

Power Provisioning for Diverse Datacenter Workloads

Christopher Stewart and Jing Li
The Ohio State University

Abstract *The workloads hosted in a datacenter share more than just servers; they also share electrical circuits. Datacenter managers provision the power capacity of these circuits to hosted workloads, often based on peak power needs. In this work, we studied the typical (mode) and peak power needs of 3 real datacenters, using 1) data from hardware manufacturers and 2) past traces of observed power needs. We found that typical power needs were nonmonotonic relative to peak needs. That is, some applications with low typical power needs had large peak needs—stemming from the diverse power utilization of datacenter workloads. Such diversity caused surprising order inversions where workloads with relatively small peak power had relatively large typical needs. Based on these results, we propose a power provisioning approach that considers power-utilization diversity. Our approach provides 1) predictable monotonic results as power capacity increases and 2) performs better than approaches commonly used in practice.*

I. INTRODUCTION

Datacenters (sometimes written data centers) host a wide range of networked applications from e-commerce and enterprise services to scientific computing [3], [15]. Viewed as very big computers, datacenters comprise more than just networked servers, they also include hardware for power delivery. This hardware supplies electric circuits for all hosted applications, supporting power workloads from many applications and heterogeneous hardware [13]. These circuits “break” when their supported power workload exceeds preset limits, leading to performance capping [4], [5], [8], [16], costly electrical upgrades [6], or brownouts [8], [18]. Power provisioning in a datacenter tries to map applications to circuits without exceeding capacity limits.

At first glance, the power-provisioning problem fits the following integer programming model: measure each workload’s power needs, use circuit capacity limits as a constraint, then find a mapping that uses power capacity well. However, in practice, a workload’s power needs can increase over time. Workload-to-circuit mappings based on a snapshot where current needs underestimate future needs risks circuit breaks. To reduce these

risks, datacenter managers normally provision based on a workload’s estimated peak power—not the typical power needs observed during normal operation. Nameplate ratings provided by hardware manufacturers are widely used to estimate peak power [7]. They are often discounted (by up to 60% [7], [8]) to reflect peaks that can be reached with real-world workloads. Recently, researchers have proposed that the measured peak power should be used instead [2], [8], [10]. Compared to nameplate ratings, the measured peak power requires that a workload’s power needs can be measured but provides tighter workload-aware estimates.

We studied the typical, measured peak, and nameplate power of 3 real datacenters and used the results to design a new power provisioning approach. By typical power, we mean the statistical mode observed during a particular time of day. In the context of our study, a power workload reflects the combined needs of a cluster of servers. Specifically, we measured power workloads at rack-level power distribution units (PDU). The datacenters that we worked with are described below:

Codename OSU contains 1300 servers (about 6500 cores) with 400kW maximum power capacity. This datacenter is open to the public, allowing customers to supply their own hardware on leased space or to supply virtual machine images. There are 300 PDU that connect to datacenter circuits. Hosted applications range from data mining and research (e.g., bio-medical) to enterprise services (e.g., PeopleSoft) to virtual learning (e.g., Blackboard).

Codename CSE: contains 165 servers (about 815 cores) with 40kW maximum power capacity. This datacenter hosts the research workload for our computer science department. Faculty members purchase their own hardware. There are 35 PDU.

Codename PROD: contains 200 servers (about 980 cores) with 80kW maximum power capacity, hosting production enterprise and academic services and storing sensitive student files for the School of Engineering. There are 62 PDU.

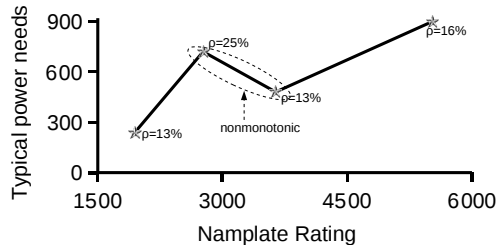


Fig. 1: Nonmonotonic nameplate ratings in OSU. The x-axis shows 4 PDU sorted by nameplate rating. ρ shows the power utilization of each (i.e., power needs divided by estimated peak power).

