This Article Will be Appear at the 39th International Symposium on Computer Architecture (ISCA), 2012

Managing Distributed UPS Energy for Effective Power Capping in Data Centers

Blind Review

Abstract

Power over-subscription is a well-known technique to reduce both one-time capital costs and operating costs for modern data centers. However, designing the power infrastructure for a lower operating power point than the aggregated peak power of all servers requires dynamic techniques to avoid high peak power costs and, even worse, tripping circuit breakers. This work proposes the first design for distributed per-server UPS architectures that store energy during low activity periods and use this energy during power spikes. It leverages the distributed nature of the UPS batteries and designs algorithms that prolong the duration of their usage. This approach shaves 19.4% of the peak power for modern servers, allowing the installation of 24% more servers within the same power budget. More servers amortize infrastructure costs better and, hence, reduce total cost of ownership per server by 6.5%, with a corresponding increase in profits. These benefits increase considerably as servers become more energy proportional.

1 Introduction

The costs of building and running a data center, and the capacity to which we can populate it, are all driven in large part by the peak power available to, or used by, that data center. This work demonstrates techniques to significantly reduce the observed peak power demand for data centers with distributed UPS batteries, enabling significant increases in data center capacity and reductions in cost.

Modern data center investments consist of one-time infrastructure costs that are amortized over the lifetime of the data center (capital expenses, or capex) and monthly recurring operating expenses (opex) [24]. Capex costs are proportional to the provisioned IT power per facility, estimated at $10-20 per Watt [13, 40, 51], as each Watt of computing power requires associated support equipment (cooling, backup, monitoring, etc.). Utilities typically charge a power premium that is tied to the peak power. This can become a significant portion of the monthly bill, up to 40% [21]. This paper examines the use of distributed batteries in the data center to reduce both capex and opex costs.

Power is still commonly over-provisioned in the data center to accommodate peaks and to allow for future expansion. However, to improve power infrastructure utilization, we can intentionally over-subscribe (under-provision) power [15, 24, 26, 29, 38]. Over-subscribing provisions power infrastructure to support a lower demand than the largest potential peak and employs techniques to prevent power budget violations. In the worst case, large power spikes may trip circuit-breakers and disable whole sections of the data center, which translates to costly down time. To avoid this, data centers can employ power capping approaches such as CPU capping, virtual CPU management, and dynamic voltage and frequency scaling (DVFS) [35, 41, 29]. CPU capping limits the time an application is scheduled on the CPU. Virtual CPU management limits virtual machine power by changing the number of virtual CPUs. DVFS attacks the peak power problem by reducing chip voltage and frequency. However, all of these techniques result in performance degradation. This is a problem for any workload that has performance constraints or service-level agreements because we are forced to apply these performance-reduction mechanisms at the exact time that performance is critical – at peak load.

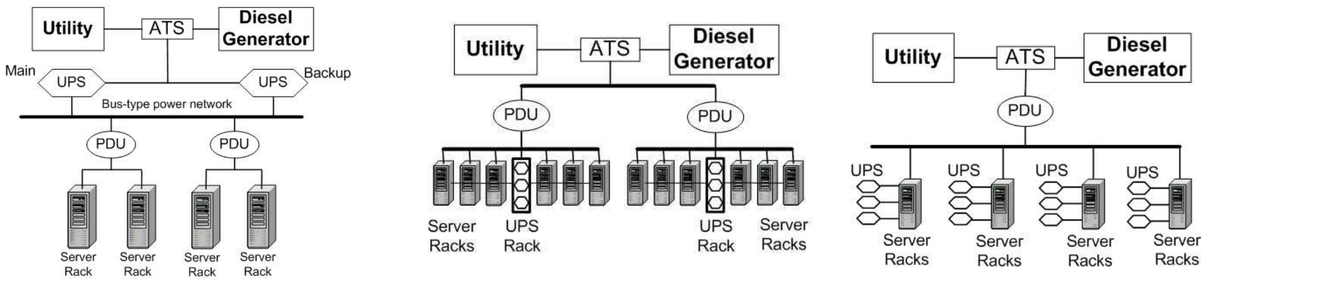
Govindan, et al. [21] introduce a new approach that has no performance overhead in the common case. They leverage the energy stored in a centralized data center UPS to provide energy during peak demand, effectively hiding the extra power from the power grid. This technique is shown to work well with brief (1-2 hours), high-magnitude power spikes that can be completely ”shaved” with the energy stored in batteries; however, it is less effective for long (8-10 hours) spikes. For longer spikes, they suggest a hybrid battery-DVFS approach.

However, many large data centers do not employ centralized batteries. Distributed, per-server batteries represent a more economical solution for battery backup. They scale proportionally with the data center size and hence are not over-provisioned to account for future data center expansions or to prevent single point failures. Google currently employs this topology in their state-of-the-art data centers enabling dramatic capex reductions compared to the centralized approach [18]. In this work, we discuss the applicability of battery-enabled power capping to distributed UPS topologies. We present details on the sizing and the technology alternatives of per server batteries and consider several approaches that orchestrate battery charging and discharging while addressing reliability and availability concerns. This research goes beyond prior work by modeling realistic data center workload patterns over a multi-day period and by arguing that battery over-provisioning is financially beneficial. Placing additional servers under a given power budget permits reductions of the data center total cost of ownership per server on the order of 6%. This is equivalent to more than $20M for a datacenter with 28,000 servers.

When leveraging a distributed UPS architecture to shave peak power, challenges arise due to the lack of heavy over-provisioning and the distributed nature of the batteries. The absence of over-provisioned batteries means we need to justify the use of larger batteries based purely on cost savings from power capping. We need policies to determine how many batteries to enable, which batteries to enable, and when. But there are also opportunities compared to the prior solution. In the centralized architecture, all power comes from either the battery or the utility. Thus, when batteries are enabled, they supply all datacenter power and drain quickly – if we only supply the over-threshold power, the batteries can sustain longer peaks. This is easily done in the distributed architecture by simply enabling enough batteries to hide the desired peak.

This paper makes the following unique contributions. (1) It is the first work to describe a solution for peak power capping which utilizes distributed UPS batteries. (2) It provides a full financial analysis of the benefits of battery-based power capping, including optimal battery technology and battery size. (3) It is the first work on battery-based power capping which models realistic workloads and demonstrates implementable policies for battery management. Prior work on centralized UPSs was more of a limit study for an oracle controller.

This paper is organized as follows. Section 2 presents common UPS topologies and the associated trade-offs. Section 3 quantifies the reduction in total cost of ownership per server with the applied solution. In section 4 we contrast alternative battery technologies for frequent battery charge/discharge in the data center context and elaborate on their properties. In section 5, we present our policies. In section 6, we discuss our experimental methodology. Section 8 reviews related work in power capping techniques, and section 9 concludes.



(a) Centralized (b) Distributed (Facebook) (c) Distributed (Google)

Figure 1: Power hierarchy topologies with (a) centralized UPS and (b,c) distributed UPS solutions.

2 Background

Power is delivered into the data center through a utility line. Data centers are also equipped with a diesel generator unit which acts as a secondary source of power during a utility failure. To facilitate switching power between the utility and the diesel generator, an automatic transfer switch (ATS) selects the source of power, which takes 10-20 seconds [21]. During this short and critical interval, the UPS units supply the power to the data center. In the centralized topology shown in figure 1(a), the power from a single UPS is fed to several Power Distribution Units (PDUs) to route the power to racks and servers. To eliminate the transfer time of the power line to the UPS, data centers commonly use double conversion UPSs. With double conversion UPSs, power is transformed from AC-to-DC to be stored in batteries and then from DC-to-AC to be used by the racks and servers. Although this organization has zero transfer time to the UPS (the UPS is always in the power path), the availability of the whole data center is dependent on the UPS. Additionally, double conversion introduces 4-10% power losses during normal operation [18].

The centralized UPS topology in figure 1(a) does not scale well for large data centers. The inefficiency of the AC-DC-AC conversions become more costly. Removing the double conversion requires transferring DC power from the centralized UPS to the servers. DC power distribution introduces higher losses on the power wires and is not suitable for long distances. Moreover, the high degree of over-provisioning renders the centralized solution increasingly expensive as the size of the data center grows. Consequently, distributed UPS placement is attractive for large data centers.

The distributed design adopted by Facebook is shown in figure 1(b). A cabinet of batteries for every 6 racks, or a total of 180 servers, replaces the centralized UPS [14]. This design avoids the double conversion by customizing the server power supply unit to support both AC power (from grid) and DC power (from the battery cabinet). Note that AC power is distributed over long distances and then DC power is generated very close to where it is actually used (adjacent racks). Google goes even further, attaching a battery on every server after the Power Supply Unit (PSU) [18], as depicted in figure 1(c). This design also avoids the AC-DC-AC double conversion, saving energy under normal operation, and brings the AC distribution even closer to the IT load.

Availability in data centers is a function of how often failures happen, the size of the failure domain, and the recovery time after each failure. UPS placement topology impacts the availability of the data center, particularly the associated failure domain. In the centralized design of figure 1(a), failure of the main UPS will trigger the backup UPS. Failure of the back up

UPS will translate to down time for the whole data center. The rack UPS design of figure 1(b) has a smaller failure domain (a UPS failure will affect 6 racks, or 180 servers) but is still significant. Thus, Facebook incorporates limited redundancy in the battery cabinet design (20% additional battery capacity). In Google's distributed design of figure 1(c), the domain of failure is only a single server, allowing the use of cheaper batteries and no redundancy. Therefore, the distributed topology is cheaper than centralized due to the smaller degree of over-provisioning, without significantly compromising data center availability. This paper explores power-shaving solutions that are appropriate for a data center with distributed batteries. We focus on the most extreme case – per-server UPS.

