal solution and (b) FRD( $\delta$ ) for workload generated using sine functions; time slot length = 6 minutes,  $D = 2$ ,  $\delta = 1$ .

optimization (1) is a linear program. Note that assignment  $x_t$  in this formulation is not constrained to be an integer. This is acceptable because data centers consist of thousands of active servers and thousands of jobs to compute at any time. Then we can round the resulting solution with minimal increase in cost. Figure 1(a) illustrates the offline optimal solutions for  $x_t$  and  $p_t$  for a dynamic workload generated using sinusoidal functions. The performance of the optimal offline solution on two realistic workloads are provided in Section V.

### III. FOLLOW RENEWABLE WITH DEFERRAL

In this section, we consider the online case, where at any time  $t$ , we neither have information about the future workload  $L_{t'}$ , nor we have information about future renewable generation  $R_{t'}$ , for  $t' > t$ . At each time  $t$ , we determine the  $x_t$  and  $p_t$  by applying optimization over the already released unassigned workload which has deadline in future  $D$  slots. Note that the workload released at or before  $t$ , can not be delayed to be assigned after time slot  $t + D$ . Hence we do not optimize over more than  $D + 1$  slots. We simplify the online optimization by solving only for  $p_t$  and determine  $x_t$  by making  $x_t = p_t$  at time  $t$ . This makes the online algorithm not to waste power that is not stored. But the cost due to power usage in the online algorithm may be higher than the offline algorithm. Thus our goal is to design strategies to reduce the power usage cost by shaping the workload curve according to the renewable generation curve via dynamic deferral. In the online algorithm, we reduce the power usage cost by optimizing the total cost for the interval  $[t, t + D]$ .

#### A. Renewable Power Prediction Model

We now illustrate our model for predicting the renewable energy  $\tilde{R}_j$  in the future time slots  $j \in [t + 1, t + D]$ . There are several renewable prediction models (e.g. ARIMA, EWMA [13] etc.) based on time series prediction which often ignore seasonal/historical components. To capture the hourly and weekly trends, we use two different methods to estimate the mean and variance of the renewable power generation. In other words, we model future power generations within a 24-hour time-frame by Gaussian random variables with known means, which are the predicted power generations, and some estimated variance. The mean for the Gaussian distribution is predicted by the widely used moving average method for time series. The variance for the Gaussian distribution is estimated from the history by the weighted average power prediction filter proposed in [14]. In this model, variances are predicted by linear regression from the previous power generations from yesterday, the day before yesterday and the same day last week. By using two different methods for mean and variance, we exploit both the temporal and historical correlation of renewable generation. To facilitate the future renewable energy prediction, we denote the set of the time slots in a 24-Hour time frame by  $\mathcal{K} \subset T$ . Let  $\tilde{\mu}_\kappa[\chi]$  and  $\tilde{\sigma}_\kappa[\chi]$  be the predicted means and standard deviations for each time slot  $\kappa$

on day  $\chi$ . Then the mean of the prediction model for Gaussian distribution is obtained as follows:

$$\tilde{\mu}_\kappa = \varepsilon_0 + \sum_{j=0}^D \varepsilon_{\kappa-j} R_{\kappa-j}, \quad \forall \kappa \in \mathcal{K}$$

Here,  $\varepsilon_j$  are the coefficients for the moving average method which can be estimated by training the model over the previous day power generations. The variance parameter  $\tilde{\sigma}_\kappa[\chi]$  is estimated from the history using the following equation:

$$\tilde{\sigma}_\kappa[\chi] = k_1 \sigma_\kappa[\chi - 1] + k_2 \sigma_\kappa[\chi - 2] + k_7 \sigma_\kappa[\chi - 7], \quad \forall i \in n, \forall \kappa \in \mathcal{K}$$

Here,  $\sigma_\kappa[\chi - 1]$ ,  $\sigma_\kappa[\chi - 2]$  and  $\sigma_\kappa[\chi - 7]$  denote the previous standard deviation values  $\sigma_\kappa$  on yesterday, the day before yesterday and the same day last week, respectively. The coefficients for the weighted average power prediction filter  $k_1$ ,  $k_2$  and  $k_7$  are selected from [14].

#### B. The Algorithm

We compare the renewable energy generation with the amount of released workload. If the renewable generation exceeds the amount of released workload, then we increase the amount of workload execution. Otherwise, we defer some of the workload and execute workload proportional to the renewable generation without violating constraints (C1) and (C2). To determine the amount of workload deferral and power usage, we use ‘ $\delta$ -delayed workload’ to compare with the renewable generation. Thus the online algorithm namely Follow Renewable with Deferral (FRD( $\delta$ )) looks ahead  $\delta$  time slots to determine the amount of execution. Let  $l_t^\delta$  be the  $\delta$ -delayed curve with delay of  $\delta$  slots for  $0 < \delta < D$ .

$$l_t^\delta = \begin{cases} 0 & \text{if } t \leq \delta, \\ L_{t-\delta} & \text{otherwise.} \end{cases}$$

We call the deadline curve as  $D$ -delayed curve and represent it by  $l_t^D$ . We determine the amount of workload execution and power usage by controlling the set of feasible choices for  $p_t$  in the optimization. For this purpose, we use the  $\delta$ -delayed curve to compare the released workload with the renewable generation. By having a lower bound on  $p_t$  for the low power and an upper bound for the high power, we maximize execution when there is renewable energy surplus and maximize deferral when there is shortage of renewable energy. Thus we control the amount of workload deferral and power storage simultaneously. Hence in the online algorithm, we shape the workload through two optimizations: Deficit Optimization and Surplus Optimization. The Deficit Optimization is used to maximize the deferral and Surplus Optimization is used to maximize the execution.

