Peak Shaving Through Optimal Energy Storage Control for Data Centers

Mehiar Dabbagh , Ammar Rayes ‡, Bechir Hamdaoui and Mohsen Guizani†
Oregon State University, Corvallis, OR 97331, dabbaghm,hamdaoub@onid.orst.edu
‡ Cisco Systems, San Jose, CA 95134, ‡ rayes@cisco.com
† Qatar University, mguizani@ieee.org

***Abstract*—We propose efficient control strategies for deciding the amount of energy that a battery needs to charge/discharge over time with the objective of minimizing the Peak Charge and the Energy Charge components of the Data Center (DC) electricity bill. We consider first the case where the DC's power demands throughout the whole billing cycle are known and we present an optimal peak shaving control strategy for a battery that has certain leakage and conversion losses. We then relax this assumption and propose an efficient battery control strategy when we only know predictions of the DC's power demands in a short duration in the future. Several comparative studies are conducted based on real traces from a Google DC in order to validate the proposed techniques.**

***Index Terms*—energy efficiency, energy storage, peak shaving, data centers, convex optimization.**

I. Background

According to [1], large IT companies such as Google, Microsoft and Amazon spend millions of dollars per month to pay the electricity bills associated with their Data Centers (DCs). These electricity bills account for 30% to 50% of the total DCs operational expenses [2]. Thus, there is clearly a great monetary incentive to cut down those expenses.

The electricity bill that the DC receives from the grid company at the end of the billing cycle (e.g. month) is made up of two components [3]: i) *Energy Charge:* which is dependent on how much energy (measured in kilo Watt hour (kWh)) that the DC consumed throughout the whole billing cycle, and ii) *Peak Charge:* which is a penalty that is proportional to the maximum amount of power (measured in kilo Watt (kW)) that was drawn by the DC during the whole billing cycle. This penalty is very expensive and is enforced by the grid company to encourage the DC to balance its power demand and to discourage the spiky power usage. The maximum amount of power drawn by the DC is calculated in a time-slotted fashion where the grid company calculates the DC's average power usage during each slot of a certain length (e.g. 15-minutes), and the peak charge is calculated based on the slot with the maximum average power among all the billing cycle's slots.

The majority of the techniques that were proposed previously to cut down the electricity bill focused exclusively on minimizing the Energy Charge while completely ignoring the Peak Charge. Techniques such as energy-aware scheduling [4, 5], job migration [6] and resource over-booking [7] reduce the Energy Charge by consolidating the DC's workload on fewer number of ON servers which allows switching a larger number of servers to sleep to save energy. Our work complements those techniques and focuses on minimizing the Peak Charge component of the electricity bill.

Minimizing the Peak Charge is achieved through *peak shaving* techniques which are divided into two categories: i) *Workload Modulation:* where some of the DC's computing tasks are dropped [8] or delayed [9] during peak periods, and ii) *Energy Storage:* where extra power is drawn from the grid during low-demand periods and is stored in batteries so that it can be used later to reduce the amount of power drawn from the grid during peak periods.

The focus of our work is on shaving the peak using Energy Storage as this technique does not cause performance degradation unlike the Workload Modulation technique [8, 9]. Furthermore, DCs are already supplied with an Uninterrupted Power Supply (UPS) battery that can be used for peak shaving. Such UPS battery is typically used to power the DC until the diesel generator starts generating power when a power outage occurs. In fact, the UPS battery found in today's DCs can store enough energy to power the DC when operating at its maximum capacity for up to 30 minutes, while the transition time needed to run the diesel generator is less than 20 seconds [10]. This battery can thus always store an amount of energy enough to power the DC during the short transition period (to be used if a power outage occurs), while the remaining storage capacity can be utilized for peak shaving.