Our study exposed a key aspect of power workloads in the datacenter: *Typical power needs were nonmonotonic relative to peak needs*. For example, Figure 1 shows an workload that had low typical power needs even though it ran on large-nameplate hardware. Such nonmonotonicity produces order inversions, i.e., situations where peak needs order workloads differently than typical needs. When two workloads are inverted, the workload with larger peak power needs will require more provisioned capacity even though it uses less typical power. We found that up to 38% and 28% of PDU pairs in our studied datacenters were inverted based on 1) nameplate ratings and 2) measured peak power respectively. The workload with larger peak power required up to 125% more capacity. Inversions reflect diverse power utilization, also shown in Figure 1. We found that the power-utilization distributions across a whole datacenter likely reflect the mixture of many types of hosted workloads.

Next, we considered a common power provisioning problem: Assign workloads to a circuit such that the circuit’s typical power draw is maximized. Nonmonotonic peak power makes it hard to predict the typical power draw of workloads assigned to a circuit. Assignments with relatively large peak needs can fall well below the median in terms of their typical power needs. Integer programming based on peak power often finds assignments that fall victim to such inversion. We studied provisioning approaches that lessen the impact of inversions while still using peak power as a proxy for typical power needs. A simple approach, *smallest peak power first (SPPF)*, totally avoids inversions, producing assignments that increase the typical power draw as circuit capacity increases. However, SPPF assignments can have relatively low typical power needs when workloads with large peak power also have large power utilization. We propose a diversity-aware provisioning approach that chooses between the assignments from SPPF and integer programming based on peak power needs and the impact of inversions. The evaluation of

our approach is ongoing, but preliminary results are positive.

The remainder of this paper is as follows. Section II presents our study of typical, measured peak, and nameplate power across whole datacenters. Section III makes the case for diversity-aware power provisioning. Section V discusses our future work.

II. NONMONOTONIC PEAK POWER

Two workloads i and k are order inverted if:

$$P_t(i) < P_t(k) \quad \text{and} \quad P_{nr}(i) > P_{nr}(k) \quad (1)$$

$$\text{or } P_t(i) < P_t(k) \quad \text{and} \quad P_{mp}(i) > P_{mp}(k) \quad (2)$$

The P function captures power workload. Parameters t , nr , and mp stand for typical needs, nameplate rating, and measured peak respectively. When a datacenter has many inverted workloads, we say that peak power is nonmonotonic relative to typical power.

Figure 2 plots PDU pairs across our 3 studied datacenters, marking the inverted pairs. In this figure, we used nameplate rating (P_{nr}) to describe a workload’s peak power. These ratings were compared with typical power needs observed between 7–11am. We found that 27%, 38%, and 23% of PDU pairs were inverted for OSU, CSE, and PROD respectively. Consider the problem of trying to maximize the typical power draw on a circuit and the widely used approach of provisioning based on peak power. The difference between the peak needs of inverted PDU reflects the potential for lost circuit capacity. This is shown on the Y axis. In the average inverted PDU pair, the PDU with larger peak needs would have over provisioned 125%, 80%, and 47% more capacity than the PDU with larger typical needs.

Impact of the Time of the Day: Next, we asked, “Do workload inversions occur only at certain times of the day?” We took snapshots of each datacenter in the morning, afternoon, and evening and on the weekends. Each snapshot provided typical power needs for counting the number of inverted PDU pairs at that time of day. To measure stability, we computed the coefficient of variation ($\frac{\sigma}{\mu}$), a widely used normalized measure of dispersion. A common rule of thumb is a coefficient of variation below 100% indicates stable, low-variance distributions [1]. We observed coefficient of variation for OSU, CSE, and PROD at only 10%, 0.2%, and 1.6%. The number of inverted PDU pairs was stable at all times of the day. We also noted that OSU and PROD, datacenters that host enterprise and web workloads affected by social patterns, had greater variation than CSE.