1) *Deficit Optimization*: The deficit optimization is applied over future  $D$  slots to find the optimum power usage for current slot by executing no more than  $\delta$ -delayed workload. Let  $t$  be the current time slot. At this slot, we apply a slightly modified version of offline optimization (1) in the interval  $[t, t + D]$ . We apply the following optimization DOPT( $l_t, l_t^\delta, R_t, S$ ) to determine  $p_t$  in order to minimize the renewable energy usage and hence maximize the deferral over  $D$  consecutive slots. We restrict the amount of execution to be no more than the  $\delta$ -delayed workload while satisfying the deadline

constraint (C1).

$$\begin{aligned}
\min_{p_t} \quad & \sum_{j=t}^{t+D} C(p_j) + \beta(|R_t - p_t| + \sum_{j=t+1}^{t+D} |\tilde{R}_j - p_j|) \quad (2) \\
\text{subj to} \quad & \sum_{j=1}^t l_j^D \leq \sum_{j=1}^t p_j \\
& \sum_{j=1}^{t+D} p_j = \sum_{j=1}^t l_j^\delta \\
& 0 \leq \sum_{j=1}^t R_j - \sum_{j=1}^t p_j \leq S
\end{aligned}$$

After solving the deficit optimization, we get the value of  $p_t$  for the current time slot and assign  $x_t = p_t$ . For the next time slot  $t+1$  we solve the deficit optimization again to find the values for  $x_{t+1}$  and  $p_{t+1}$ . Note that the deadline constraint (C1) and the release constraint (C2) are satisfied at time  $t$ , since from the formulation  $\sum_{j=1}^t l_j^D \leq \sum_{j=1}^t p_j \leq \sum_{j=1}^t l_j^\delta \leq \sum_{j=1}^t L_j$ .

2) *Surplus Optimization*: The *surplus optimization* is applied over future  $D$  slots to find the optimum power usage for current slot by executing more than  $\delta$ -delayed workload. Let  $t$  be the current time slot. At this slot, we apply the following optimization  $\text{SOPT}(l_t, L_t, R_t, S)$  to determine  $p_t$  in order to maximize the renewable energy usage and hence minimize the deferral over  $D$  consecutive slots. We maximize the amount of execution to be no more than the released workload  $L_t$  while satisfying the release constraint (C2).

$$\begin{aligned}
\min_{p_t} \quad & \sum_{j=t}^{t+D} C(p_j) + \beta(|R_t - p_t| + \sum_{j=t+1}^{t+D} |\tilde{R}_j - p_j|) \quad (3) \\
\text{subj to} \quad & \sum_{j=1}^t l_j^D \leq \sum_{j=1}^t p_j \\
& \sum_{j=1}^{t+D} p_j = \sum_{j=1}^t L_j \\
& 0 \leq \sum_{j=1}^t R_j - \sum_{j=1}^t p_j \leq S
\end{aligned}$$

To determine which optimization to apply at each time  $t$ , we take the intersection of  $\delta$ -delayed curve  $l^\delta$  with the renewable curve  $R$ . If  $\tilde{R}_{t+1} \geq l_{t+1}^\delta$ , we apply the surplus optimization to maximize renewable energy usage. Otherwise, we apply the deficit optimization to maximize deferral. This way we ensure feasibility of the solution from the optimization as proved in Theorem 1.

Note that the deadline constraint (C1) and the release constraint (C2) are satisfied at time  $t$ , since  $\sum_{j=1}^t l_j^D \leq \sum_{j=1}^t p_j \leq \sum_{j=1}^t l_j^\delta \leq \sum_{j=1}^t L_j$ . Algorithm 1 summarizes the steps for  $\text{FRD}(\delta)$ . For each new time slot  $t$ , Algorithm 1 predicts the renewable power in the time slots  $[t+1, t+D]$  and checks whether the curves  $l_t^\delta$  and  $R_t$  intersect. If there is a surplus of renewable energy, Algorithm 1 applies surplus optimization (SOPT); Deficit optimization (DOPT), otherwise. Figure 1(b) illustrates the nature of solutions from  $\text{FRD}(\delta)$  for  $x_t$  and  $p_t$ . Note that  $\delta$  is a parameter for the online algorithm  $\text{FRD}(\delta)$ .

---

### Algorithm 1 $\text{FRD}(\delta)$

---

```

1: surplus  $\leftarrow$  0;  $p_0 \leftarrow$  0
2:  $l^D[1 : D] \leftarrow$  0;  $l^\delta[1 : \delta] \leftarrow$  0
3: for each new time slot  $t$  do
4:    $l^D[t + D] \leftarrow L[t]$ 
5:    $l^\delta[t + \delta] \leftarrow L[t]$ 
6:   if  $l^\delta$  intersects  $R$  then
7:      $\tilde{R}[t + 1 : t + D] \leftarrow \text{Predict}(R[1 : t], \mu_t, \sigma_t)$ 
8:     if  $l^\delta[t + 1] \leq \tilde{R}[t + 1]$  then
9:       surplus  $\leftarrow$  1
10:    else
11:      surplus  $\leftarrow$  0
12:    end if
13:  end if
14:  if surplus = 0 then
15:     $p[t : t + D] \leftarrow \text{DOPT}(l[1 : t], l^\delta[1 : t], R[1 : t], \tilde{R}[t + 1 : t + D], S)$ 
16:  else
17:     $p[t : t + D] \leftarrow \text{SOPT}(l[1 : t], L[1 : t], R[1 : t], \tilde{R}[t + 1 : t + D], S)$ 
18:  end if
19:   $x_t \leftarrow p_t$ 
20: end for

```

---

### C. Analysis of the Algorithm

We first prove the feasibility of the solutions from the  $\text{FRD}(\delta)$  algorithm and then analyze the competitive ratio of this algorithm with respect to the offline formulation (1). First, we have the following theorem about the feasibility.