The main challenge with the Energy Storage peak shaving technique is to come up with a good control strategy that decides when to charge (discharge) energy and the amount of energy that needs to be charged (discharged). Two aspects make finding an efficient strategy further challenging:

- *Battery Energy Losses:* Batteries are not ideal devices in reality as they lose a certain percentage of their stored energy over time and these losses are called *leakage losses*. Also when routing a certain amount of energy to the battery, a certain percentage of the routed energy gets lost due to conversion operations and such losses are called *conversion losses*.
- *Workload Uncertainty:* The DC's workload is quite sporadic and the duration of the billing cycle is long (typically one month). This makes it hard to make optimal control decisions as it is hard to know the DC's future

This work was supported in part by Cisco and National Science Foundation NSF CAREER award CNS-0846044.

power demands throughout the whole billing cycle.

This paper proposes a battery control strategy that considers these two aspects. We start first by assuming that the DC's power demands throughout the whole billing cycle are known in advance (*full-horizon knowledge*), and we present an approach that finds the optimal control strategy for a battery with specific leakage and conversion losses. The proposed full-horizon approach provides an upper bound of how much monetary savings a certain type of battery can achieve and helps with selecting what type of battery the DC should be equipped with based on the DC's workload demands. Unlike the dynamic programming approach proposed in [11], our proposed full-horizon approach formulates a convex optimization problem that allows considering both types of losses (conversion and leakage) when calculating the optimal strategy.

We then consider the case where we only know the DC's power demand for a short duration in future (*limited-horizon knowledge*), where we propose an algorithm that uses this knowledge to decide at each time step how much energy the battery needs to charge or discharge while accounting for the battery's energy losses. The proposed limited-horizon control algorithm is compared against the well-known Threshold control strategy [12] where we show that knowing the future demands for a short duration of one hour allows our algorithm to reduce a significant portion of the electricity bill compared to the Threshold control strategy.

To sum everything up, our main contributions are the following. We:

- Propose an optimal energy storage control strategy for a lossy battery under full-horizon knowledge of the future DC's power demands.
- Propose an algorithm for making efficient energy storage control decisions that considers limited-horizon knowledge of the future DC's power demands while accounting for battery losses.
- Evaluate our proposals and conduct comparative studies using real traces from a Google DC.

The rest is organized as follows. Section II introduces Google DC traces and provides an experimental study that illustrates why one should worry about reducing the DC's Peak Charge. Section III explains the approach for finding the optimal full-horizon energy storage control strategy. Section IV describes the limited-horizon algorithm. Section V provides comparative evaluation of our proposed approaches. Finally, Section VI concludes and provides directions for future work.

II. The Case for Google DC

In order to illustrate the significance contribution of the Peak Charge component to the total electricity bill, we conduct an experiment where we rely on real workload traces [13] from a Google DC that is made up of around 12K servers to calculate how much power that DC consumes over time. Google traces report the tasks that clients submitted to one of Google DCs. Each task is assigned a linux container and utilizes an amount of CPU resources over time. In order to calculate Google

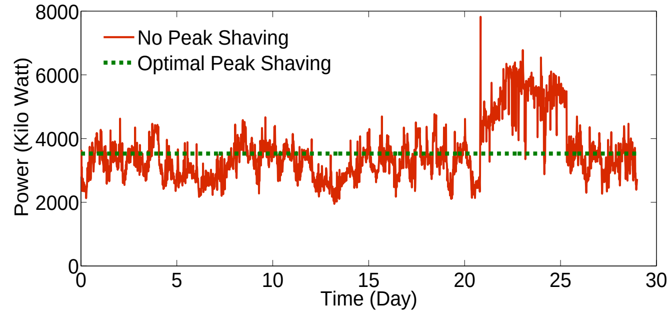

Fig. 1: The power drawn from the Grid by Google DC.

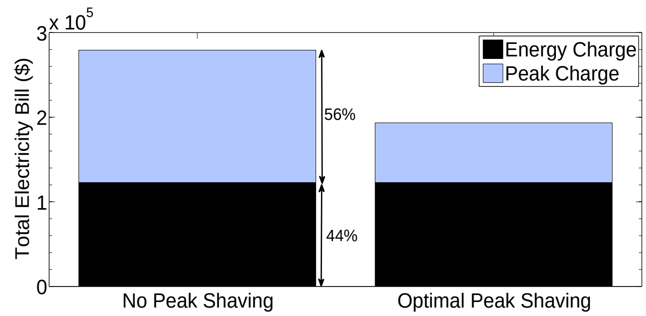

Fig. 2: Breakdown of Google DC total electricity bill (based on the power consumption in Fig. 1).