We also studied turnover among inverted PDU pairs, asking “are the inverted PDU pairs in the morning

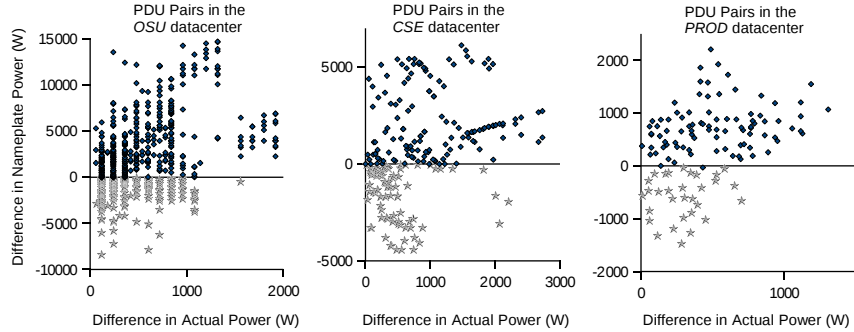


Fig. 2: Nonmonotonic peak power and order inversions. In this figure, peak power is a PDU’s nameplate rating. Each point reflects one pair of PDU. All possible pairs are shown. The X axis shows the difference in actual power needs between two PDU in a pair, (i.e., $P_t(i) - P_t(k)$). The Y axis shows the difference in nameplate ratings ($P_{np}(i) - P_{np}(k)$). Stars mark PDU pairs that are inverted.

the same as the inverted pairs at night?” We created a unique ID for each PDU pair in our study. We performed set logic on the IDs of the inverted pairs from the morning, afternoon, evening, and weekend data. We say that an inverted pair persisted if it was in the intersection of inverted pairs for all times of day. 70%, 97%, and 78% of inverted pairs persisted in OSU, CSE, and PROD respectively. If we excluded weekends (i.e., inverted pairs that persisted throughout work days), the numbers rise across all studied datacenters to 80%, 100%, and 96%. Weekends affected OSU and PROD the most because of their supported workloads.

Impact of Measured Peak Power: Measured peak power tailors the estimated peak power to a workload’s observed history, providing an upper bound that is often closer to typical needs than nameplate ratings [7], [8]. It is quickly becoming the preferred approach to estimate peak power needs [10], [14]. Figure 2. We found that measured peak power significantly reduced the number of inverted PDU pairs for OSU and CSE. Table I shows the observed reduction in inverted PDU pair. When we looked into these results, we saw that many PDU in PROD had measured peaks that were much larger than typical needs (like nameplate ratings), explaining the increased number of inverted pairs. Table I also shows that measured peak power reduced the wasted circuit capacity caused by inverted PDU (when trying to maximize typical power on a circuit).

	OSU	CSE	PROD
% Fewer inversions	67%	84%	-19%
Average reduction in impact	52%	38%	24%

TABLE I: As a peak power estimator, measured peak normally reduces inversions and their impact compared to nameplate ratings

Even though measured peak reduced the number of inverted pairs, it did not solve the problem. OSU, CSE,

and PROD were still afflicted with inverted PDU in 8%, 6%, and 28% of their possible pairs. The average inverted pair still led to wasted capacity of 59%, 49%, and 35% respectively. Our conclusions from these results were that 1) measured peak power should be used in place of nameplate power because it normally reduces inversions and lessens their impact, and 2) additional measures are needed to deal with inversions.