**Theorem 1:** The  $\text{FRD}(\delta)$  algorithm gives feasible solution for any  $0 < \delta < D$ .

*Proof:* We prove this theorem inductively by showing that the choice of any feasible  $p_t$  from an optimization applied in the interval  $[t, t+D]$  do not result in infeasibility in the optimization applied in the next time slot  $[t+1, t+D+1]$ . Initially, the optimization in  $\text{FRD}(\delta)$  is applied for the interval  $[1, D+1]$  with  $\sum_{j=1}^k l_j^D = 0$  for  $1 \leq k \leq D$ . Hence the optimization applied in the intervals  $[1, D+1]$  gives feasible  $p_1$  because  $\sum_{j=1}^k l_j^D \leq \sum_{j=1}^k l_j^\delta \leq \sum_{j=1}^k L_j$  for  $1 \leq k \leq D$ .

Now suppose the  $\text{FRD}(\delta)$  gives feasible  $p_t$  in an interval  $[t, t+D]$ . We have to prove that there exists feasible choice for  $p_t$  for the optimization applied at  $[t+1, t+D+1]$ . The deadline constraint (C1) and the release constraint (C2) are satisfied for  $p_t$ . Hence,  $\sum_{j=1}^t l_j^D \leq \sum_{j=1}^t l_j^\delta \leq \sum_{j=1}^t L_j$ . Since  $0 < \delta < D$ ,  $\sum_{j=1}^t l_j^D \leq \sum_{j=1}^{t+1} l_j^D \leq \sum_{j=1}^{t+1} l_j^\delta \leq \sum_{j=1}^{t+1} L_j \leq \sum_{j=1}^t L_j$ . Thus for any feasible choice of  $p_t$ , we can always obtain feasible solution for  $p_{t+1}$  such that the above inequality holds. ■

We now analyze the competitive ratio of the online algorithm with respect to the offline formulation (1). We denote the execution cost of the solution vectors  $X = (x_1, x_2, \dots, x_T)$  and  $P = (p_1, p_2, \dots, p_T)$  by  $\text{cost}_e(X, P) = \sum_{t=1}^T C(x_t)$ , power wastage cost by  $\text{cost}_w(X, P) = \beta \sum_{t=1}^T |R_t - p_t|$  and total cost by  $\text{cost}(X, P) = \text{cost}_e(X, P) + \text{cost}_w(X, P)$ . We have the following lemma.

**Lemma 2:**  $\text{cost}_w(X, P) \leq 2\beta \sum_{t=1}^T R_t$

*Proof:* Power wastage cost at time  $t$  is  $W_t = \beta |R_t - p_t| \leq \beta(R_t + p_t)$ , since  $p_t \geq 0$ . Then  $\text{cost}_w(X, P) \leq \beta \cdot \sum_{t=1}^T (R_t + p_t) \leq 2\beta \sum_{t=1}^T R_t$  since  $\sum_{t=1}^T p_t \leq \sum_{t=1}^T R_t$ . ■

Let  $X^*$  and  $P^*$  be the offline solution vectors from optimization (1). The following theorem proves that the competitive ratio of the  $\text{FRD}(\delta)$  algorithm is bounded by a constant with respect to the offline formulation (1).

**Theorem 3:**  $cost(X, P) \leq \frac{e_0 + e_1 + 2\beta}{e_1} cost(X^*, P^*)$ .

*Proof:* Since the offline optimization assigns all the workload in the  $[1, T]$  interval,  $\sum_{t=1}^T x_t^* = \sum_{t=1}^T L_t \leq \sum_{t=1}^T p_t^*$ , where we used  $x_t^* \leq p_t^*$  for all  $t$ . Hence  $cost(X^*, P^*) \geq cost_e(X^*, P^*) = \sum_{t=1}^T C(x_t^*) = \sum_{t=1}^T (e_0 + e_1 x_t^*) \geq \sum_{t=1}^T e_1 L_t$ . Now  $\sum_{t=1}^T R_t \leq \sum_{t=1}^T L_t$  because otherwise we could use a large battery to store all the renewable energy and execute the workload using that stored energy. Hence  $cost(X^*, P^*) \geq \sum_{t=1}^T e_1 L_t \geq \sum_{t=1}^T e_1 R_t$

In the online algorithm, we set  $x_t = p_t$  and  $\sum_{j=1}^t p_j \leq \sum_{j=1}^t L_j$  for all  $t \in [1, T]$ . Hence by lemma 2, we have  $cost(X, P) = cost_e(X, P) + cost_w(X, P) \leq \sum_{t=1}^T (e_0 + e_1 p_t) + 2\beta \sum_{t=1}^T p_t \leq \sum_{t=1}^T (e_0 + e_1 R_t) + 2\beta \sum_{t=1}^T R_t \leq (e_0 + e_1 + 2\beta) \sum_{t=1}^T R_t$ . ■

#### IV. FOLLOW RENEWABLE WITH NON-UNIFORM DEFERRAL

We now consider the general case where deadline requirements are not same for all the jobs in a workload. Let  $\nu$  be the maximum possible deadline. We decompose the workload according to their associated deadline. Suppose  $L_{d,t} \geq 0$  be the portion of the workload released at time  $t$  and has deadline  $d$ ,  $0 \leq d \leq \nu$ . We have  $\sum_{d=0}^{\nu} L_{d,t} = L_t$ . The workload to be executed at any time slot  $t$  can come from different previous slots  $t-d$  where  $0 \leq d \leq \nu$  as illustrated in Figure 2(a). Hence we redefine the deadline curve  $l_t$  and represent it by  $l'_t$ . Assuming  $L_{d,t} = 0$  if  $t \leq 0$ , we define  $l'_t = \sum_{d=0}^{\nu} L_{d,(t-d)}$ . Then the offline formulation remains the same as formulation (1) with the deadline curve  $l_t$  replaced by  $l'_t$ .