DC's power consumption, we parse the traces and track at each time slot how much CPU resources are being utilized by all the tasks that are currently hosted on the Google DC. We then calculate for each time slot what is the least number of servers needed to be kept ON to server those tasks as if state-of-the-art Energy Charge minimizing techniques [4, 6, 7] were applied to consolidate the DC's workload. The power consumption of each ON server is then calculated based on the power model in [14] where the server's consumed power, P_{on} increases linearly from P_{idle} to P_{peak} as the server's CPU utilization, ν, increases from 0 to 100%. More specifically, $P_{on}(\nu) = P_{idle} + \nu(P_{peak} - P_{idle})$, where $P_{peak} = 400$ and $P_{idle} = 200$ Watts. The rest of the DC servers that are not hosting any tasks don't consume any power as they are assumed to be switched off completely or put to highly power efficient sleep states to save energy. Google DC is assumed to have a Power Usage Efficiency (PUE) of 1.7, which is a typical value for DCs [15] and means that for every watt spent on IT power, an additional 0.7 watt is spent by non-computing infrastructure (e.g. cooling devices).

Fig. 1 plots the calculated power drawn by Google DC (referred to by No Peak Shaving) over the entire trace period (29 days). Fig. 1 also plots the power consumption of the Google DC when the same energy that was consumed by the DC during the 29-day period was spread evenly over the entire billing duration. This case is referred to by Optimal Peak Shaving as it represents the case where the DC had the same Energy Charge as the No Peak Shaving case but where the Peak Charge was minimal.

We then calculate the electricity bill for Google DC during the entire 29-day period[1] and using real power prices [16], where the price to calculate the Energy Charge is $\alpha = 0.05\,\$/kWh$, whereas the price to calculate the Peak Charge is $\beta = 20\,\$/kW$ and where the Peak Charge is calculated based on dividing the billing cycle into slots of length $\tau = 15$ minutes. We plot in Fig 2 the contribution of both the Energy Charge and the Peak Charge components to the total electricity bill based on the power consumption of No Peak Shaving and Optimal Peak Shaving cases that were shown in Fig. 1. Observe that for the No Peak Shaving case, the Peak Charge contributes to 56 % of the total electricity bill while the Energy Charge accounts for the remaining 44 %. Observe also that the Optimal Peak Shaving case reduces the Peak Charge paid by the DC during that month by $ 86K when compared with the No Peak Shaving case, which translates into a 31% reduction of the total electricity bill. These numbers highlight the high contribution of the Peak Charge to the total electricity bill and show the potentials for saving significant amount of money by applying peak shaving techniques such as the Energy Storage technique discussed in this paper.

III. Full-Horizon Optimal Control

A. Power Notations and Battery Model:

We consider a time-slotted model, where the whole billing cycle is divided into n slots and each slot has a duration of τ minutes. The index i is used to refer to one of the billing cycle's slots where $1 \leq i \leq n$ holds. For a slot i, the following notations are used:

- D_i: is the DC's power demands that must be met during the i^{th} slot.
- g_i: is the power taken from the grid to serve the DC's power demands during the i^{th} slot.
- c_i^-: is the power discharged from the battery to serve the DC's power demands during the i^{th} slot.
- c_i^+: is the power taken from the grid to charge the battery during the i^{th} slot.
- t_i: is the total power taken from the grid during the i^{th} slot.
- r_i: is the amount of power stored in the battery at the beginning of the i^{th} slot.

Where all the above mentioned power notations are measured in kilo Watt and the flow of these power notations is shown in Fig. 3.

The specs of the battery are summarized by the tuple $\Phi = (\eta_c, \eta_l, C^+_{max}, C^-_{max}, R_{max})$, where:

- η_c represents the battery's conversion efficiency and falls within the range $(0, 1]$, and means that only $\eta_c c_i^+$ percent out of the c_i^+ power that the battery draws from the grid ends up being stored in the battery, whereas the remaining $(1-\eta_c)c_i^+$ gets lost due to conversion operations.