3 Total Cost of Ownership analysis

Modern data centers are typically power limited [51]. This means that the overall capacity (number of servers) is limited by the initial provisioning of the power supporting equipment, such as utility substations, diesel generators, PDUs, and cooling. If we reduce the peak computing power, we can add additional servers while remaining within the same power budget, effectively amortizing the initial investment costs over a larger number of servers. Moreover, extra work done per data center should result in fewer data centers, greatly reducing capex costs.

Distributed UPSs are currently designed to support the whole computing load long enough to ensure safe transition from the main grid to the diesel generator. This time window (less than one minute) translates to batteries with insufficient amount of stored energy for meaningful peak power shaving. Therefore, to enable peak power capping using UPS stored energy in the distributed context, we need to over-provision per server battery capacity. This section explores the degree of over-provisioning that makes financial sense, compares alternative battery technologies for server-level UPSs, and describes how to select battery parameters in order to maximize total profits.

The profitability of an investment is defined as the generated revenue minus the associated total cost of ownership (TCO). The data center revenue equals the number of servers times the income per server. We assume constant income per server. Therefore, maximizing the profitability per server is equivalent to minimizing the TCO per server. We now explain how placing more servers within the same power budget reduces TCO per server. Our TCO analysis is inspired from the data center cost chapter in Barroso and Holzle [24]. For simplicity, we assume sufficient initial capital, hence there are no monthly loan payments, and full capacity for the data center (limited by the provisioned power) from the first day. Financing the investment should similarly scale all TCO components and should not affect our results. Reducing the deployment time of the data center is a problem orthogonal to reducing capex costs and will not be treated here. The TCO/server is given by:

$$
\begin{aligned}
TCO/server =& (dataCenterDepreciation + dataCenterOpex + serverDepreciation + serverOpex)/N_{servers} \\
=& ((FacilitySpaceDepr + UPSDepr + PowerInfrastructureDepr + CoolingDepr + RestDepr) \\
&+ dataCenterOpex + serverDepr \\
&+ (ServerRepairOpex + (ServerEnergyOpex + ServerPowerOpex) * PUE))/N_{servers}
\end{aligned} \tag{1}
$$

In equation 1, data center depreciation is the monthly depreciated cost of building a data center. We assume, similarly to [24], 10 year straight-line depreciation, which means that assets lose a fixed amount of their value each month. The assets required for a data center are land, UPS and power infrastructure (diesel generators, PDUs, back-room switchgear), cooling infrastructure (CRAC, economizers), as well as several other components such as engineering, installation labor,

Data center Critical Power	10 MW
Server	Idle Power: 175W, Peak Power: 350W (measured)
Number of servers	28000 (critical power / server peak)
Average Server Utilization	50% [24]
Utility Prices	Energy: 4.7 c/KWh, Power: 12 $/KW [6, 21]
Server cost	$2000
PUE	1.15 [18]
Amortization Time	Infrastructure: 10 years, Servers: 4 years [24]

Table 1: TCO model assumptions

TCO component	TCO/month (TCO/month/server)	TCO/server trend with extra servers
Facility Space depreciation	193,750$ (6.92$)	Decreasing
UPS depreciation	3,733$ (0.13$)	Constant
Power Infrastructure depreciation	170.417$ (6.09$)	Decreasing
Cooling infrastructure depreciation	70,000$ (2.50$)	Decreasing
Rest depreciation (racks,monitoring,engineering,installation)	357,938$(12.78$)	Decreasing
Data center opex (maintenance, lighting)	229,527$ (8.20$)	Decreasing
Server depreciation	1,166,667$(41.67$)	Constant
Server opex (Service/repairs)	58,333$ (2.08$)	Constant
PUE overhead	55,467$ (1.98$)	Constant
Utility monthly energy cost	252,179$ (9.01$)	Constant
Utility monthly power cost	117,600$ (4.20$)	Decreasing
Total	2,675,610$(95.56$)	Decreasing

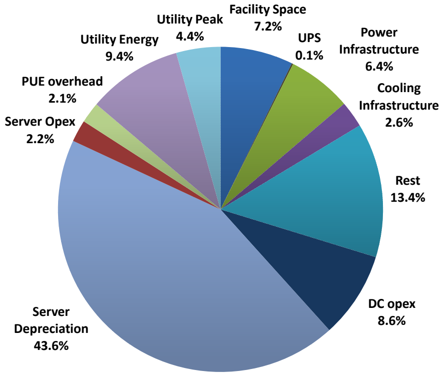


Figure 2: Total Cost of Ownership (TCO) [5]. The trend for most components of TCO/server as we cap power and add servers is to decrease.

racks, and system monitors that we include in *RestDepreciation*. The data center opex is the monthly cost for running the data center (infrastructure service, lighting). We collect the depreciation and opex cost information for a data center with 10MW provisioned computing power (critical power) from APC's commercial TCO calculator [5].

Server depreciation is the monthly depreciated value of the servers. However, servers typically have shorter lifetimes and are depreciated over 4 years. Server opex consists of server repairs (5% of server depreciation value) and the electricity bill. Utility charges have a power component and an energy component. The power component is based on the peak sustained power for a 15 minute window over the period of a month [6] while the energy is based on the total data center energy used. To account for the electricity consumed by infrastructure, excluding servers, we scale the total server peak power and energy by the power usage effectiveness (PUE), assumed at 1.15 [18]. For this study we use a customized commodity server, similar to Sun Fire X4270. This server has a total of 8 cores (Intel Xeon 5570) at 2.40 GHz, 8 GB of memory and costs $2000. The measured idle server power is 175W and the measured peak is 350W. We assume 50% average utilization for this analysis, which corresponds to operation at 262.5W. The inputs to our TCO model are summarized in table 1.

The table and the pie chart in figure 2 show the breakdown of TCO/month/server. The major TCO component is server depreciation (43.6%). Infrastructure related components, specifically facility space, power, cooling and data center opex costs, account for more than 35%. In the same table, we also present how the ratio of each TCO component per server changes if we are able to add additional servers within the same power budget. Server depreciation remains constant as well as server opex costs. However, infrastructure costs per server decrease as these costs get amortized over a larger number of servers. UPS TCO/server is constant because there is one battery for each additional server. The utility energy component remains constant because we assume the same utilization for the additional servers. The utility peak component decreases because the total power budget stays the same. The UPS cost (estimated as the total cost of the server-attached batteries) represents a very small portion of the TCO, it is marginally visible in the pie chart. Our proposal over-provisions batteries and increases the cost of the distributed UPS. In return, we can reduce aggregate peak power and safely accommodate a larger number of servers, effectively reducing other infrastructure costs per server. While our conclusions should remain valid across a number of input parameters, we acknowledge that the margins for battery overprovisioning vary according to the exact inputs to the model. We plan to make this battery-aware data center TCO model publicly available at the publication time of this paper.

4 Characterizing batteries for distributed UPSs

The properties of a UPS module depend on the batteries attached to it. Current UPS designs rely on lead-acid batteries because of their ability to provide large currents for high power applications at low cost. In this section, we discuss the alternative battery technologies for distributed UPSs, model battery behavior when employed for peak power capping, and elaborate on the selection of their parameters (capacity, cost, depth of discharge) to minimize TCO/server.

The spider graph in figure 3 compares the major competing battery technologies for high power applications, typical for servers, at the range of 12V and 15-30A: lead-acid (LA), Lithium Cobalt Oxide (LCO), and Lithium Iron Phosphate (LFP). Other technologies like NiCd, NiMH, or other lithium derivatives are excluded because they are dominated by one of the discussed technologies across all metrics. LA never performs best along any dimension except low temperatures, which is not applicable to data center applications [19]. LA is cheapest per Wh; however, LFP offers an order of magnitude more recharge cycles, hence provides better \$/Wh/cycle than LA. LCO is the most expensive technology and provides comparable recharge cycles to LA. The advantage of LCO technology is its high volumetric density (Wh/l) and gravimetric density (Wh/Kg). Lithium batteries have longer service life than LA and also recharge faster. Regarding safety of use, LA may release toxic gases when over-charged, while LCO may catch fire. LFP batteries are the safest because they have the highest margins for over-charging.

Properly selecting the technology and battery size depends on its use. UPS batteries in modern data centers are discharged only during a power outage. According to [39], the number of utility outages that affect data centers ranges from 1.5 to 4.4 per year. Therefore, cost, service life, and size are the most important parameters. The selection criteria become quite different when we re-purpose the batteries to be aggressively charged and discharged. Recharging cycles become crucial because continuous battery use may drastically shorten battery lifetime, resulting in frequent replacement costs that negatively affect TCO/server. Hence \$/Wh/cycle is a better metric than \$/Wh alone. Since LCO does poorly on both cost and cycles, it is not considered further.

We now focus on the per server distributed UPS design and explore the degree of overprovisioning that is most financially beneficial. Battery cost is estimated based on its 20h-rated capacity in Amp-hours (Ah) and the cost per Ah. We derive the required battery capacity based on the amount of power we want to shave and the corresponding energy stored in a battery for a given daily power profile. We derive the cost per Ah from [10, 34]. Tables 2 and 3 show all the inputs for the battery sizing estimation.