$$\begin{aligned} \min_{x_t, p_t} \quad & \sum_{t=1}^T C(x_t) + \beta \sum_{t=1}^T |R_t - p_t| \quad (4) \\ \text{subj. to} \quad & \sum_{j=1}^t l'_j \leq \sum_{j=1}^t x_j \leq \sum_{j=1}^t L_j \quad \forall t \\ & \sum_{j=1}^T x_j = \sum_{j=1}^T L_j \\ & 0 \leq \sum_{j=1}^t R_j - \sum_{j=1}^t p_j \leq S \quad \forall t \\ & 0 \leq x_t \leq p_t \quad \forall t \end{aligned}$$

We now consider the online case. At each time, we need to determine the optimum assignment and power usage that reduces the power wastage cost from the original workload while satisfying each individual deadline. We can apply the FRD( $\delta$ ) algorithm from the previous section with  $D = D_{min}$  where  $D_{min}$  is the minimum deadline for the workload. But often  $D_{min}$  is small ( $\sim 0$ ), FRD( $\delta$ ) does not work well because  $\delta < D_{min}$  becomes too small to detect the intersection of curves  $l^\delta$  and  $R$ . Hence we use a novel approach for distributing the workload  $L_t$  over the  $D_t$  slots such that the difference between renewable generation and power usage in adjacent time slots is minimal (see Figure 2(c)). We call this algorithm as Follow Renewable with Non-uniform Deferral (FRND) algorithm.

In the FRND algorithm, we apply optimization to determine  $p_t$  at each time slot  $t$  and make  $x_t = p_t$ . The optimization is applied over the interval  $[t, t + \nu]$  since at time slot  $t$  we can have workload that has deadline up to  $t + \nu$  slots. Hence at each time  $t$ , the released workload is a vector

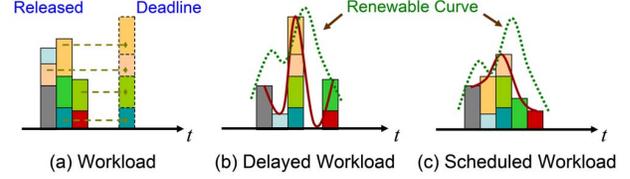


Fig. 2. Illustration of workload with different deadline requirements. (a) workload released at different times have different deadlines, (b) the delayed workload  $l'_t$ , may not follow the renewable curve due to large variation, (c) distribution of workload in adjacent slots by FRND to reduce the difference between execution curve and renewable curve.

of  $\nu + 1$  dimension. Let,  $\mathbf{L}_t = (L_{0,t}, L_{1,t}, \dots, L_{\nu,t})$  where  $L_{d,t} = 0$  if there is no workload with deadline  $d$  at time  $t$ . Let  $\mathbf{y}_t$  be the vector of unassigned workload released up to time  $t$ . The vector  $\mathbf{y}_t$  is updated from  $\mathbf{y}_{t-1}$  at each time slot by adding the newly released workload  $\mathbf{L}_t$ . Then  $\mathbf{y}_t = \mathbf{L}_t + (y_{1,t-1}, y_{2,t-1}, \dots, y_{\nu,t-1}, 0)$  where  $\mathbf{y}_t = (0, 0, \dots, 0)$  if  $t \leq 0$ . Then the optimization FRND-OPT( $\mathbf{y}_t, R_t, S$ ) applied at each  $t$  over the interval  $[t, t + \nu]$  is as follows:

$$\min_{p_t} \sum_{j=t}^{t+\nu} C(p_j) + \beta (|R_t - p_t| + \sum_{j=t+1}^{t+\nu} |\tilde{R}_j - p_j|) \quad (5a)$$

$$\text{subject to} \quad \sum_{j=0}^{\nu} p_{t+j} = \sum_{j=0}^{\nu} y_{j,t} \quad (5b)$$

$$\sum_{k=0}^j p_{t+k} \geq \sum_{k=0}^j y_{k,t} \quad 0 \leq j \leq \nu - 1 \quad (5c)$$

$$0 \leq \sum_{j=1}^t R_j - \sum_{j=1}^t p_j \leq S \quad (5d)$$

Note that the optimization (5) solves for  $\nu + 1$  values. We only use  $p_t$  as the power usage and assignment of workload at time  $t$  and use the other values to update the  $\mathbf{y}_{t+1}$  vector for the next time slot. Algorithm 2 summarizes the procedures for FRND. The FRND algorithm gives feasible solutions because it works with the unassigned workload and constraint (5c) ensures deadline constraint (C1) and constraint (5b) ensures the release constraint (C2). The competitive ratio for the FRND algorithm is same as the competitive ratio for FRD( $\delta$ ) because in FRND,  $p_t = x_t$  and release constraint (C2) holds at every  $t$  making  $\sum_{t=1}^T p_t = \sum_{t=1}^T x_t \leq \sum_{t=1}^T L_t$ .

#### Algorithm 2 FRND

```

1:  $y[0 : \nu] \leftarrow 0$ 
2:  $p_0 \leftarrow 0$ 
3: for each new time slot  $t$  do
4:    $y[0 : \nu] = \{y[1 : \nu], 0\} + L_t[0 : \nu]$ 
5:    $\tilde{R}[t+1 : t+D] \leftarrow \text{Predict}(R[1 : t], \mu_t, \sigma_t)$ 
6:    $p[t : t+\nu] \leftarrow \text{FRND-OPT}(y[0 : \nu], \tilde{R}[t+1 : t+D], S)$ 
7:    $x_t \leftarrow p_t$ 
8: end for

```

#### V. SIMULATION

In this section, we evaluate the cost incurred by the FRD( $\delta$ ) and FRND algorithms relative to the optimal solution in the context of workload generated from realistic data and real renewable power generation data.