[1]Google revealed workload traces for only 29 days and thus in our analysis we consider the length of the billing cycle to be 29 days rather than a month of 30 or 31 days.

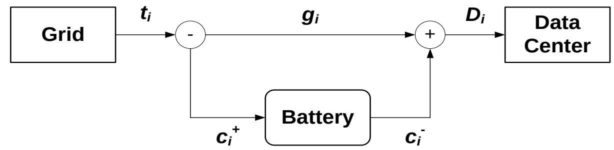

Fig. 3: Illustration of the power flow.

- η_l represents the battery's leakage efficiency and falls within the range $(0, 1]$ and means that the battery losses $(1-\eta_l)$ percent of its stored energy per one slot of time due to leakage losses.
- C^+_{max} and C^-_{max} represent the maximum amount of power that the battery can draw from the grid and that the battery can discharge to the data center respectively.
- R_{max} represents the battery's maximum storage capacity.

In addition to those specs that are summarized by Φ, the initial energy that is stored in the battery at the beginning of the first slot in the billing cycle is referred to by R_{init}.

B. Formulated Problem:

Given that we know the DC's power demands throughout the whole billing period referred to by $\vec{D} = \{D_1, D_2, ..., D_n\}$, the specs of the battery Φ, and the initial amount of energy that the battery holds R_{init}, we find the optimal control strategy with the minimal electricity bill by solving the following optimization problem.

Objective: We seek to minimize the total electricity bill which is made up of the Energy Charge and the Peak Charge and thus can be expressed as:

$$\text{Minimize} \quad \underbrace{\alpha\xi\sum_{i=1}^{n} t_i}_{\text{Energy Charge}} + \underbrace{\beta \max_i\{t_i\}}_{\text{Peak Charge}}$$

Where: α is the energy price measured in ($\$/kWh$), ξ is a constant that converts the total energy that the DC draws from the grid into kWh and is calculated as: $\xi = \tau/60$, and β is the peak price measured in ($\$/kW$).

Constraints: The optimization problem is solved subject to the following constraints. One,

$$t_i = g_i + c_i^+ \qquad ,1 \leq i \leq n \tag{C.1}$$

which states that for each time slot i the total power drawn from the grid is the aggregation of the power used to sever the DC's power demand and the power drawn to be stored in the battery. Two,

$$g_i + c_i^- = D_i \qquad ,1 \leq i \leq n \tag{C.2}$$

which states that the DC's power demand must be met by the power taken from the grid and the power discharged from the battery for each time slot i. Three,

$$r_i = \begin{cases} R_{init} & ,i = 1 \\ \eta_c c^+_{i-1} + \eta_l(r_{i-1} - c^-_{i-1}) & ,1 < i \leq n \end{cases} \tag{C.3}$$

Which calculates how much power will be stored in the battery at the beginning of each slot while accounting for the conversion and leakage losses. Four,

$$c_i^- \leq r_i \qquad ,1 \leq i \leq n \tag{C.4}$$

which states that the amount of discharged power at time slot i must not exceed the battery's stored energy. Five,

$$c_i^- \leq C_{max}^- \qquad ,1 \leq i \leq n \tag{C.5}$$

which states that the discharged power at any time slot must not exceed the maximum discharging rate that the battery supports. Six,

$$c_i^+ \leq C_{max}^+ \qquad ,1 \leq i \leq n \tag{C.6}$$

which states that the charged power at any time slot must not exceed the maximum supported charging rate. Seven,

$$r_i \leq R_{max} \qquad ,1 \leq i \leq n \tag{C.7}$$

which states that the amount of energy that the battery stores is bounded by the battery's maximum storage capacity. Finally,

$$t_i, g_i, c_i^+, c_i^-, r_i \geq 0 \qquad ,1 \leq i \leq n \tag{C.8}$$

which states that the decision variables are all non-negative.

The formulated problem is a convex optimization problem [17] as the objective is a convex function that we seek to minimize, all equality constraint functions are affine, and all non-equality constraints are convex functions. The solution of convex problems can be found quickly and there are well-developed tools that can be used to calculate the optimal solution efficiently such as the CVX package [18], which is the one used in our implementation.