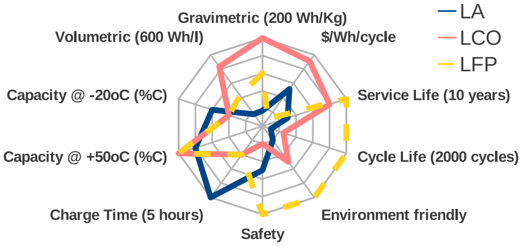


Figure 3: Comparison of battery technologies across different metrics [10, 52]

Input	Value		Reference
	LA	LFP	
Service time	4yrs	10yrs	[54, 34]
Battery Cost per Ah	2$/Ah	5$/Ah	[34, 10]
Depth of Discharge	40%	60%	Estimated (see figure 6)
Peukert's exponent	1.15	1.05	[23]
Existing Server Battery Capacity	3.2Ah		[18]
Recharge Cycles	f(DoD) – Table 3		[54, 50]
Battery Voltage	12V		[18]
Max Bat. Discharge Current	23A		Estimated (ServerPeak * PSUeff / Voltage)
PSUeff	0.8		[9]
Discharges per day	1		Based on waveforms from [17]
Battery losses	5%		[46, 55]

Table 2: Input values used for the battery cost estimation.

To derive the required battery capacity, we first set a peak power reduction goal and estimate the total energy that needs to be shaved at the data center level over the period of a day. We assume all batteries get charged and discharged once per day because, according to [17], all the traffic profiles of large distributed applications demonstrate a single peak. The daily shaved energy is equivalent to the integral between the power curve and the flat power line we set as the peak goal. For simplicity in this section we consider a power peak as a diurnal square pulse with a specified height and duration. For that workload, the required data center discharge energy is given by equation 2.

$$E_{DataCenter} = DataCenterPeakPower \times PowerReduction \times PeakTimePerDay \tag{2}$$

To simplify the analysis, we assume that all servers discharge their batteries at the same rate. We relax this assumption (and several other assumptions applied to this initial analysis) in Section 7. Equation 3 estimates the energy that each battery should provide to the associated server. Since the distributed battery is attached after the power supply the power drawn from the battery goes directly to the motherboard and is not wasted on losses of the Power Supply Unit (PSUefficiency).

$$E_{server} = \frac{E_{DataCenter} \times PSUefficiency}{N_{servers}} \tag{3}$$

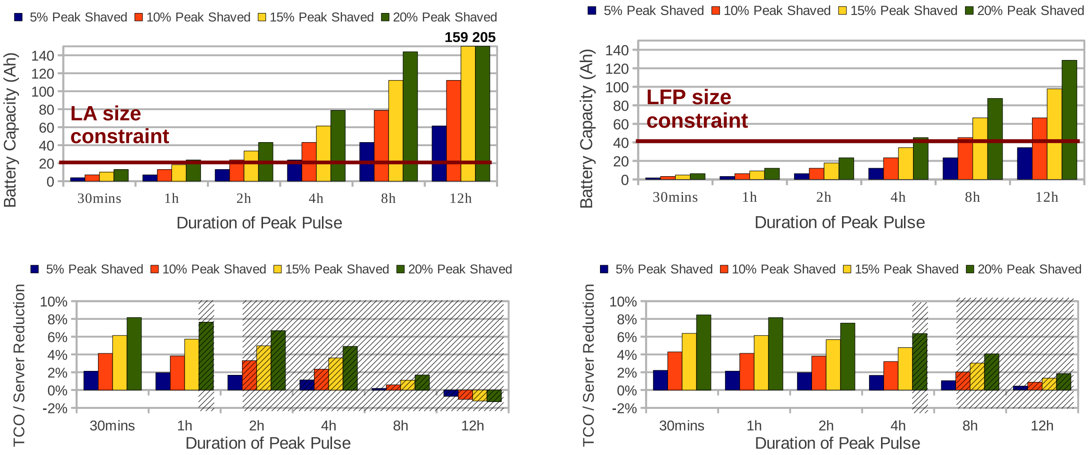
Given the energy each battery must provide, we estimate the energy stored per battery and the corresponding battery capacity using Peukert's law. This relation is given by equation 4, where C_{1h} is the battery capacity in Ah (1h means that the battery capacity, equivalent to charge, is measured drawing constant current for 1h), I is the discharge current, PE is Peukert's exponent, and T is the battery discharge time [49, 46]. Lead-acid batteries typically have a Peukert's exponent in the range of 1.05-1.25 while Lithium Iron Phosphate batteries have in the range of 1.03-1.07 [23].

$$T = \frac{C_{1h}}{I^{PE}} \quad \Rightarrow \quad C_{1h} = T \times I^{PE} = \frac{E_{server}}{V \times I} \times I^{PE} = \frac{E_{server}}{V} \times I^{PE-1} \tag{4}$$

We also account for battery depth of discharge (DoD), the degree to which we allow the battery to be drained. Fully discharging the battery (100% DoD) to extract the required amount of energy would seriously degrade the lifetime of the battery and translate to higher battery replacement costs (see table 3). Limiting DoD also allows us to use excess capacity

DoD (%)	10	20	30	40	50	60	70	80	90	100
Recharge cycles LA	5000	2800	1860	1300	1000	830	650	500	410	330
Recharge cycles LFP	100000	40000	10000	6000	4000	3000	2000	1700	1200	1000

Table 3: Recharge cycles as a function of depth of discharge (DoD). Deep battery discharge results in a fewer recharge cycles[50, 54].



(a) LA

(b) LFP

Figure 4: Battery capacities for different pulse widths and portion of peak power shaved. We also show the monthly TCO per server savings, assuming current battery costs, for the specified capacities of Lead-acid (LA) and Lithium Iron Phosphate (LFP) batteries. When the battery cannot fit within a 2U server, the associated savings are hatch shaded.

for power capping without increasing exposure to power failures. Consequently, we only want to discharge the battery partially. However, the less we discharge a battery, the larger battery capacity we need in order to discharge the same amount of energy. For discharge current, we conservatively assume the max value of the server current ($I_{MAX} = 23A$). Additionally, batteries lose a portion of their capacity as they age. Once they reach 80% of their original capacity, battery manufacturers consider them dead. We pessimistically take this effect into account by scaling the capacity by a factor of 1/0.8. Using equation 4, we get the provisioned 1h-rated battery capacity for each server battery (equation 5).

$$C_{1h}\ provisioned = C_{1h} \times \frac{1}{DoD} \times \frac{1}{0.8} = \frac{E_{server}}{V} \times I_{discharge}^{PE-1} \times \frac{1}{DoD} \times \frac{1}{0.8} \qquad (5)$$

We convert the 1h-rated capacity into 20h-rated with the equation: $C_{20h}\ provisioned = \left(C_{1h\ provisioned} \times 20^{PE-1}\right)^{1/PE}$ [46, 49]. The 20h-rated capacity can be used to estimate the battery cost, because that is what manufacturers report.

The previous capacity estimation methodology allows us to translate a peak power reduction goal to per-server provisioned battery capacity and the associated cost. To compute the monthly UPS depreciation, we also need to know the average battery lifetime. The battery lifetime is equal to the min of the battery service time in months and the number of recharge cycles as a function of depth of discharge, divided by 30 (one recharge cycle per day):

$$UPSDepr = \frac{C_{20h}\ provisioned \times BatteryCostPerAh \times N_{servers}}{MIN\left(serviceLife, cycles(DoD)/30\right)} \qquad (6)$$

We use the described equations to contrast LA with LFP technologies as we vary the peak time in the power profile, study the effect of decreasing battery cost per Ah, and identify the depth of discharge that minimizes TCO/server. Figure 4 shows the provisioned battery capacity for a given peak power time and a targeted reduction in peak power as well as the respective TCO/server reduction. More energy needs to be stored in the battery to achieve the same reduction in peak power as the duration of peak power demand increases. Hence, the cost of the distributed UPS increases. In the LA case, over-provisioning is no longer helpful when the peak power lasts for 12 hours. This means that the additional distributed

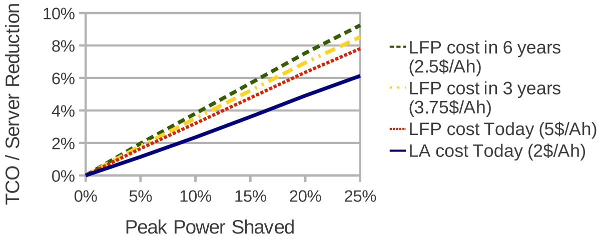


Figure 5: For the 2h pulse we show the projection of savings (ignoring space constraints) as the battery cost changes in the future [2].

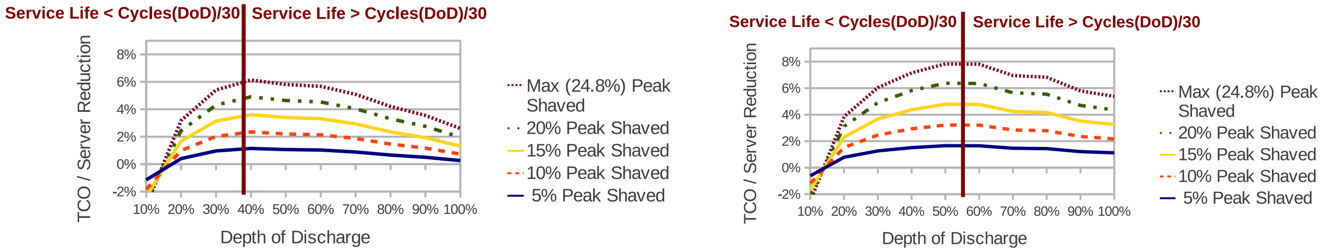
UPS cost is greater than the reduction of TCO/server due to amortization of the infrastructure costs on more servers. LFP batteries remain beneficial at 12 hours of peak demand. Size constraints only allow shaving 5% of the 2-hour peak demand, in the LA case, while we can shave 5% of an 8-hour pulse with LFP. In the TCO/server diagrams in figure 4, we denote the battery capacities that do not fit in a 2U server by hatch shading the respective columns. For the same spike duration, it always makes sense to shave more peak with a bigger battery, within size limitations. To further quantify these profits, we find using the analysis of section 3 that 6.8% monthly TCO/server reduction translates to $6.4 per month per server, or more than $21M over the 10-year lifetime of a data center with 28,000 servers.