Power Workload Diversity: Nonmonotonic peak power is caused by the diverse range of power workloads supported in the studied datacenters. Power utilization, defined below, provides a normalized metric for power workload.

$$\rho = \frac{P_t(i)}{P_{nr}(i)} \quad (3)$$

$$\text{or } \rho = \frac{P_t(i)}{P_{mp}(i)} \quad (4)$$

Figure 3 shows the diverse power utilization (ρ) supported across OSU. This is the source of inverted PDU pairs. When nameplate rating is the denominator, the median and 95th percentile power utilization differ by more than 68%. Using the measured peak power, the median and 95th percentile differ by 50%.

Figure 3 highlights two important differences between nameplate ratings and measured peak power. First, utilizations with nameplate ratings are generally low with a few outliers (75th percentile is only 25% utilization). Comparatively, measured peak produces large utilization. This is why many researchers have recommended that datacenter managers move to measured peak power as the base estimator for provisioning decisions. Second, we observe that both distributions contain reverse knee points, indicating an opportunity to cluster workloads according to their power utilization. Power utilization is related to device (e.g., CPU) utilization which is well known to be biased under interactive

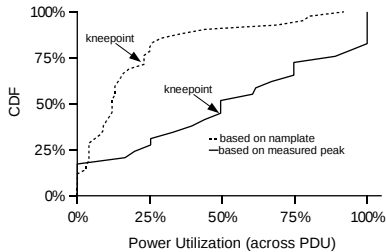


Fig. 3: Power utilization in OSU. Each point reflects a single PDU. The line based on nameplate rating shows $\frac{P_r(i)}{P_{nr}(i)}$. The line based on measured peak shows $\frac{P_r(i)}{P_{mp}(i)}$.

and throughput-oriented datacenter workloads [8], [12], [17], which could explain clustering.

III. POWER PROVISIONING

Section II showed that peak power was nonmonotonic relative to typical power needs because the studied datacenters supported diverse power workloads. For this section, we studied commonly used power provisioning approaches to understand how frequently they waste circuit capacity on inverted PDU. Each approach was used to select k workloads from N under the following rules:

1. The total peak needs of the selected k workloads can not exceed preset capacity limits.
2. The goal is for the selected k workloads to use as much typical power as possible.

These rules were chosen to help the managers of our studied datacenters support large customers supplying their own equipment. In one datacenter, managers received a new order for several PDU clusters that had peak needs greater than the typical available power on any given circuit. To make room, they had to move k workloads from circuits targeted for the new order. That is, for each targeted circuit, they wanted to migrate workloads in a way that moved as much typical power as possible within the capacity limits of other circuits. At another datacenter, managers always considered large future orders when they assigned workloads to circuits. There, managers attempted to fill the most-filled circuits with k of N new orders before placing the remaining new orders on unfilled circuits.

A. Provisioning Approaches

We studied the following approaches: integer programming, smallest peak power first, and our own approach that considers power workload diversity. Throughout this section and in our preliminary results, we refer to the PDU listed in Table II, which shows the

OSU		CSE		PROD	
Util.	Meas. Peak	Util.	Meas. Peak	Util.	Meas. Peak
4%	3119W	99%	948W	82%	917W
4%	3119W	99%	1946W	93%	1236W
13%	3823W	98%	700W	66%	1144W
9%	3823W	100%	84W	92%	762W
100%	480W	96%	240W	97%	1448W
33%	1080W	93%	1348W	98%	1440W
5%	2500W	98%	1006W	15%	1416W
4%	960W	99%	948W	91%	1157W

TABLE II: Measured peak-power needs and power utilization of 8 PDU recently added to the studied datacenters. Actual power needs were observed during the morning (7–11am).

workloads of 8 PDU recently added to each studied datacenter, including: their power utilization and measured peak power.

Integer Programming: When peak needs match typical needs, our provisioning problem is an instance of weighted knapsack, a well known integer programming problem. Weighted knapsack is NP-complete, but approximation schemes can find good nearly optimal solutions quickly and have been known for decades [11]. Unfortunately in practice, typical needs often fall below peak needs and peak needs are used in provisioning. Here, we used the following knapsack integer programming model: 1) measure peak power needs, 2) use circuit capacity as a constraint, and 3) find the assignment of peak-power needs that uses as much circuit capacity as possible.