IV. Limited-Horizon Control

We discussed previously how to find the optimal control strategy for a battery with leakage and conversion losses when the DC's power demands throughout the whole billing cycle are known. We now consider the case where we only know predictions of the DC's power demands in a short duration in the future (referred to as the prediction window), and we propose an algorithm that uses these predictions in order to decide how much energy the battery needs to charge/discharge at each time slot while accounting for the battery's energy losses. A pseudo code of the proposed algorithm is presented in order to better illustrate our algorithm. This pseudo code gets launched at the beginning of each slot j in the billing cycle and takes the following inputs:

- j the index of the current slot for which battery control decisions need to be made, where $1 \leq j \leq n$.
- D_j the power demanded by the DC at the j^{th} slot.
- Φ the specs of the battery.
- w the length of the prediction window which represents the number of slots in the future for which the DC's power demands need to be predicted.
- t_{max} the maximal amount of power drawn from the grid so far (up to the j^{th} slot in the current billing cycle), which is calculated as: $t_{max} = \begin{cases} 0 & ,j=1 \\ \max_{1 \leq k < j}\{t_k\} & ,j>1 \end{cases}$.

As illustrated in the pseudo code, our proposed algorithm starts by predicting the power demands of the DC in the future w slots (Line 1), where these predicted power demands are referred to by $\hat{D}_{j+1}, \hat{D}_{j+2}, \ldots, \hat{D}_{j+w}$. These predicted demands can be obtained using any machine learning technique that provides accurate predictions. The focus of this paper is on how to use these predictions to make good battery control decisions and not on what technique to use to obtain accurate predictions. The second step that the algorithm performs is to fetch r_j which represents the amount of available stored energy in the battery at the beginning of the j^{th} slot (after accounting for both conversion and leakage losses). Now in order to determine the best control actions, our algorithm solves in Line 3 a convex optimization problem called the Limited-Horizon Optimization Problem (LHOP) that has similarity with full-horizon optimization problem that was introduced in Section III but with the following key differences. First, LHOP seeks to minimize only the electricity costs associated with providing the DC's predicted power demands in the period from slot j to slot $j+w$. Recall that the Peak Charge is calculated based on the slot with the maximum power drawn from the grid throughout the whole billing cycle. When calculating the Peak Charge, LHOP considers the maximum power drawn from the grid from the the first slot in the billing cycle and up to the $j+w$ slots. The slots beyond $j+w$ are not considered by LHOP as it is hard to predict the DC's power demands for more than w slots. The constraints (C'.1 to C'.8) in LHOP are the exact equivalent of the constraints (C.1 to C.8) that were explained before for the full optimization problem with the exceptions that the constraints consider only the decision variables involved in the period from slot j and up to $j+w$ and that the control decisions need to meet the predicted DC power demands. LHOP is a convex optimization problem [17] as the objective is a convex function that we seek to minimize, all equality constraint functions are affine, and all non-equality constraints are convex functions. Solving LHOP returns the best control decisions that need to be made in the period from slot j and up to slot $j+w$. Our algorithm then commits only to the control decisions in the j^{th} slot (Line 4) that are returned by solving LHOP, while the control decisions in each of the following slots will be determined later when the pseudo code of our algorithm is launched again at the beginning of each one of those following slots.

V. Evaluation

We rely on the real power prices and on the estimated power demands for Google DC that were introduced in Section II, and we conduct comparative experiments to evaluate how much money Google DC ends up saving in a one billing cycle when operating a battery of a certain type using our proposed peak shaving battery control techniques. Different types of batteries are considered in our experiments where we evaluate

Algorithm 1 LimitedHorizonControl(j, D_j, Φ, w, t_{max})

$\overbrace{[\hat{D}_{j+1},\, \hat{D}_{j+2},\, \ldots,\, \hat{D}_{j+w}\,]}^{\text{Predicted Power Demands}} \leftarrow$ predictFutureDemands(w)

$r_j \leftarrow$ getAmountOfStoredEnergy();