Figure 5 presents the monthly TCO/server savings as the battery costs change. The projection for LA batteries is that costs do not change, while LFP prices are expected to be reduced due to the push for cheaper hybrid and electric cars [3]. For these graphs we assume that LFP cost reduces yearly at 8% [2]. At 4h peak per day, we achieve 7% TCO/server reduction for lead-acid, ignoring space considerations, while this value drops to 1.35% for a battery that fits within a 2U server design. Using LFP batteries today we can achieve 8.5% TCO/server reduction and these savings will increase to 9.6% in the next 6 years.

Figure 6 shows the relation between depth of discharge and the TCO/server gains for both LA and LFP technology. There is a clear peak for the values 40% and 60% DoD, respectively. For low DoD values, the battery costs dominate the savings, because we need larger batteries to provide the same capping. For large DoD values, the lifetime of the battery decreases and more frequent replacements increase the UPS cost. The peak reduction of TCO/server occurs when the number of recharge cycles / 30 is equal to the battery service life. Note that due to the battery overprovisioning, less than 5% charge can sustain the server for 1 min and ensure data center continuity. Therefore, lifetime considerations affect TCO/server well before data center continuity becomes a concern (DoD $>$ 95%).

To summarize our discussion on battery technologies and battery properties, we conclude:

- LFP is a better, more profitable choice than LA for frequent discharge/recharge cycles on distributed UPS designs. This is due to the increased number of cycles and longer service lifetime, better discharge characteristics, higher energy density, and the reduction in battery costs expected in the near future.
- Battery-based peak power shaving using existing batteries is only effective for brief spikes. To tolerate long spikes, larger batteries are necessary. However, the benefits from increased peak power shaving outweigh the extra battery costs even when high demand lasts 12 hours.
- It makes sense to increase the capacity of the battery to the extent that it fits under the space constraints. This translates to increased power reduction and more savings.



(a) LA

(b) LFP

Figure 6: The relation between targeted depth of discharge and the reduction in TCO.

- For each battery technology, there is a depth of discharge value that maximizes savings (40% for LA and 60% for LFP). This is the point where battery lifetime is no longer limited by the battery service time and needs to be replaced earlier due to frequent charging and discharging.

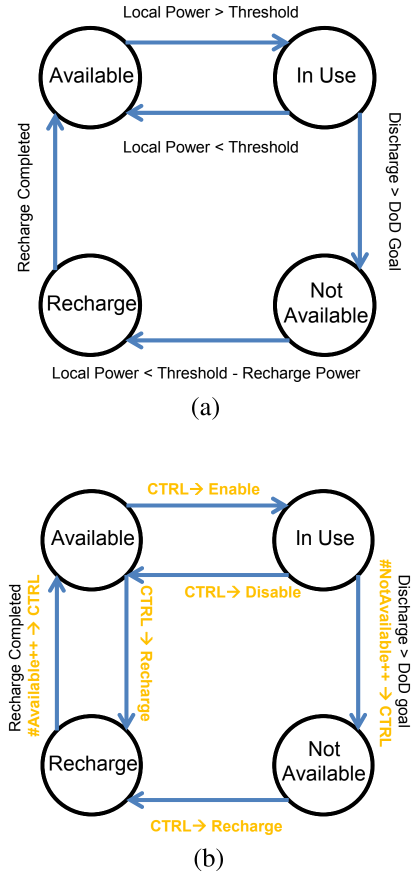
5 Policies

The analysis in the previous section assumes a simplified model of the power profile and perfect coverage of that peak by the batteries. As we move to a more complex model of real data center workloads and the associated power profiles, we investigate a number of policies for peak power shaving which react to the observed load on the data center power infrastructure.

We evaluate four policies for peak power shaving. In the first policy, we model a single cluster of batteries, as in the centralized UPS, providing energy for the whole 1000 server cluster. For the rest of the policies, there is a single battery per server. The centralized battery policy is examined for comparison purposes with [21]. For the centralized approach, we assume all power comes from either the utility or the battery. Prior work assumes that frequent duty-cycling of the UPS could create the illusion of partitioned power delivery. However, this is a troublesome solution. Rapid duty-cycling of large batteries creates large harmonics and requires huge capacitors. It also places pressure and wear on a switch that is designed for infrequent use and represents a potential single point of failure. Conversely, with completely distributed UPSs there is no need for duty-cycling. Since stored energy is already segmented into multiple batteries, we simply enable enough distributed batteries to hide the over-threshold power.

We examine policies that work at several different levels of the power hierarchy. Power budgeting at the PDU level coordinates the batteries of all the servers powered by the same PDU. Power budgeting at the cluster level coordinates all the machines in a cluster. The communication protocol to remotely enable/disable batteries or start recharge can be easily supported with existing network interfaces, such as SNMP or IPMI. The actual control algorithm can be implemented entirely in software. The policies manage battery discharge and also recharge. Recharging the batteries requires appreciable power and is thus best performed when the overall data center load is low. To summarize, we consider the following policies:

1. **Centralized LA-based UPS (Centr)**. Similar to the strategy proposed by [21], this policy uses a large, centralized UPS to shave peak power. When data center power exceeds a preset threshold, this policy switches from grid power to battery power. The value of the power threshold defines how aggressively we cap power. When the overall power



(a)

(b)

Figure 7: Per battery state machines for fully distributed approach with only local decisions (a) and coordinated approach based on a controller (b). For the controller logic see figure 8.

```
State=[NumAvail,NumInUse,NumNotAvailable,NumRecharging]
/* NumAvail: Batteries with charge currently idle (Available state)*/
/* NumInUse: Batteries with charge currently discharging (Inuse state)*/
/* NumNotAvail: Batteries without sufficient charge (NotAvailable state)*/
/* NumRecharging: Batteries currently recharging (Recharge state)*/

/* Get difference between current and targeted power */
delta = load - threshold
/* Get difference in batteries */
$\Delta$Bats = abs(delta)/serverAveragePower
if (delta $> 0$) then
    /* Over peak goal */
    EnBats = min(NumAvail, $\Delta$Bats)
    NumInUse += EnBats
    NumAvail -= EnBats
    Enable EnBats batteries
end if
if (delta $< 0$) and ($\Delta$Bats $> 25$) then
    /* Under peak goal */
    DisBats = min(NumInUse,$\Delta$Bats)
    NumAvail += DisBats
    NumInUse -= DisBats
    Disable DisBats batteries
    RSlackBats = $\Delta$Bats-DisBats
    if RSlackBats $> 0$ then
        NumRecharging += RSlackBats
        Recharge RSlackBats batteries
    end if
end if
```

Figure 8: Controller algorithm

consumption is less than the threshold and there is sufficient margin to recharge the battery without exceeding the budget, battery recharge is enabled. In this policy the UPS does not support load-proportional discharge (i.e. either the utility or the battery provides 100% of data center power). We model the cluster level device as a UPS with lead-acid batteries provisioned to last 48 minutes under full load, similar to [21]. The battery recharge begins once it reaches the LA discharge goal (40% DoD) and we detect sufficient margin between the preset threshold and the current data center power so that the recharge power will not result in a budget violation.

2. **Server with Local Controller (ServCtrl)** This policy is similar to *Centr*, but applied in a completely distributed fashion. Each server has its own local LFP battery and a controller that periodically monitors server power. Figure 7(a) shows the state machine for this controller. If measured server power is higher than the local power threshold (peak power cap / number of servers), then the controller switches the server to battery power. Recharge activates when battery depth of discharge reaches the set goal (60% for LFP) and there is sufficent margin between the current server power and the target power cap to accommodate recharge power.

3. **PDU with Centralized Controller (PduCtrl)**. This policy implements a controller per PDU. Each controller coordinates the operation of the batteries associated with the servers under a common PDU in order to match the energy to be shaved with the number of discharging batteries. It periodically estimates the power difference between current PDU power and the targeted PDU peak. As soon as this delta becomes positive, the controller estimates the approximate number of batteries that should start discharging (*abs(delta)/serverAveragePower*). Similarly, when the delta is negative and there are discharging batteries, the local controller will signal a number of batteries proportional to the magnitude of the estimated difference to stop discharging. We introduce an additional condition that the number of batteries we want to stop needs to be more than 25, which provides some hysteresis. The exact value should be tuned according to how fast the workload changes and how fast the controller responds (our controller period is 3 mins). Higher values than 25 would make the controller slow to stop batteries discharging and would waste stored energy. Lower values risk the controller fluctuating between enabling and disabling modes when near the threshold.

   Figure 7(b) and 8 show the state machine for the local battery controller and pseudo-code for the algorithm running on the PDU level controller. Arcs labeled in bold (orange) correspond to events sent to or from the centralized controller, whereas the other arcs, such as determining when the DoD goal has been met, remain local decisions. The controller attempts to distribute the enabling and disabling of batteries evenly by employing a static sequence that is interleaved across racks, and sets the ordering for both battery enabling and disabling. When no batteries are currently enabled, the controller gradually signals discharged batteries to recharge. The controller also forces batteries that have not yet discharged to the DoD goal, but have not recently been recharged, to begin recharging in anticipation of the next day's peak. Staggering recharge limits the possibility of power violations during low demand periods due to recharge power drawn from the utility.

4. **Cluster with Centralized Controller (ClustCtrl)**. This policy applies the same logic as *PduCtrl*, but at the cluster level. Data center power delivery involves circuit breakers at several levels of the hierarchy. The previous policy, *PduCtrl*, maintains a power budget at the PDU level allowing over-subscription with more racks. This policy targets a power budget at the cluster level, enabling over-subscription with more PDUs. We again employ a sequence to enable and disable batteries to evenly distribute the power load across the level of the power hierarchy.