Section II showed that peak needs are nonmonotonic relative to typical needs. Such nonmonotonic peak power means that a subset of workloads with high combined peak power may perform poorly in terms of typical power. For example, consider the integer programming approach under a circuit capacity of 960W in OSU (see Table II). The 960W and 480W PDU are inverted, but knapsack selects the PDU that uses only 38W of power typically—maximizing peak power. The better choice would be the PDU with typical needs of 480W. Note, the integer-programming approach’s choice is poor for 2 reasons: First, it does not use as much typical power as possible. Second, it uses less typical power than the choice under 959W circuit capacity. The latter point makes this approach unpredictable since capacity increases lead to choices that perform worse.

Smallest Peak Power First: Inverted PDUs only waste circuit capacity when the PDU with larger peak power is chosen instead of the PDU with larger typical needs. The smallest peak power first (SPPF) approach never makes this mistake by filling capacity in ascending order of peak power needs. PDU with larger peak needs are chosen only if the PDU with smaller peak needs are chosen also. SPPF provides predictable

```

DA_Provisioning (candidates, capacity, utilCDF) {
# candidates -> { $P_{mp}(0), \dots, P_{mp}(i)$ }
# capacity -> int  $C$ 
# utilCDF -> Hashtable(keys= $K^{th}$  percentile, val=power util.)

assignment base_solution = {};
assignment alt_solution = {};

base_solution = SPPF(candidates, capacity);
int alt_count=sumPeakNeeds(base_solution);
while (alt_count < capacity)
  alt_count++;
  alt_solution = knapsack(candidates, capacity);
  if (DA(alt_solution) > DA(base_solution))
    base_solution = alt_solution;
return base_solution;
}

float DA(assignment A) {
float tot_cost = 0;
float cert = 0.1;
forall a in A
  float max_cost = 0;
  forall c in candidates and not in A
    if ( $P_{mp}(c) * utilCDF\{1 - cert\} >$ 
         $P_{mp}(a) * utilCDF\{cert\}$ )
      if ( $P_{mp}(a) > P_{mp}(c)$ ) # Inverted PDU
        float this_cost = ( $P_{mp}(a) - P_{mp}(c)$ )
        if (this_cost > max_cost)
          max_cost = this_cost;
  tot_cost += max_cost;
return sumPeakNeeds(a) - tot_cost;
}

```

TABLE III: Pseudo-code of our provisioning approach.

monotonic results, i.e., an increase in circuit capacity never decreases the typical power draw of selected workloads. Recall from Section II, the average inversion could use 24–58% more circuit capacity than needed. Since SPPF never chooses the wrong PDU, it recovers such lost capacity that may hinder other approaches, e.g., integer programming.

SPPF performs poorly when workloads with large peak power should be assigned to a circuit, i.e., PDU with large peak and large utilization. Recently added PDU in CSE (shown in Table II) provide an example of this scenario. Hosting scientific research-oriented workloads, the PDU in CSE normally operate near their measured peak power.

Diversity-Aware Provisioning: The integer programming approach fills the available peak power capacity. SPPF handles inverted PDUs correctly. We believe that the best of both can be achieved by approaches that consider a datacenter’s diverse power workloads. As a proof of concept, we designed the approach in Table III.

Our approach accepts 3 inputs: 1) the list of candidate PDUs, 2) a target capacity, and 3) the CDF of power utilization across the whole datacenter. First, we compute the SPPF assignment. If the SPPF approach uses all available capacity, we simply return this assignment.

Otherwise, we save it as our base assignment and look for knapsack assignments that use more available capacity. When we find a possible knapsack solution, we compute its diversity aware (DA) score by subtracting the capacity that each PDU could lose to an inversion from the capacity used. (Note, the inversion may include any unused PDU.) If the DA score of the new solution is greater than the DA score of our base, the new solution becomes the base.