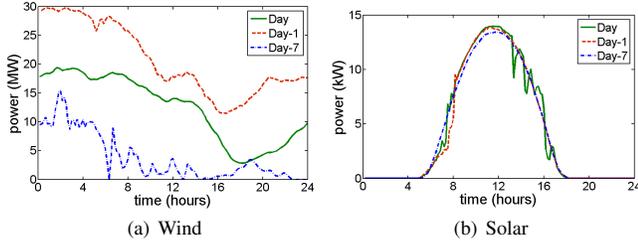


Fig. 3. Illustration of the renewable power (a) wind traces on 01 August 2006, 31 July 2006, 25 July 2006 (b) solar traces on 01 August 2012, 31 July 2012, 25 July 2012 used in the experiments.

### A. Simulation Setup

We use realistic parameters in the simulation setup and provide conservative estimates of energy savings resulting from our proposed FRD( $\delta$ ) and FRND algorithms.

*Cost benchmark:* A common approach for power usage in data centers is to follow the workload curve [18]. In this policy, the amount of power usage at each time is determined by the amount of released workload. Such an approach is naïve and does not take into account the flexibilities of workload deferral and availability of renewable energy. Yet this is a conservative estimate as it meets all the deadlines. For the follow the workload algorithm, we assume the data center has enough energy storage (or brown energy) to meet workload demand during periods of low renewable energy. The cost of that deficit energy demand is captured in the energy wastage parameter  $\beta$ . We compare the total energy cost from the FRD( $\delta$ ) and FRND algorithms with the ‘follow the workload’ ( $x = p = L$ ) strategy and evaluate the energy reduction.

*Wind traces:* The wind power generation data over time is taken from the publicly available western wind dataset from National Renewable Energy Laboratory (NREL) website [15]. Figure 3(a) shows the wind power generated over time in 10 minutes granularity for 24 hours on 01 August, 2006. It is the latest dataset available publicly and has been used in earlier comparisons [21]. At each time slot, we predict the future wind power from the traces on 31 July 2006 and 25 July 2006. For the experiments, we normalize the power data with the workload to capture the variation in the wind power to align with the workload variation.

*Solar traces:* We use the solar power generation data from the PV panels at UC San Diego campus. Figure 3(b) shows the variation in the solar power traces over 24 hours on 01 August 2012. At each time slot, we predict the future solar power from the traces on 31 July 2012 and 25 July 2012. At night, there is no solar power. So for the simulation, we use the data from 8 am to 4 pm to capture the variability and schedule the workload that are released within that time frame. Note that we do not use the solar thermal generation as it requires significant infrastructure for a solar thermal plant. Since solar thermal plants typically incorporate a day’s thermal storage [16], we cannot apply variability mitigation techniques via workload deferral. Similar to wind traces, we normalize the solar data to match the workload.

*Cost function parameters:* The total cost is characterized by  $e_0$  and  $e_1$  for the execution cost and  $\beta$  for the energy wastage cost. In the execution cost,  $e_0$  represents the proportion of the fixed cost and  $e_1$  represents the load dependent energy consumption. Modern servers are getting increasingly energy proportional and the energy consumption is dominated by the load dependent component [17]. Therefore  $e_0$  is negligible compared to  $e_1$  and we choose  $e_0 = 0$  and  $e_1 = 1$ . The energy wastage cost parameter  $\beta$  represents the penalty for not using renewable energy for executing the workload. This cost

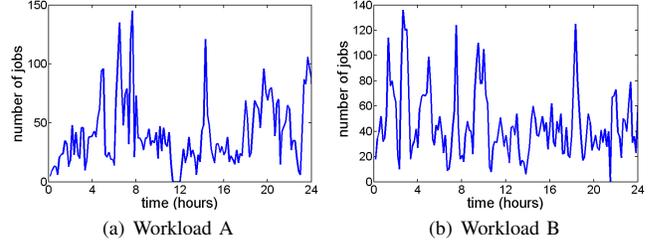


Fig. 4. Illustration of the two MapReduce traces as dynamic workload used in the experiments.

is analogous to using stored energy from battery (or carbon intensive energy) instead of using renewable energy. This cost comes from the energy loss in batteries, chemical and harmful effect on environment, internal resistance for charging and discharging etc. We choose a penalty parameter  $\beta = 10$  for using storage energy instead of renewable energy. We also examine the sensitivity of  $\beta$  on the solution by varying it from 1-10. Since we are penalizing the stored energy usage, we assume that data center has enough storage capacity to supply stored (brown) energy when needed. Hence we use a large value for  $S$ , storage capacity in the simulation.

*Workload description:* We use two publicly available MapReduce traces as examples of dynamic workload. The MapReduce traces were released by Chen et. al. [19] which are produced from real Facebook traces for one day (24 hours) from a cluster of 600 machines. We count the number of different types of job submissions over a time slot length of 10 minutes and use that as a dynamic workload (Figure 4) for simulation. The two samples we use represent strong diurnal properties and have variation from typical workload (Workload A) to bursty workload (Workload B).

*Deadline assignment:* For FRD( $\delta$ ), the deadline  $D$  is uniform and is assigned in terms of number of slots the workload can be delayed. For our simulation, We vary  $D$  from 1 – 6 slots which gives latency from 10 minutes upto 1 hour. This is realistic as deadlines of 8-30 minutes for MapReduce workload have been used in the literature [7], [20]. For FRND, we use k-means clustering to classify the workload into 10 groups based on the map, shuffle and reduce bytes. The characteristics of each group are depicted in Table I. From Table I, it is evident that smaller jobs dominate the workload mix which justifies the tight response times for the jobs. For each class of jobs we assign a deadline from 1 – 10 slots such that smaller class (i.e. larger jobs) has larger deadline and larger class of jobs has smaller deadline.

### B. Analysis of the Simulation

We now analyze the impact of different parameters on cost savings provided by FRD( $\delta$ ) and FRND. We then compare FRD( $\delta$ ) and FRND for uniform deadline (FRND-U) for both solar and wind power traces.