6 Methodology

This section describes our methodology for evaluating battery-stored energy as a data center peak power capping solution. In section 3 we derived upper bound power savings based on a simplified model of the workload and oracle knowledge of that workload. Here we present the tools used to to model a variable, data-driven workload and realistic reactive capping policies that do not rely on oracle knowledge.

UPS type	Span	Voltage	Max Current	Capacity	Time peak load sustained	Recharge time (from 100%DoD)
Centralized (LA)	cluster (1000 servers)	208V	1658A	2018Ah	48mins	4h
Distributed (LFP)	server	12V	23A	40Ah	92mins	2h

Table 4: Provisioned battery characteristics

Workload	Service Time Mean	Interarrival Time Mean	Reference
Search	50ms	42ms	[31]
Social Networking	1sec	445ms	[4]
MapReduce	2 mins	3.3 mins	[7]

Table 5: Workloads

6.1 Power Modeling

We developed a discrete-event simulator that captures the behavior of 1000 server nodes at the back-end of large distributed web applications. Each server is modeled as a queue with 8 consumers (cores) per server to service the incoming requests. Thus, we simulate a large network of M/M/8 queues. Our simulator monitors all levels of the data center power delivery hierarchy, namely the servers, racks, PDU, cluster, and data center. At the server level, we model power using the linear model shown in equation 7 similar to [15, 41]. Actual utilization is captured as the number of active CPUs divided by the maximum number of CPUs per server. We measure the idle power of a Sun Fire X4270 server with a Hioki powermeter as 175W and the peak while fully utilized as 350W. Hence, P_{IDLE} equals 175W and $P_{DYNAMIC}$, the dynamic power range, equals 175W. The changes in terms of instant power at the server level propagate to higher levels of the power hierarchy.

$$P = P_{IDLE} + cpu\_utilization \times P_{DYNAMIC} \tag{7}$$

For our results, we assume a distributed UPS with LFP batteries attached to each server, provisioned at 40Ah, the maximum capacity that fits within the server size constraints. We compare against a centralized UPS with LA batteries. The LA battery provisioning is performed similarly to [21] to sustain the whole computing load for 48 minutes. Table 4 summarizes the characteristics of the batteries we use in our results section. To properly capture Peukert's effect during discharge, we account for changes in current every time the power draw on an individual server changes. The recharge time depends linearly with the DoD when it starts charging.

6.2 Workloads

Table 5 presents the parameters of the workloads we use in our simulator. We assume a mix of web search, social networking, and MapReduce background computation. To capture the dynamic behavior of our workloads throughout the day, we use the Google Transparency Report [17] and scale interarrival time accordingly. We collect the traffic data for a whole year (10/1/2010-9/30/2011) for two google products in the United States. Google unencrypted search represents search traffic, and Orkut represents social networking traffic (similar to Facebook). MapReduce is a company internal product and, as such, does not appear in the Transparency report. Instead, we reproduce the weekly waveform that appears in figure 3 of [7] and repeat it over the period of a year.

We model a data center which serves all three types of workloads, with relative total demand placed on the servers in the ratios shown in Table 6. The relative loads of search vs Facebook/Orkut is chosen to match worldwide demand as reported by www.alexa.com [1]. Note that we use Orkut data to define the shape of the social networking demand curve,

Workload	Cite used	Relative Normalized Traffic
Search	www.google.com	29.2%
Social Networking	www.facebook.com	55.8%
Map Reduce	-	15%

Table 6: Relative traffic numbers as obtained from [1]. Map reduce jobs is assumed 15%.

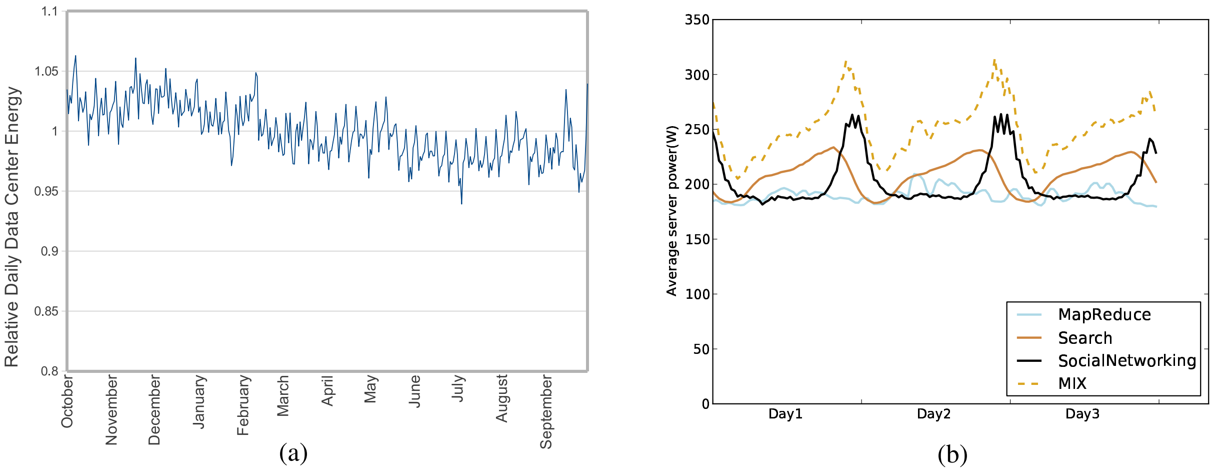


Figure 9: On the left we see the variation of data center energy throughout the year. During weekends and the summer traffic is lower. The average energy corresponds to 240.1W per server or utilization of 37.1%. On the right we zoom on the three days with higher energy requirements (11/17/2010-11/19/2010). The average power for these days is 250.5W and the corresponding utilization 43%.

but use Facebook data to gauge the magnitude of the load. The daily peak of the mix load is set to 80% of the data center peak computational capability. This number leaves sufficient computing margin to ensure stability of the system, and is consistent with published data center design goals, as shown in figure 1 in [31]. Note that because of this restriction the peak observed value of the average server power, 315W, is less than than the peak achievable power of 350W.

Figure 9(a) shows the day-to-day variation of the daily data center energy. The yearly daily average corresponds to 240.1W per server and varies moderately throughout the year. Weekends and summer months demonstrate lower traffic. We test our policies on the three consecutive days with the highest demand in energy. Graph 9(b) zooms on these days (11/17/2010-11/19/2010) and presents the daily power profile for each workload separately, as well as the combined mix. The peaks of Search and Facebook are adjacent resulting in a waveform with broader peak. Map reduce traffic increases the variance of the graph.

We evaluate the workload mix in figure 9(b) under two different web service allocations: 1) restricting each service to its own dedicated set of servers (split case), 2) co-locating all web services, with highest priority for search, lower for social networking and lowest for map-reduce jobs (mixed case).

Additionally, we emulate the scheduling of jobs across individual servers. Specifically, we consider a simple round-robin scheduling policy, similar to the operation of a naive web traffic load balancer, and a load-aware policy with knowledge of server CPU utilization. This scheduler is responsible for allocating the work among servers and is independent from the per-server scheduler that maps jobs to specific cores. In our simulated data center, the load-aware policy is extremely effective at distributing the load evenly, probably unrealistically so. Thus, the round-robin scheduler represents a more uneven distribution of work. A deployed load-aware scheduler probably falls somewhere between the two.

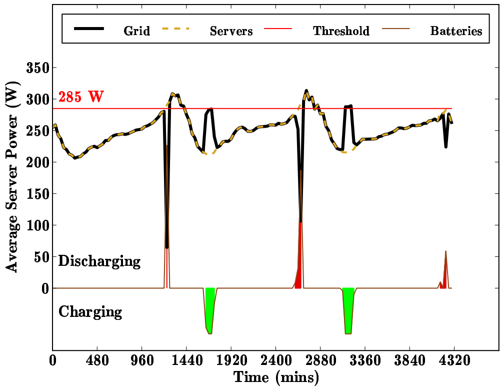


Figure 10: Centr – This plot shows the average power each server generates for the centralized-UPS configuration, as well as the contribution of the average battery during discharge and charge, and the average power observed from the utility grid. Grid power is equivalent to server minus battery power.

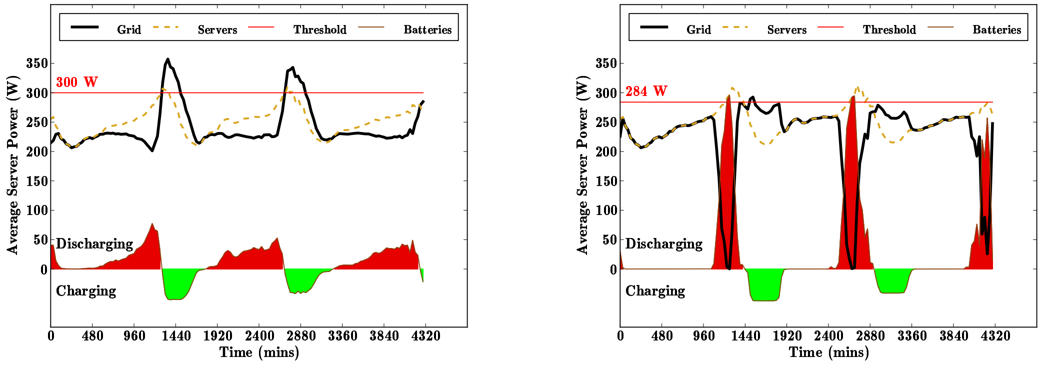
7 Results

The capacity of the battery, as well as the targeted DoD level, place an upper limit on the power capping that is possible for a given traffic pattern. In practice, though, the max achievable power capping also depends on the effectiveness of the policies that control the batteries. Setting the power capping threshold aggressively creates lower margins for wasted energy in our solution. There are two sources of battery energy waste: spatial and temporal. Spatial waste enables more batteries than necessary to shave a portion of overall power, while temporal waste enables batteries when capping is not required.