Solve Limited-Horizon Optimization Problem:

$$\textbf{Minimize} \quad \underbrace{\alpha\xi \sum_{i=j}^{j+w} t_i}_{\text{Energy Charge}} + \underbrace{\beta max\{ \max_{j \le i \le j+w} \{t_i\}\,,\, t_{max}\}}_{\text{Peak Charge}}$$

subject to

$$t_i = g_i + c_i^+ \qquad , j \le i \le j+w \qquad \text{(C'.1)}$$

$$g_i + c_i^- = \begin{cases} D_i & , i = j \\ \hat{D}_i & , j < i \le j+w \end{cases} \qquad \text{(C'.2)}$$

$$r_i = \begin{cases} r_j & , i = j \\ \eta_c c_{i-1}^+ + \eta_l (r_{i-1} - c_{i-1}^-) & , j < i \le j+w \end{cases} \qquad \text{(C'.3)}$$

$$c_i^- \le r_i \qquad , j \le i \le j+w \qquad \text{(C'.4)}$$

$$c_i^- \le C_{max}^- \qquad , j \le i \le j+w \qquad \text{(C'.5)}$$

$$c_i^+ \le C_{max}^+ \qquad , j \le i \le j+w \qquad \text{(C'.6)}$$

$$r_i \le R_{max} \qquad , j \le i \le j+w \qquad \text{(C'.7)}$$

$$t_i, g_i, c_i^+, c_i^-, r_i \ge 0 \qquad , j \le i \le j+w \qquad \text{(C'.8)}$$

Make Control Actions Specified by (t_j, g_j, c_j^+, c_j^-)

the case when the DC is supplied by each of the following battery types:

- **Lead-Acid (LA):** this battery type uses electrochemistry to store and to discharge energy.
- **Lithium-Ion (LI):** relies also on electrochemistry but uses different chemical components where the cathode is a lithiated metal oxide and the anode is a graphite carbon.
- **Ultra-Capacitors (UC):** uses a double layer electrochemistry to store energy between the electrodes.
- **Fly-Wheels (FW):** uses the momentum of a wheel/cylinder to store energy.
- **Optimal (OPT):** represents an ideal battery that has zero conversion and leakage losses and unlimited charging/discharging rate.

The first four types represent the most popular battery technologies that are found in DCs [16], whereas the OPT battery represent an unrealistic ideal battery. The specs of these batteries are presented in Table I and are based on [16]. We assume that there is no limit on the capacity of each battery type (i.e., $R_{max} = \infty$) and that there is no stored energy initially at the beginning of the billing cycle when evaluating the different types of batteries (i.e., $R_{init} = 0$).

TABLE I: Specs for the considered battery types [16].

	LA	LI	UC	FW	OPT
Conversion Efficiency (%)	75	85	95	95	100
Leakage Efficiency (% per day)	70	90	80	1	100
Max Charging Rate (mega Watt)	16	16	8	8	∞
Max Discharging Rate (mega Watt)	8	8	8	8	∞

A. Full-Horizon Control Evaluation

Fig. 4 shows the total electricity bill for running the Google DC for a one billing cycle under different scenarios, where the

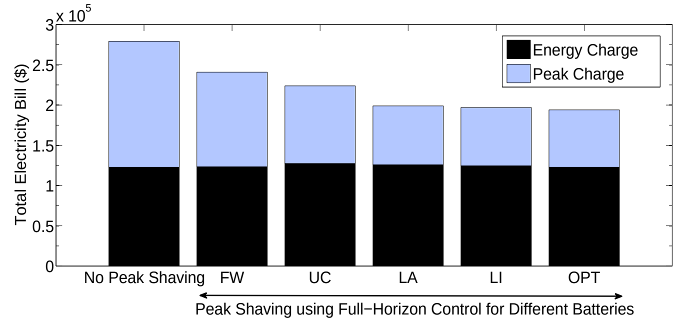

Fig. 4: Full-horizon control monetary savings for the different types of batteries based on Google traces.