*Impact of deadline:* The first parameter we study is the impact of different deadline requirements of the workload on the non-renewable energy savings. Figure 5 shows that even for deadline  $D$  as small as 2 slots, the energy cost is reduced by  $\sim 20\%$  for FRND-U,  $\sim 10\%$  for FRD( $\delta$ ) while the offline algorithm gives a cost saving of  $\sim 40\%$  compared to the ‘follow the workload’ algorithm. It also shows that for all the algorithms, large  $D$  gives more cost savings as more workload can be delayed to reduce the difference between renewable curve and workload curve. As  $D$  increases the energy cost reduction from FRND-U and FRD( $\delta$ ) approaches offline cost saving which is as much as 40%.

TABLE I  
CLUSTER SIZES AND DEADLINES FOR WORKLOAD CLASSIFICATION FOR FRND

Cluster	Workload A					Workload B					Deadline (#slots)
	#Jobs	%Jobs	Map(MB)	Shuffle(MB)	Reduce(MB)	#Jobs	%Jobs	Map(MB)	Shuffle(MB)	Reduce(MB)	
1	5691	96.56	0.02	0.00	0.67	6313	95.10	0.02	0.00	0.48	1
2	116	1.97	44856.77	15493.69	83.89	223	3.36	39356.46	6594.93	99.26	2
3	27	0.46	57121.85	148012.87	16090.40	41	0.62	110076.24	282.08	1.60	3
4	23	0.39	125953.59	0.00	51.89	25	0.38	379363.01	0.00	521.45	4
5	19	0.32	0.33	0.00	49045.29	16	0.24	0.04	0.00	40355.53	5
6	8	0.14	207984.10	414045.45	3095.56	7	0.11	132529.27	383548.19	31344.38	6
7	5	0.08	541522.77	0.00	0.05	4	0.06	258152.65	1020741.05	22631.52	7
8	3	0.05	0.05	0.00	203880.59	3	0.05	0.29	0.00	311410.40	8
9	1	0.02	7201446.27	48674.26	0.10	3	0.05	1182734.09	3.93	0.01	9
10	1	0.02	934594.27	8413335.44	0.06	3	0.05	0.56	0.00	622103.12	10

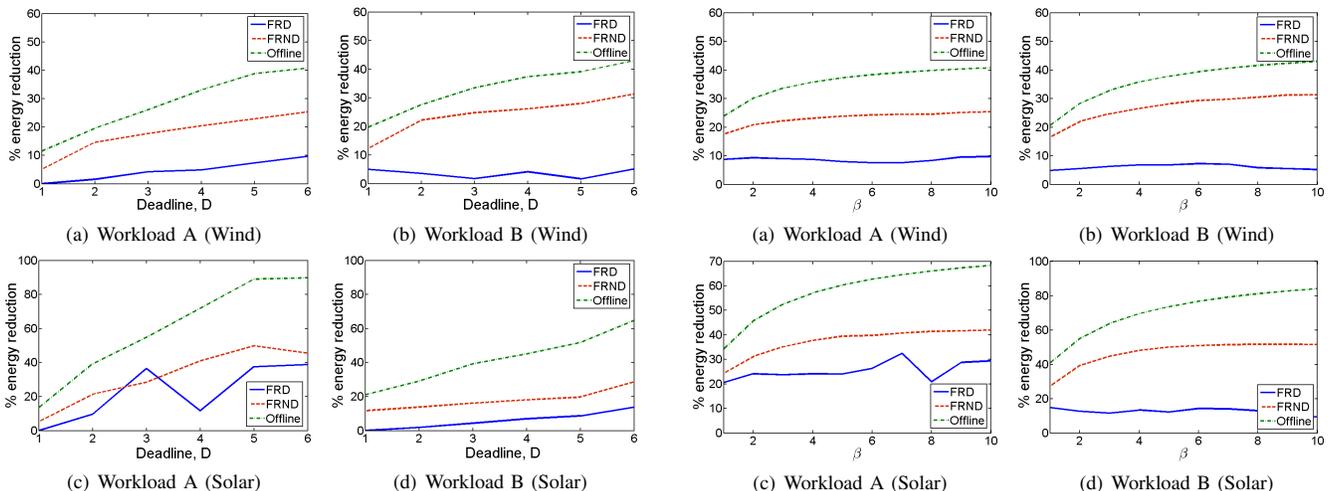


Fig. 5. Impact of deadline on energy cost reduction by Offline, FRND-U and FRD( $\delta$ ) with  $\delta = D/2$  with respect to the ‘follow the workload’ policy with wind (a,b) and solar (c,d) traces.

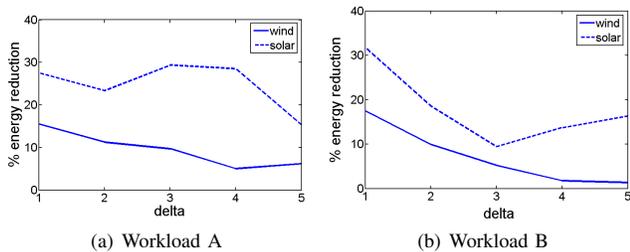


Fig. 6. Impact of  $\delta$  for FRD( $\delta$ ) with deadline  $D = 6$ .

*Impact of  $\delta$  for FRD( $\delta$ ):* The parameter  $\delta$  is used as a lookahead to detect when the renewable curve intersects the workload curve and thus controls the amount of execution in the FRD( $\delta$ ) algorithm. If  $\delta$  is large, then we may have delayed more workload than needed and may need to execute more than the available renewable energy. On the other hand if  $\delta$  is small, we may not have enough workload to execute to follow the renewable energy. Although the cost savings from FRD( $\delta$ ) largely depends on the nature of the workload curve, Figure 6 shows that  $\delta \sim D/2$  is a conservative estimate for better cost savings.