In this section, we gradually lower the peak power threshold until a policy begins to violate it. We show results for the lowest threshold (per server) that succeeds (red line in figures 10, 11, 12,13,14). Thus, we can compare policies based on that threshold. Some policies are not effective enough to cap power over a reasonable range. For those we give examples to illustrate why they fail. On an average day, it is to be expected that conservative estimates of peak power will result in a decent margin between battery capacity and the shaved peak load (some days the batteries may not be used at all). However, because we are modeling the worst days of the year, it is reasonable to expect that the available battery capacity is fully utilized. This methodology is reflective of what would happen on those days.

Figure 10 highlights the limitations of a centralized approach to peak power shaving in the absence of support for rapid duty-cycling between grid and battery supply. The UPS discharges rapidly when supporting the entire data center power during peak, rather than just the portion above the peak power goal. Once the battery reaches the DoD goal of 40%, in the middle of the peak demand, the cluster switches back to the grid and a power spike is observed. This behavior is consistent across all three days (see minutes 1250-1350, 2550-2650, 4250-4350). The other interesting phenomenon occurs during battery recharge (see minutes 1600-1850, 3000-3250). We assume in all experiments that we cannot interrupt the recharge of a battery – LA battery recharge guidelines recommend a multi-stage recharging process for best battery operation [46], which requires uninterrupted sequencing. As a result, small power spikes during the recharge time may lead to a peak power budget violation because we cannot stop the recharge.

The *ServCtrl* policy (figure 11) assumes distributed, per-server batteries and does not require any centralized coordina-



(a) Round-robin

(b) Balanced

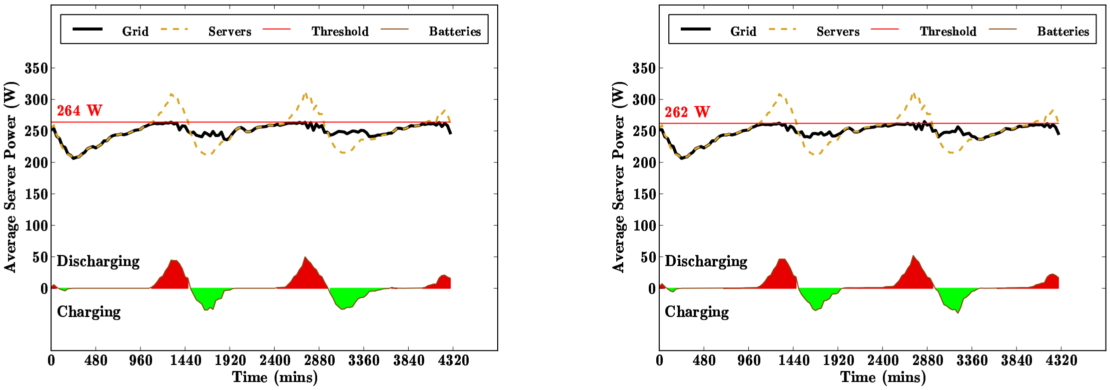
Figure 11: ServCtrl – Power distribution of the local controller policy, for round-robin and load-aware scheduling scenarios.

tion. It relies completely on local server information. It is easy to implement, but due to the lack of coordination this scheme does not make efficient use of battery stored energy. Specifically, *ServCtrl* introduces temporal energy waste when transient effects create imbalances in the load across servers, resulting in battery discharge even if the total data center power does not exceed the threshold, leaving fewer batteries available to hide the real peak. We can even have batteries recharging during the peak. In the round-robin case, we cannot effectively shave peak for any meaningful power threshold,

When very effective load-balancing is in place, we see fewer instances of unnecessary discharge, but we observe a new problem. Once traffic increases to the degree that the power of each server crosses the threshold, all batteries begin discharging. As a result, a power dip follows. This effect is clearly visible in figure 11(b). Because the batteries reduce overall datacenter power well below the threshold, we again are unable to ride out a long peak, similar to the centralized approach. Additionally, the lack of coordination of recharge cycles mimics the centralized architecture problem – a small jump in load during the recharge cycle takes grid power beyond the desired threshold of 284W in figure 11(b).

There is a trade-off between recharge time and recharge current. Large values for recharge current (power) reduce recharge time but make it harder to find the necessary margin to initiate a recharge without violating the power budget. On the other hand, small values of recharge current provide ample margin for batteries to recharge, but risk having the battery charging when the next peak arrives. For the *ServCtrl* policy we use a small recharge current of 3.7A (0.1C) that corresponds to a charge time of 10h. In the coordinated policies, *PduCtrl* and *ClustCtrl*, the controller initiates the recharge of each server battery. It is much easier to find sufficient power slack to recharge a battery without violating the PDU or the cluster power budget respectively. For these policies, we use a high recharge current of 18.5A(0.5C) that corresponds to a charge time of 2h.

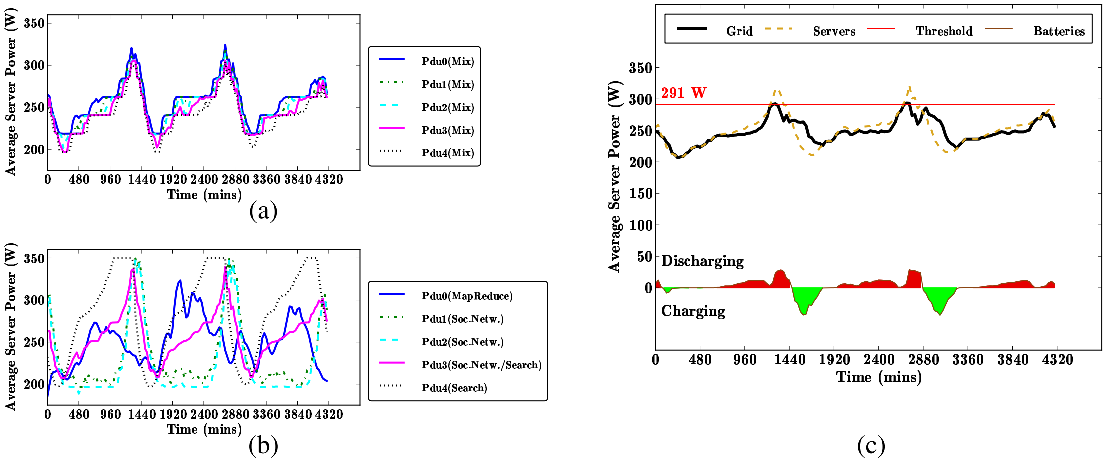
Figure 12 shows that the *PduCtrl* policy performs much better than *ServCtrl*, maintaining a power threshold of 264W for round-robin and 262W for the load-aware scheduler. This is the result of coordination among batteries to achieve a local cap at the PDU level. Just enough batteries in each PDU region are activated to reduce power below the threshold, thus preserving other batteries to ride out the full peak. Battery recharge is similarly coordinated so that no more than the available spare power is used for recharge. Global imbalances in the loads seen by each PDU result in slight noise in the total power; however, because each PDU is enforcing its threshold, that noise only results in grid power varying a little



(a) Round-robin

(b) Balanced

Figure 12: PduCtrl This plot shows power when we apply separate controllers for every PDU of the system, for round-robin and load-aware scheduling scenarios.



(a)

(b)

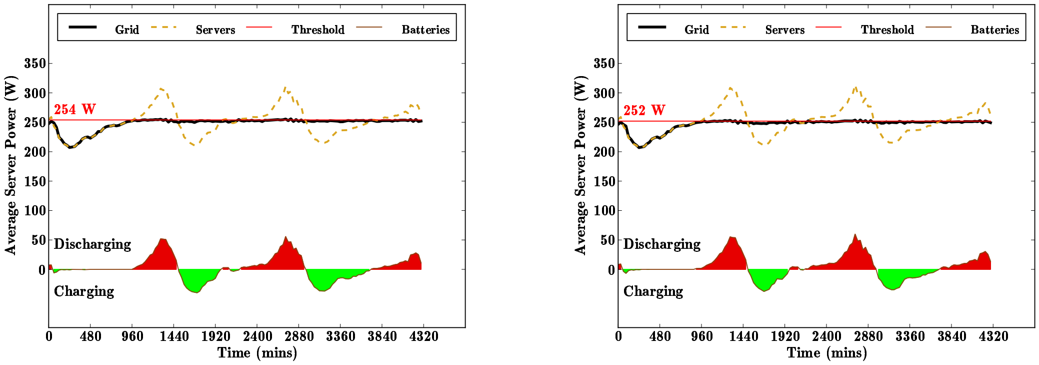
(c)

Figure 13: PduCtrl Split This plot shows the effect of segmenting the three webservices into predefined PDUs. In (a) we show the server average power per PDU (without batteries) for the mixed case. In (b) we show the split case. When webservices run on split servers there are fewer available batteries to deal with a power peak. This is why in (c), when we use the batteries we can only guarantee peak power of 291W.

below the threshold.

That result holds when all three services run on all PDUs, because each PDU sees a similar power profile. For the *PduCtrl* we also study the scenario where each service is allowed to run on a subset of the PDUs. In this case, batteries are statically partitioned. As a result, search batteries are not available to help with the Facebook peak, and vice versa. Globally, we have batteries charging and discharging at the same time, which is clearly suboptimal. The lowest power budget that we can enforce in the split case is 291W (figure 13). This analysis motivates resource sharing among applications, despite the associated complexity for fairness and quality of service.