Our approach is a heuristic. We sacrifice the guaranteed predictability of SPPF for the ability to select PDU with large peaks. Our goal is to show that diversity-aware provisioning can perform well. We leave the design of an optimal diversity-aware approach to future work.

Preliminary Results: Figures 4 and 5 compare the three approaches detailed above under nameplate ratings and measured peak power respectively. Integer programming results were very hard to predict under nameplate ratings, reflecting the large impact of software workloads on typical power needs. Under measured peak power, integer programming results were less varied (esp. for CSE), but inverted PDU still led to poor choices for OSU and PROD. Recall, OSU and PROD host web and enterprise workloads where typical and measured peak power could differ a lot. As expected, SPPF assignments were monotonic as circuit capacity increased. However, SPPF also tended to make sub-optimal choices as the available capacity increased. As described earlier, this occurs because SPPF can waste circuit capacity too, up to the peak needs of the next largest PDU to be selected. Our approach made good decisions in the face of inverted PDU, falling back to the safe SPPF approach when integer programming performed poorly. Further, it exploited peak power as proxy for typical power—often performing better than both SPPF and integer programming. Specifically, it matched or outperformed SPPF in 98% of the measured-peak experiments reported in Figure 5. It matched or outperformed the integer programming approach in 89% of measured-peak experiments.

Comparison to Other Strategies: We also implemented three other competing provisioning strategies. First come first serve orders PDU according to their arrival in the datacenter (inferred from the dates of their server makes and models). Largest peak power first orders PDU according to their peak power, but in descending order (the opposite of SPPF). Smallest then largest implements an alternative heuristic that applies SPPF then LPPF alternatively. Our diversity-aware outperforms these approaches in 80%, 93%, and 70% of our measured-peak conditions. Even taking the absolute best across all studied approaches, our diversity-aware approach was the best in 62% of our

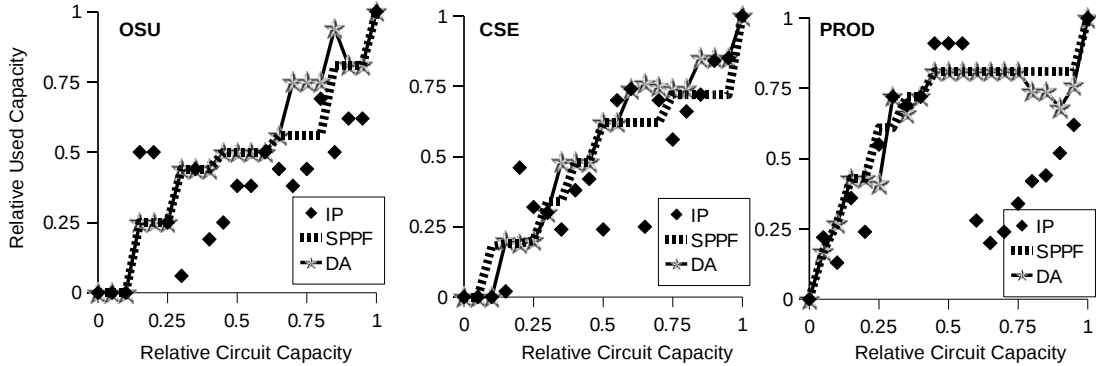


Fig. 4: Comparison of provisioning strategies. X-axis shows the circuit capacity relative to sum of **nameplate ratings** observed across candidate PDU (i.e., $\frac{C}{\sum_{n \in N} P_{nr}(n)}$). Y-axis shows the typical power draw on the circuit relative to the sum of typical power across candidate PDUs (i.e., $\frac{\sum_{i \in \text{Assignment}} P_a(i)}{\sum_{n \in N} P_a(n)}$). Larger values on x-axis indicate larger circuit capacity. Larger values on the y-axis indicate a better provisioning strategy.