*Performance of FRND:* We evaluated the energy cost savings from FRND by assigning different deadlines to different jobs by classifying the workload as shown in Table I. For conservative estimates of deadline requirements (1-10), we found 10.26% non-renewable energy reduction for Workload A and 10.90% non-renewable energy reduction for Workload B for wind traces. For solar traces, FRND gives 16.4% non-

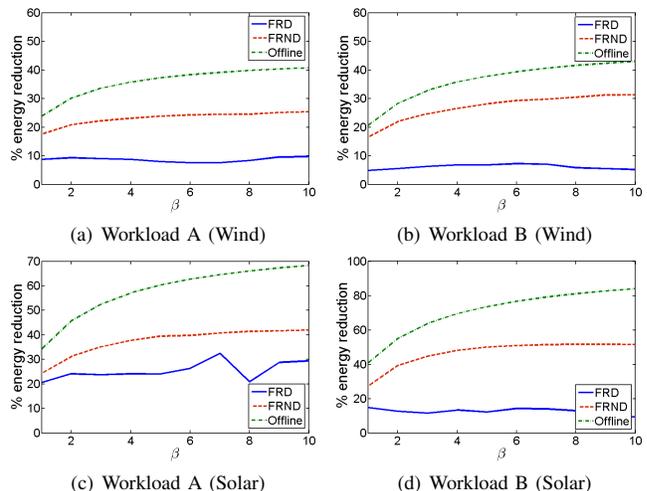


Fig. 7. Impact of  $\beta$  on energy reduction by Offline, FRND-U and FRD( $\delta$ ) with  $\delta = D/2$  with respect to the ‘follow the workload’ policy with wind (a,b) and solar (c,d) traces with  $D = 6$ .

renewable energy reduction for Workload A and 24.53% non-renewable energy reduction for Workload B.

*Comparison of FRD( $\delta$ ) and FRND:* We compare FRND for uniform deadline (FRND-U) with FRD( $\delta$ ) for  $\delta = D/2$ . Figure 5 illustrates the non-renewable energy reduction for FRD( $\delta$ ) and FRND-U with different deadlines  $D = 1 - 6$ . For both the workload and renewable traces, FRND-U performs better than FRD( $\delta$ ). However for some workload (e.g. Figure 5(c)), looking ahead using  $\delta$  as in FRD( $\delta$ ) can be more beneficial than distributing workload for  $D$  consecutive slots as in FRND. Hence, generally (but not always) FRND produces better results compared to FRD. However, FRND does require a greater specification of workload demands which may not always be available for all jobs. Both the algorithms are based on linear program and on a CVX solver they take around 10-12 milliseconds to compute schedule at each step.

*Impact of  $\beta$  on energy reduction:* We now measure the sensitivity of power wastage parameter  $\beta$  on our algorithms. Since we do not know the exact value of the cost of renewable energy wastage with respect to the execution cost, we vary the parameter  $\beta$  from 1 to 10 with a fixed deadline  $D = 6$ . Figure 7 shows the impact of  $\beta$  on the non-renewable energy reduction for different algorithms with wind and solar power. From these figures, we find that non-renewable energy reduction increases with the increase in the penalty parameter  $\beta$ .

## VI. RELATED WORK

Greening data centers is becoming an increasingly important topic in operating cloud-scale data centers for two main reasons: (1) the global energy crisis and environmental

concerns (e.g. global warming) [21] and (2) increasing energy consumption in data centers [1]. With the importance of energy management in data centers, many scholars have applied energy-aware scheduling because of its low cost and practical applicability. Beloglazov et. al. [22] gave the taxonomy and survey on energy management in data centers. Most of the prior work on dynamic power management via workload scheduling either use models based on queueing theory [6], or control theory [23]. Power capping and power budgeting techniques have also been proposed which only shave the peak power generation. Gmach et. al. [8] use wind and solar energy to cap the power usage in a data center. They address the problem of variability in using renewable energy and employ extra energy storage to overcome this problem. But this only shaves the peak power and does not shape the workload curve according to renewable power generation. Liu et. al. [10] studied demand response in the context of data centers considering local generation of power and devised workload shifting techniques to avoid coincident peak of workload curve with peak demand hours. Urgaonkar et. al. [24] proposed the use of energy storage to shift peak demand away from high peak periods. In contrast to these works, our algorithms not only reduce the peak but also shape the other portions of the workload curve according to the renewable generation by exploiting the flexibility of energy storage.

There has been significant recent work on renewable energy integration for power management in data centers. Le et. al. [9] model an optimization problem which uses the market prices of brown and green energy to decide how much energy of each type should be bought in each interval. But they do not make server level scheduling decisions based on the amount of green energy. Stewart and Shen [3] analyze the energy requirement distributions of different requests and how to integrate green energy to the system. They identified the problem of variability in green energy but do not propose solutions. Krioukov et. al. [18] use renewable energy for execution of MapReduce jobs. They schedule the tasks with the availability of green energy and terminate them when the amount of green energy is not sufficient in the subsequent intervals. But none of these works provide guarantee on latency requirements for the completion of the jobs. In contrast, we formulate our algorithms such that none of the jobs violate their respective deadlines while leveraging flexibilities of power storage and workload deferral.

Many applications in real world require delay bound or deadline constraint e.g. see Lee et. al. [26]. In the context of energy conservation, deadline is a critical adjusting tool between performance loss and energy consumption. Energy efficient deadline scheduling was first studied by Yao et al. [25]. They proposed algorithms, which aim to minimize non-renewable energy consumption for independent jobs with deadline constraints. Most prior work on energy management in data centers, merely consider minimizing the average delay without any bound on the delay. Recently, Mukherjee et. al. [5] proposed online algorithms considering deadline constraints to minimize the computation, cooling and migration energy for machines. Goiri et. al. [27] utilize availability of green energy and deadline information to schedule jobs in a data center. The algorithm does not put any bound on the latency requirements of the jobs and some of the jobs may violate their deadline if many jobs are delayed to be executed later.