Figure 14 shows the *ClustCtrl* policy applied on the mixed scenario for the round-robin balancing and the load-aware balancing. The lowest power cap for this policy is 254W and 252W for the two cases. Note that both of these results are very close to the ideal scenario which would reduce power to 250W (average power for the worst day). This increased efficiency is a direct result of being able to take a more global view of power. Imbalances between the PDUs no longer



(a) Round-robin

(b) Balanced

Figure 14: ClustCtrl Power distribution for the global (cluster-level) policy, again for both the round-robin and balanced scheduling cases.

result in undershooting the power threshold, allowing us to preserve batteries that much longer.

Results for the *ClustCtrl* policy with the partitioned workload are identical to figure 14. While we assume the same distribution of workload to PDUs as in the previous results, at the cluster level it appears as a mixed workload and the controller has no difficulty in adapting to the load appropriately. With cluster-level power control, then, we are able to discharge batteries on PDUs that are not experiencing a local peak, and target only data-center power peaks.

There are many considerations that might determine the right level to apply our battery control policies. Our results show that the policy becomes most effective as we move up the power hierarchy. Most importantly, the policy should be applied no lower than the level at which the component workloads of the datacenter are mixed together. These results indicate that with properly sized batteries and an effective control policy, we can do much more than shave the extreme peaks of the load – in fact, we almost completely flatten the power profile very close to the average power. Capping peak power from 315W to 254W corresponds to a reduction of 19.4%. This reduction will allow 24% more servers within the same provisioned power and reduce TCO/server by 6.5% (see section 3), resulting in more than $20M savings in costs for a data center with 28,000 servers over its lifetime.

7.1 Guard band and DVFS – when projections fail

Prior work on power capping either applied performance degrading techniques, like DVFS, at peak load, or fall back to it as a failsafe when the batteries fail [21]. However, applying techniques such as DVFS at peak load is often an unacceptable option. Many datacenter applications track performance by watching the tail of the performance distribution – e.g., 95th percentile response time. Applying DVFS at peak load, even for a short time, can have a disastrous effect on those metrics, DVFS not only extends latencies, but also reduces throughput and induces higher queuing delays. Reducing performance at peak load increases the response time of those jobs already at the tail of the distribution.

Our technique does not apply DVFS, even when the peak power exceeds our conservative estimates, nor do we give up and allow the grid power to increase. In all the previous algorithms we disable the batteries once we hit the DoD goal. This design decision is driven by the fact that TCO/server savings presents a maximum at the DoD goal (Section 4). However,

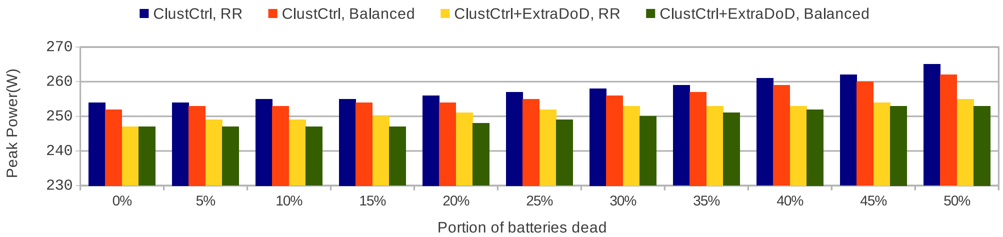


Figure 15: As the number of unusual batteries increase, the lowest possible peak power increases. Allowing to exceed our DoD goal occasionally, permits even higher peak power reduction. Load imbalances discharge batteries at different rates and make power capping harder.

another benefit of the high DoD limit is additional stored energy in our batteries that can be used in case of emergency. With LFP per-server batteries there is approximately 35% guard band before we are in danger of not having sufficient reserves to survive a grid failure. This guard band can be used during days where the power profile exceeds worst-case peak power estimates. Our projections for the optimal DoD level were based on daily discharge; however, going below 40% to say, 35% or 30%, a couple times a year, or even once a month, will have no significant impact on the lifetime of the battery. Thus, we never need to apply performance-reducing mechanisms at peak load unless our estimated peak power is off by enormous margins. That does not mean that DVFS cannot still be an effective power management technique. In fact, DVFS can be applied completely orthogonally to our approach and when combined demonstrate interesting synergies. During low utilization, battery-based peak power capping increases power off-peak by requiring battery recharging. We can apply DVFS to reduce power at low demand, without impacting service-level agreements, creating more margin for recharging and accelerating the recharge cycle. This technique is not employed in the results shown in this paper, but would allow us to shave even more power with minimal performance impact.

7.2 Failure analysis

In large data centers it is common to cluster maintenance operations to reduce costs. This means that a non-negligible portion of batteries may be unusable for peak power shaving purposes before these batteries get replaced. Figure 15 shows how the lowest achievable peak changes when we assume that a portion of batteries has failed. We compare the best policy *ClustCtrl* with and without the use of the additional energy provided by discharging our batteries beyond the DoD goal. The peak threshold gradually increases with a larger portion of dead batteries. However, the increase is relatively small. Even when half of the batteries are dead we can still shave 16% of peak power. For these experiments, we find that we do not need to modify the algorithm of the controller to handle the unusable components. The controller signals a faulty component to start discharging, but no decrease in power takes place. As a result, the controller signals additional batteries in the next round and eventually corrects for the failure without any direct feedback. We also observe that the additional energy from deeper discharge of the batteries allows us to shave more power with fewer batteries. However, if the datacenter is allowed to enter deeper discharge while dead batteries stack up for an extended period, then it can have an impact on the battery lifetime. With the extra energy and small portion of dead batteries, we can achieve a peak value that is lower than the average power of 250W in the 3-day period we study. Clearly this is not sustainable for longer time periods, because the batteries do not have the opportunity to fully recharge in anticipation of the next day peak.

7.3 Energy proportionality

The server used for this study is a representative modern platform, with idle power close to 50% of peak, based on our measurements. In the future, servers are expected to become increasingly energy proportional. Here we model the impact of such a server that is completely energy proportional and study how it impacts the effectiveness of our techniques. We assume the same workload and peak power.

Energy proportional servers impact the power demand in two interesting ways. They increase the height of the peak, relative to the average power, since power is significantly lower during off-peak periods. Peaks are also thinner – as long as demand is not at 100% of the peak, which represents most, if not all, of the peak regions. Sharper peaks make our techniques more effective. Consequently, we can further reduce the power threshold. Our simulations indicate the ability to reduce the peak observed power from 280W to 175W, a reduction of 37.5%. That results in an increase in server capacity of 60%.

8 Related Work

Peak Power Provisioning and Capping: Reducing power consumption in server clusters is a well-studied problem in the literature [45, 36, 30, 20]. The overall idea is to combine CPU throttling, dynamic voltage/frequency scaling (DVFS), and switching entire servers on/off depending on the workload. Raghavendra, et al. [45] note that more efficient power management solutions are possible by managing power at the rack (or ensemble) level than at individual blades. They devise proactive and reactive policies based on DVFS to cap power budgets at the rack level. Nathuji and Schwan [36] introduce the notion of power tokens to deal with heterogeneity across hardware platforms. Govindan, et al. [20] combine applications with heterogeneous characteristics in terms of burstiness. As a result, the power budget is exceeded statistically infrequently. DVFS is used as a failsafe mechanism to prevent against lasting peak power violations.

Femal et al. [16] were among the first to use formal control theory to maximize throughput while capping power. Raghavendra, et al. [41] extend the control theory idea to present a hierarchy of coordinated controllers that cap power across different levels of the power hierarchy and minimize performance impact. They argue for nesting controllers that operate at different time granularities to ensure stability and emphasize the information flow between the controllers.

Power Modeling: Several studies model the power consumption of single nodes using different predictors and modeling techniques [33, 16, 20, 8, 26, 27, 28, 32, 44, 48]. Some works attempt to predict peak power consumption based on processor utilization alone [32, 15], while others rely on board-level power measurements [28] and performance counter statistics [12, 20, 48]. The power modeling techniques also vary in complexity, including simple lookup-based models [44], linear regression [26], and advanced AI techniques such as Gaussian mixture models [12].

Virtualization: Peak power management strategies developed to reduce server power cannot always be directly applied in modern virtualized data centers. This is because physical machine power decisions affect multiple co-scheduled virtual machines at the same time, with potentially different performance impact per virtual machine. Wang, et al. [53] propose control strategies to power on and power off components in order to investigate the energy savings when per-virtual-machine performance goals are set. Nathuji, et al. [35] treat virtual machine level power decisions as hints passed to the hypervisor which is responsible for applying the correct universal power budgeting decisions. Recent work also proposes performing per-VM energy accounting [11, 29]. For our experiments we consider large scale distributed applications, co-existing on the servers. The service consolidation requires virtualization for fault isolation and ease of administration.

Using batteries in data centers: Battery power management has been well studied in the embedded/mobile system domain with various works proposing techniques to adjust the drain rate of batteries in order to elongate the system operation time [37, 43, 42, 47]. Prior research has also investigated analytical models for battery capacity and voltage in portable devices [37, 47, 25]. In the most closely related work, Govindan, et al [21] introduce the idea of reducing data center peak power by leveraging the stored energy in a centralized UPS. During peak load, power from the UPS batteries augments the main grid, effectively hiding the peak from the utility service. During low load, the batteries recharge, consuming additional power. In our work, we explore how this idea applies on distributed UPS topologies, a less over-provisioned and hence more challenging environment. For our analysis, we utilize much more realistic test cases and do not assume oracle workload knowledge. We also present results for a complete data center cluster, rather than just a few machines. In a separate, recently published work [22], the same authors also argue towards a distributed UPS solution from a cost and reliability perspective. They find that a hybrid distributed UPS placement, at PDU and server level, yields the most promising topology. They do not consider battery energy for peak power capping in that work, but this finding provides additional motivation for our work on the use of distributed batteries for power capping.