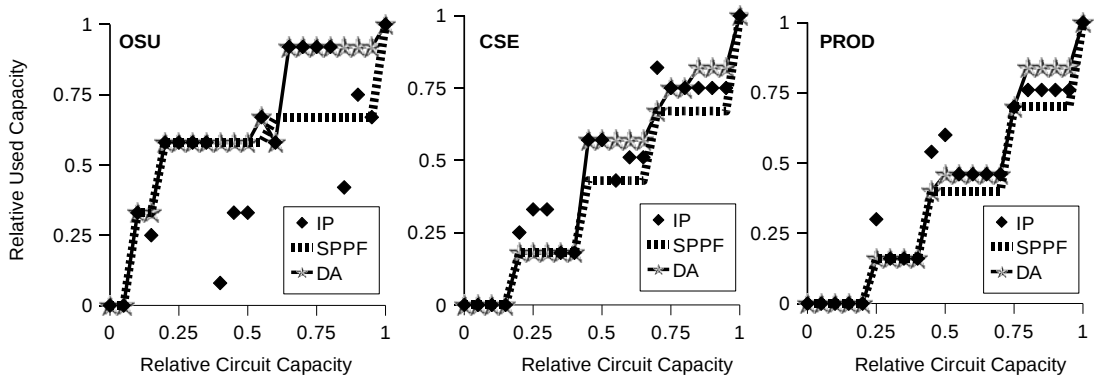


Fig. 5: Comparison of provisioning strategies. X-axis shows the circuit capacity relative to sum of **measured peak power**.

tests. Further, when our approach was not the best, it trailed the leader on average by only 12%.

IV. RELATED WORK

Nathuji et al. [13] exploit the platform heterogeneity of datacenters, showing high variance in the power efficiency of application workloads across different platforms. Such heterogeneity affirms our observation of diverse power workloads in datacenters. Fan et al. [8] show different peak power needs for different types of applications. They were among the first to observe that nameplate ratings tend to overestimate actual power needs, leading to wasted circuit capacity. They also quantify the significant gap between nameplate ratings and measured peak power. Our work shows that both of these peak power estimators are nonmonotonic relative to actual power needs.

In Power Routing, Pelley et al. [14] propose software control for the mapping of servers to circuits, allowing datacenter managers to dynamically control the applications placed on a circuit. On one hand, this infrastructure would allow managers to acquire the measured peak power, but some order inversions persist even under the measured peak power. Wang et al. [19] and Femal et al. [9] focused on the infrastructure for dynamic power allocation for overprovisioned circuits. They also studied measured peak power but as a function of time. Femal et al. [9] allocate peak power while maximizing throughput and balancing load according to service-level requirements. Complementing these works, our work proposes diversity-aware provisioning which accounts for nonmonotonic measured peak power. In future work, we hope to apply our provisioning approach in a dynamic setting.

V. FUTURE WORK

This paper presents an empirical study of 3 real datacenters, focusing on their power workloads. Our study shows that power utilization is not only low; it is also diverse. Such workload diversity complicates power provisioning, making it difficult to use all available power capacity. We have also proposed a new provisioning strategy that outperforms approaches commonly used in practice. Our strategy assigns workloads with large peak power to a circuit only if their expected utilization exceeds the expectation of many small-peak workloads.

In future work, we would like to explore diversity-aware provisioning further. Our current approach provisions power to one circuit. Our first step is to extend our approach to multiple circuits. Unlike our one-circuit case study, multiple circuits requires balancing order inversions—not just avoiding them. Our second step is to provide theoretic bounds on diversity-aware provisioning to complement our empirical results. Finally, we also plan to study diversity-aware provisioning in dynamic contexts, e.g., PDU with built in transfer switches.