## VII. CONCLUSION

We have shown that significant reduction in non-renewable energy consumption can be achieved by joint optimization on dynamic deferral of workload and storage for renewable energy inside data centers. We have proposed two new algorithms, FRD( $\delta$ ) and FRND, for power usage and scheduling

workload while guaranteeing the deadlines. The algorithms use the flexibility in the latency requirements of the workload for energy savings and guarantee bounded cost and bounded latency under very general settings - arbitrary workload, general deadline and general energy cost models. Further, our experiments show that the proposed algorithms are simple to implement and do not impose significant computational overhead. Additionally, the algorithms have constant competitive ratios and offer noteworthy energy savings as proved by theory and validated by simulation.

## ACKNOWLEDGMENT

This work was sponsored in part by the California Energy Commission (CEC) and NSF Variability Expedition.

## REFERENCES

- [1] *Server and Data Center Energy Efficiency*, Final Report to Congress, U.S. Environmental Protection Agency, 2007.
- [2] Z. Liu, M. Lin, A. Wierman, S. Low, and L. Andrew, Greening Geographical Load Balancing, in Proc. ACM SIGMETRICS, 2011.
- [3] C. Stewart and K. Shen, *Some Joules Are More Precious Than Others: Managing Renewable Energy in the Datacenter*, in Proc. Power Aware Comput. and Sys., October 2009.
- [4] E. Pakbaznia and M. Pedram, *Minimizing data center cooling and server power costs*, in Proc. ISLPED, 2009.
- [5] T. Mukherjee, A. Banerjee, G. Varsamopoulos, and S. K. S. Gupta, *Spatio-Temporal Thermal-Aware Job Scheduling to Minimize Energy Consumption in Virtualized Heterogeneous Data Centers*, Computer Networks, 53(17), 2009.
- [6] M. Lin, A. Wierman, L. H. Andrew, E. Thereska, *Dynamic right-sizing for power-proportional data centers*, in Proc. IEEE INFOCOM, 2011.
- [7] A. Verma, L. Cherkasova, R. Campbell, *Resource Provisioning Framework for MapReduce Jobs with Performance Goals*, in Middleware, 2011.
- [8] D. Gmach, J. Rolia, C. Bash, Y. Chen, T. Christian, A. Shah, R. Sharma and Z. Wang, *Capacity Planning and Power Management to Exploit Sustainable Energy*, In Proc. International Conference on Network and Service Management (CNSM), 2010.
- [9] K. Le, R. Bianchini, T. D. Nguyen, O. Bilgir, M. Martonosi, *Capping the brown energy consumption of Internet services at low cost*, In Proc. of the International Conference on Green Computing (IGCC), 2010.
- [10] 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 Proc. of IFIP Performance, 2013.
- [11] M. A. Adnan, R. Sugihara, Y. Ma and R. Gupta, *Dynamic Deferral of Workload for Capacity Provisioning in Data Centers*, UCSD Technical Report, CoRR, abs/1109.3839, 2012.
- [12] *SPEC power data on SPEC website at <http://www.spec.org>*.
- [13] C. Holt, *Forecasting seasonals and trends by exponentially weighted moving averages*. International Journal of Forecasting, 2004.
- [14] A. H. Mohsenian-Rad and A. Leon-Garcia, *Optimal residential load control with price prediction in real-time electricity pricing environments*, IEEE Trans. Smart Grid, 1(2), pp. 120-133, Sep. 2010.
- [15] National Renewable Energy Laboratory. <http://www.nrel.gov/>
- [16] D. Mills, *Advances in solar thermal electricity technology*, Solar Energy, 76, pp. 19-31, 2004.
- [17] L. A. Barroso, and U. Hözl, *The case for energy-proportional computing*, IEEE Computer, 40(12), pp. 33-37, 2007.
- [18] A. Krioukov, C. Goebel, S. Asplough, Y. Chen, D. Culler, R. Katz, *Integrating Renewable Energy Using Data Analytics Systems: Challenges and Opportunities*, IEEE Data Engineering Bulletin. March 2011.
- [19] Y. Chen, A. Ganapathi, R. Griffith and R. Katz, *The Case for Evaluating MapReduce Performance Using Workload Suites*, in Proc. IEEE MAS-COTS, 2011.
- [20] K. Kc and K. Anyanwu, *Scheduling Hadoop Jobs to Meet Deadlines*, in Proc. IEEE CloudCom, 2010.
- [21] Z. Liu, M. Lin, A. Wierman, S. Low, and L. H. Andrew, *Geographical load balancing with renewables*, In Proc. GreenMetrics, June 2011.
- [22] A. Beloglazov, R. Buyya, Y. C. Lee, A. Zomaya, *A taxonomy and survey of energy-efficient data centers and cloud computing systems*, Advances in Computers, Elsevier: Amsterdam, 2011.
- [23] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautham, *Managing server energy and operational costs in hosting centers*, in Proc. ACM Sigmetrics, 2005.
- [24] R. Urgaonkar, B. Urgaonkar, M. Neely, and A. Sivasubramaniam, *Optimal power cost management using stored energy in data centers*, in Proc. of the ACM Sigmetrics, 2011.
- [25] F. Yao, A. Demers, and S. Shenker, *A scheduling model for reduced CPU energy*, in Proc IEEE FOCS, pp. 374-382, 1995.
- [26] C. B. Lee, and A. Snaveley, *Precise and realistic utility functions for user-centric performance analysis of schedulers*, in Proc. HPDC, 2007.
- [27] I. Goiri, Le, M. E. Haque, R. Beauchea, T. Nguyen, J. Guitart, J. Torres, R. Bianchini, *GreenSlot: Scheduling Energy Consumption in Green Datacenters*, in Proc. of Supercomputing, November 2011.