total bill is broken down into the Energy Charge and the Peak Charge for each scenario. The "No Peak Shaving" scenario represents the case when the DC's power demands are drawn only from the grid without using any battery for peak shaving. The other scenarios in Fig. 4 show the total electricity bill for Google DC when different types of batteries were used to shave the peak where each type of those batteries is operating based on the decisions of the full-horizon controller that was proposed in Section III. The results clearly highlight that the DC's total electricity bill can be reduced significantly if our proposed full-horizon control technique was used to control how much energy needs to be charged/discharged over time for the different types of batteries. Observe that the total electricity bill is lower for the LI and LA battery types when compared to the FW and UC as the former types have lower leakage losses than the latter types, which allows storing larger amount of energy to be used to shave the peak that is encountered later, without leaking much of their stored energy over time. The Energy Charge of the FW, UC, LI and LA batteries are slightly higher than those of "No Peak Shaving" due to the leakage and conversion losses which increase the amount of energy that the DC consumes over time. However, these extra Energy Charge leads into significant reduction in the Peak Charge which leads in turn into significant reduction of the total electricity bill. It is worth mentioning that the results in Fig. 4 represent the maximum amount of savings that each battery type can achieve and thus can be considered as a factor when deciding what type of battery a DC should be supplied with based on its power demands. Other factors can also affect such choice including the capital expenses and the facility space that each battery type requires.

B. Limited-Horizon Control Evaluation

Fig. 5 shows the total electricity bill associated with running Google DC for a one billing cycle when different types of batteries are used for peak shaving. For each battery type, we show the total electricity bill when operating the battery using the following control techniques:

- **Limited-Horizon (Oracle):** is the control algorithm that we proposed in Section IV when operating under 100% accurate predictions of the DC's power demand in each of the following four slots (i.e., $w = 4$).

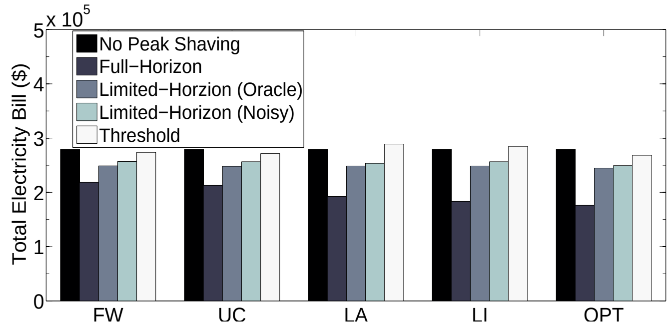

Fig. 5: Comparative evaluation of the different control techniques for each battery type based on Google traces.

- **Limited-Horizon (Noisy):** is similar to the previous case with the exception that a random noise drawn from a Gaussian distribution with zero mean and a standard deviation of 200 kilo Watts is added to the predicted power demand that are provided to our algorithm in each time step. The added noise represents prediction errors and can take either a positive or negative value to mimic over or under estimation of the predicted power demands.
- **Threshold:** is a well-known technique [12] that compares the demanded power at each slot against a threshold. If the demanded power is below the threshold, then the power difference is charged into the battery. Otherwise, the battery discharges the difference. Tuning the threshold for each battery type is done by evaluating different threshold values on one-day traces and then picking the value with the least electricity expenses as the selected threshold.

In addition to those control techniques, we show the total electricity bill for the No Peak Shaving and the Full-Horizon control cases. From Fig. 5, observe that for each type of battery the Limited-Horizon control (both Oracle and Nosy cases) had a lower total electricity bill than the No Peak Shaving case. As expected, the Noisy case had higher electricity bill than the Oracle case due to the added prediction errors that affected slightly the decisions of our algorithm. Obviously the total electricity bill for the Limited-Horizon control is higher than that of the Full-Horizon control as the latter has the advantage of knowing the DC's power demand throughout the whole billing cycle which allows it to make the optimal battery control decisions. Finally, observe that the Limited-Horizon control under both Oracle and Noisy case outperformed the well-known Threshold technique. These results are when the Limited-Horizon Algorithm relied only on predictions of the DC's power demands in the following four slots. Recall that each slot has a duration of 15 minutes and thus this represents the case where our algorithm knows the predicted power demands in a short duration of one hour in the future. Further experiments (not included here due to space limitation) showed that considering longer prediction window would reduce further the electricity bill of the Limited-Horizon controller and push it very close to the Optimal Full-Horizon case.