VI. ACKNOWLEDGEMENTS

Joel Poualeu, Todd Wolhurst, and Dave Kniesely provided invaluable technical assistance for data collection. Kathleen Starkoff graciously allowed us to study the university-wide datacenters. Rajiv Ramnath, Daiyi Yang, and Nan Deng provided feedback on early drafts of this paper. The anonymous reviewers for WEED 2011 provided excellent feedback as well. This work was funded by the Christopher Stewart's startup funds from The Ohio State University.

REFERENCES

- [1] Coefficient of variation. http://en.wikipedia.org/wiki/Coefficient_of_variation.
- [2] F. Ahmad and T. N. Vijaykumar. Joint optimization of idle and cooling power in data centers while maintaining response time. In *Conference on Architectural Support for Programming Languages and Operating Systems*, Mar. 2010.
- [3] L. Barroso and U. Holzle. *The Datacenter as a Computer – An Introduction to to the Design of Wharehouse-Scale Machines*. Morgan and Claypool Publishers, 2009.
- [4] J. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle. Managing energy and server resources in hosting centers. In *ACM Symp. on Operating Systems Principles*, Oct. 2001.
- [5] J. Choi, S. Govindan, B. Urgaonkar, and A. Sivasubramaniam. Profiling, prediction, and capping of power consumption in consolidated environments. In *IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, Sept. 2008.
- [6] D. Clark. Power-hungry computers put data centers in bind. The Wall Street Journal Online, 2005.
- [7] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan. Full-system power analysis and modeling for server environments. In *In Workshop on Modeling Benchmarking and Simulation (MOBS)*, 2006.
- [8] X. Fan, W. Weber, and L. Barroso. Power provisioning for a warehouse-sized computer. In *Int'l Symp. on Computer Architecture*, June 2007.
- [9] M. Femal and V. W. Freeh. Boosting data center performance through non-uniform power allocation. In *IEEE Int'l Conference on Autonomic Computing*, June 2005.
- [10] S. Govindan, J. Choi, B. Urgaonkar, A. Sivasubramaniam, and A. Baldini. Statistical profiling-based techniques for effective power provisioning in data centers. In *EuroSys Conf.*, Apr. 2009.
- [11] O. Ibarra and C. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *Journal of the ACM*, 22(4), 1975.
- [12] D. Meisner, B. Gold, and T. Wensich. Powernap: Eliminating server idle power. In *Conference on Architectural Support for Programming Languages and Operating Systems*, Mar. 2009.
- [13] R. Nathuji, C. Isci, and E. Gorbato. Exploiting platform heterogeneity for power efficient data centers. In *IEEE Int'l Conference on Autonomic Computing*, June 2007.
- [14] S. Pelley, D. Meisner, P. Zandevakili, T. Wensich, and J. Underwood. Power routing: Dynamic power provisioning in the data center. In *Conference on Architectural Support for Programming Languages and Operating Systems*, Mar. 2010.
- [15] L. Ramakrishnan, K. Jackson, S. Canon, S. Cholia, and J. Shalf. Defining future platform requirements for e-science clouds. In *Symposium on Cloud Computing*, June 2010.
- [16] N. Sharma, S. Barker, D. Irwin, and P. Shenoy. Blink: Supply-side power management for server clusters. In *Conference on Architectural Support for Programming Languages and Operating Systems*, Mar. 2011.
- [17] C. Stewart, T. Kelly, A. Zhang, and K. Shen. A dollar from 15 cents: Cross-platform management for internet services. In *USENIX Annual Technical Conf.*, June 2008.
- [18] C. Stewart and K. Shen. Some joules are more precious than others: Managing renewable energy in the datacenter. In *Workshop on Power Aware Computing and Systems(HotPower)*, Sept. 2009.
- [19] X. Wang and M. Chen. Cluster-level feedback powere control for performance optimization. In *Int'l Symp. on High Performance Computer Architecture*, Feb. 2008.