9 Conclusions

State-of-the-art data centers such as Google's and Facebook's have adopted distributed UPS topology in response to the high cost associated with a centralized UPS design. In this work we explore the potential of using battery-stored energy in a distributed UPS topology to shave peak power. We describe how to provision the capacity of the battery and elaborate on how recharge cycles, the depth of discharge, and the workload power profile affect the potential for peak power shaving. We leverage the distributed nature of the batteries and design a controller to use them only when needed and thus prolong the duration of their usage, without violating the targeted power budget. Significant peak power reductions of up to 19.4%, are possible with our technique. These reductions allow us to provision more servers under the same power budget and reduce the TCO per server by 6.5%, significantly increasing the computation that can be done per facility and saving millions of dollars per datacenter.

References

[1] Alexa. the web information company, traffic metrics, search analytics, demographics for websites. `http:/www.alexa.com`, 2008.

[2] D. Anderson. An evaluation of current and future costs for lithium-ion batteries for use in electrified vehicle powertrains. In *Master's Thesis, Duke University*, 2009.

[3] anon. In search of the perfect battery. *The Economist*, Mar. 2008.

[4] Apache. *http://incubator.apache.org/olio/*.

[5] APC. InfraStruxure Total Cost of Ownership, Infrastructure cost report. `http://www.apc.com/tools/isx/tco/`, 2008.

[6] D. E. Carolinas. Utility bill tariff. `http://www.duke-energy.com/pdfs/scscheduleopt.pdf`, 2009.

[7] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz. The case for evaluating MapReduce performance using workload suites. In *Technical Report No. UCB/EECS-2011-21*, 2011.

[8] J. Choi, S. Govindan, B. Urgaonkar, and A. Sivasubramaniam. Profiling, prediction, and capping of power consumption in consolidated environments. In *MASCOTS*, pages 3–12, 2008.

[9] C. S. Computing. Power supply efficiency specifications. http://www.climatesaverscomputing.org/resources/certification, 2011.

[10] A. P. P. Corp. Portable Power Product design, assemble and quality control. http://www.batteryspace.com/lifepo4cellspacks.aspx, 2000.

[11] G. Dhiman, G. Marchetti, and T. Rosing. vgreen: A system for energy-efficient management of virtual machines. *ACM Trans. Design Autom. Electr. Syst.*, 2010.

[12] G. Dhiman, K. Mihic, and T. Rosing. A system for online power prediction in virtualized environments using gaussian mixture models. In *DAC '10: Proceedings of the 47th Design Automation Conference*, pages 807–812, New York, NY, USA, 2010. ACM.

[13] I. Dupont Fabros Technology. Sec filing (s-11) 333-145294, August 9, 2007.

[14] Facebook. Hacking conventional computing infrastructure. http://opencompute.org/, 2011.

[15] X. Fan, W.-D. Weber, and L. A. Barroso. Power provisioning for a warehouse-sized computer. In *International Symposium on Computer Architecture*, June 2007.

[16] M. E. Femal and V. W. Freeh. Boosting data center performance through non-uniform power allocation. In *ICAC*, pages 250–261, 2005.

[17] Google. *http://www.google.com/transparencyreport/traffic/.*

[18] Google. Google Summit. http://www.google.com/corporate/datacenter/events/dc-summit-2009.html, 2009.

[19] Google. Best data center efficiency practices: Adjust the thermostat. http://www.google.com/corporate/datacenter/best-practices.html, 2011.

[20] S. Govindan, J. Choi, B. Urgaonkar, A. Sivasubramaniam, and A. Baldini. Statistical profiling-based techniques for effective power provisioning in data centers. In *EuroSys*, Apr. 2009.

[21] S. Govindan, A. Sivasubramaniam, and B. Urgaonkar. Benefits and limitations of tapping into stored energy for datacenters. In *International Symposium on Computer Architecture*, June 2011.

[22] S. Govindan, D. Wang, L. Chen, A. Sivasubramaniam, and A. Sivasubramaniam. Towards realizing a low cost and highly available datacenter power infrastructure. In *HotPower*, 2011.

[23] F. Harvey. Table with Peukert's exponent for different battery models. http://www.electricmotorsport.com/store/ems_ev_parts_batteries.php, 2001.

[24] U. Hoelzle and L. A. Barroso. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan and Claypool Publishers, 2009.

[25] M. Jongerden and B. Haverkort. Battery modeling. In *Technical Report, TR-CTIT-08-01, Centre for Telematics and Information Technology*, 2008.

[26] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya. Virtual machine power metering and provisioning. In *1st ACM symposium on Cloud computing*, 2010.

[27] R. Koller, A. Verma, and A. Neogi. WattApp: an application aware power meter for shared data centers. In *7th international conference on Autonomic computing*, 2010.

[28] A. Lewis, J. Simon, , and N.-F. Tzeng. Chaotic attractor prediction for server run-time energy consumption. In *HotPower*, 2010.

[29] H. Lim, A. Kansal, and J. Liu. Power budgeting for virtualized data centers. In *USENIX Annual Technical Conference*, 2011.

[30] D. Meisner, B. Gold, and W. Thomas. Powernap: Eliminating server idle power. In *14th international conference on Architectural support for programming languages and operating systems*, 2009.

[31] D. Meisner, C. M. Sadler, L. A. Barroso, W.-D. Weber, and T. F. Wenisch. Power management of online data-intensive services. In *International Symposium on Computer Architecture*, pages 319–330, 2011.

[32] D. Meisner and T. F. Wenisch. Peak power modeling for data center servers with switched-mode power supplies. In *international symposium on Low power electronics and design*, pages 319–324, 2010.

[33] A. Merkel, J. Stoess, and F. Bellosa. Resource-conscious scheduling for energy efficiency on multicore processors. In *5th European conference on Computer systems*, 2010.

[34] E. motor sport. EV construction, thundersky batteries. http://www.electricmotorsport.com/store/ems_ev_parts_batteries.php, 2001.

[35] R. Nathuji and K. Schwan. Virtualpower: coordinated power management in virtualized enterprise systems. In *twenty-first ACM SIGOPS symposium on Operating systems principles*, 2007.

[36] R. Nathuji and K. Schwan. Vpm tokens: virtual machine-aware power budgeting in datacenters. In *17th international symposium on High performance distributed computing*, 2008.

[37] D. Panigrahi, S. Dey, R. R. Rao, K. Lahiri, C.-F. Chiasserini, and A. Raghunathan. Battery life estimation of mobile embedded systems. In *VLSI Design*, pages 57–63, 2001.

[38] S. Pelley, D. Meisner, P. Zandevakili, T. F. Wenisch, and J. Underwood. Power routing: dynamic power provisioning in the data center. In *Architectural support for programming languages and operating systems*, 2010.

[39] E. N. Power. *National Survey on Data Center Outages*, 2010.

[40] S. press release. Savvis sells asserts related to two datacenters for $200 million, June 29 2007.

[41] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu. No ”power” struggles: coordinated multi-level power management for the data center. In *13th international conference on Architectural support for programming languages and operating systems*, 2008.

[42] D. N. Rakhmatov. Battery voltage prediction for portable systems. In *International Symposium on Circuits and Systems*, 2005.

[43] D. N. Rakhmatov and S. B. K. Vrudhula. Energy management for battery-powered embedded systems. *ACM Trans. Embedded Comput. Syst.*, 2(3):277–324, 2003.

[44] P. Ranganathan and P. Leech. Simulating complex enterprise workloads using utilization traces. In *Workshop on Computer Architecture Evaluation using Commercial Workloads*, 2007.

[45] P. Ranganathan, P. Leech, D. Irwin, and J. Chase. Ensemble-level power management for dense blade servers. In *International Symposium on Computer Architecture*, June 2006.

[46] T. B. Reddy and D. Linden. *Linden's Handbook of Batteries (4th edition)*. McGraw-Hill, 2011.

[47] P. Rong and M. Pedram. An analytical model for predicting the remaining battery capacity of lithium-ion batteries. In *Design, Automation Test in Europe*, 2003.

[48] K. Singh, M. Bhadauria, and S. A. McKee. Real time power estimation and thread scheduling via performance counters. In *dasCMP: Workshop on Design, Architecture and Simulation of Chip Multi-Processors*, 2008.

[49] SmartGauge. Peukert's law equation and its explanation. http://www.smartgauge.co.uk/peukert.html, 2011.

[50] M. Swierczynski, R. Teodorescu, and P. Rodriguez. Lifetime investigations of a lithium iron phosphate (LFP) battery system connected to a wind turbine for forecast improvement and output power gradient reduction. In *BatCon*, 2008.

[51] W. P. Turner and K. G. Brill. Cost Model: Dollars per kW plus Dollars per Square Floor of Computer Floor, 2009.

[52] B. University. Online university education about batteries. http://batteryuniversity.com/, 2003.

[53] X. Wang, M. Chen, C. Lefurgy, and T. W. Keller. SHIP: Scalable hierarchical power control for large-scale data centers. In *Parallel Architecture and Compilation Techniques*, 2009.

[54] Windsun. Lead-acid batteries: Lifetime vs Depth of discharge. http://www.windsun.com/Batteries/Battery_FAQ.htm, 2009.

[55] L.-L. Zhang, G. Liang, A. Ignatov, M. C. Croft, X.-Q. Xiong, I.-M. Hung, Y.-H. Huang, X.-L. Hu, W.-X. Zhang, and Y.-L. Peng. Effect of Vanadium Incorporation on Electrochemical Performance of LiFePO4 for Lithium-Ion Batteries. In *Journal of Physical Chemistry*, June 2011.