VI. Conclusion and Future Work

This paper proposes efficient battery control strategies for shaving the peak charge and for minimizing the total DC electricity bill. Our proposed control strategies are based on solving a well-formulated convex optimization problem that considers both leakage and conversion battery losses and that operates under full-knowledge or under limited-knowledge of the DC's future power demands. The monetary savings that the proposed techniques achieve were estimated based on real traces from a Google DC. For future work, we intend to include other battery aspects (e.g. life efficiency, capital cost and required space) as constraints in the full-horizon optimization problem. We also plan to develop a technique that can predict accurately the DC's future power demands.

References

[1] J. Koomey, "Growth in data center electricity use 2005 to 2010," *A report by Analytical Press, completed at the request of The New York Times*, 2011.

[2] L. Barroso, J. Clidaras, and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis lectures on computer architecture*, vol. 8, no. 3, pp. 1–154, 2013.

[3] C. Wang, B. Urgaonkar, G. Kesidis, U. Shanbhag, and Q. Wang, "A case for virtualizing the electric utility in cloud data centers," in *Proceedings of USENIX conference on Hot Topics in Cloud Computing*, 2014.

[4] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Release-time aware vm placement," in *IEEE GLOBECOM Workshop on Cloud Computing Systems, Networks, and Applications (CCSNA)*, 2014.

[5] M. Shojafar, N. Cordeschi, D. Amendola, and E. Baccarelli, "Energy-saving adaptive computing and traffic engineering for real-time-service data centers," in *IEEE Communication Workshop (ICCW)*, 2015.

[6] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Towards energy-efficient cloud computing: Prediction, consolidation, and overcommitment," *IEEE Network Magazine*, 2015.

[7] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Efficient datacenter resource utilization through cloud resource overcommitment," in *INFOCOM Workshop on Mobile Cloud and Virtualization*, 2015.

[8] H. Xu and B. Li, "Reducing electricity demand charge for data centers with partial execution," in *ACM Proceedings of the international conference on Future energy systems*, 2014, pp. 51–61.

[9] C. Wang, B. Urgaonkar, Q. Wang, G. Kesidis, and A. Sivasubramaniam, "Data center power cost optimization via workload modulation," in *International Conference on Utility and Cloud Computing*, 2013.

[10] R. Urgaonkar, B. Urgaonkar, M. Neely, and A. Sivasubramaniam, "Optimal power cost management using stored energy in data centers," in *ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, 2011, pp. 221–232.

[11] A. Oudalov, R. Cherkaoui, and A. Beguin, "Sizing and optimal operation of battery energy storage system for peak shaving application," in *IEEE Power Tech.*, 2007, pp. 621–625.

[12] M. Johnson, A. Bar-Noy, O. Liu, and Y. Feng, "Energy peak shaving with local storage," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 3, pp. 177–188, 2011.

[13] C. Reiss, J. Wilkes, and J. Hellerstein, "Google cluster-usage traces: format+ schema," *Google Inc., White Paper*, 2011.

[14] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Energy-efficient resource allocation and provisioning framework for cloud data centers," *IEEE Transactions on Network and Service Management*, vol. 12, no. 3, pp. 377–391, 2015.

[15] M. Ghamkhari and H. Mohsenian-Rad, "Optimal integration of renewable energy resources in data centers with behind-the-meter renewable generator," in *nternational Conference on Communications (ICC)*, 2012.

[16] D. Wang, C. Ren, A. Sivasubramaniam, B. Urgaonkar, and H. Fathy, "Energy storage in datacenters: what, where, and how much?," in *ACM SIGMETRICS Performance Evaluation Review*, 2012, pp. 187–198.

[17] S. Boyd and L. Vandenberghe, "Convex optimization," *Cambridge University Press*, 2004.

[18] Inc. CVX Research, "CVX: Matlab software for disciplined convex programming, version 2.0," 2012.