# Power Provisioning for Diverse Datacenter Workloads

Jing Li
The Ohio State University

Christopher Stewart
The Ohio State University

## ABSTRACT

*The applications 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 applications, often based on their peak power needs. In this work, we studied the actual and peak power needs of 3 real datacenters, using data from 1) hardware manufacturers and 2) actual, observed power needs to estimate peak needs. We found that actual power needs were nonmonotonic relative to peak needs. That is, some applications with low actual power needs had large peak needs—stemming from the diverse power utilization of datacenter applications. Such diversity caused surprising order inversions where applications with smaller peak power had larger actual 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.*

## 1. INTRODUCTION

Datacenters (sometimes written data centers) host a wide range of networked applications from e-commerce and enterprise services to scientific computing [3, 23]. 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 [20]. These circuits "break" when their supported power workload exceeds preset limits, leading to performance capping [4,6,10,25], costly electrical upgrades [7], or brownouts [10, 29]. 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 application's power needs, use circuit capacity limits as a constraint, then find a mapping that uses power capacity well. However, in practice, an application's power needs can increase over time. An application-to-circuit assignment based on a snapshot where actual needs underestimate future needs risks circuit breaks. To reduce these risks, datacenter managers normally provision based on the estimated peak power—not the application's actual needs at a particular moment. Nameplate ratings provided by hardware manufacturers are widely used to estimate peak power [9]. They are often discounted (by up to 60% [9, 10]) 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, 10, 13]. Compared to nameplate ratings, the measured peak power requires that application power needs can be measured but provides tighter workload-aware estimates.

We studied the actual, measured peak, and nameplate power of 3 real datacenters and used the results to design a new power provisioning approach. In the context of our study, an application reflected the combined software workload running on a cluster of servers. Application boundaries were defined by power distribution units. Beyond the study, our methods and results allow for application boundaries to be defined by other factors, e.g., the customer name or server ID. The datacenters that we worked with are described below:

> **Codename OSU** contains 1300 servers with 400kW maximum power capacity for computing. 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 with 40kW maximum power capacity for computing. 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 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.

Even though the managers of these datacenters supported our research, they could not grant *carte blanche* access to the datacenter floor. Instead, we were given read-only access—we could not unplug or move any hardware being used in produc-

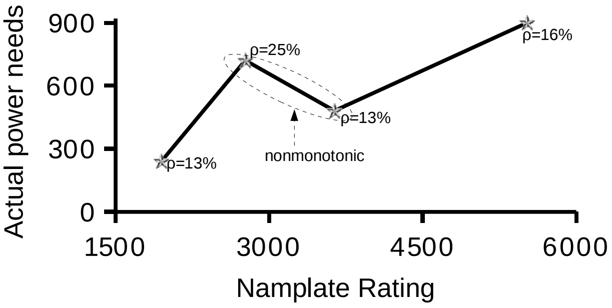

**Figure 1: Nonmonotonic nameplate ratings in OSU. The x-axis shows 4 PDU sorted by nameplate rating. $\rho$ shows the power utilization of each (i.e., power needs divided by estimated peak power).**

tion. We also could not change the software systems of hosted applications. To comply with these constraints, we collected only data that was visible during a walk through on the datacenter floor. Fortunately, this data provided valuable information on most power workloads supported across the whole datacenter.

Our study exposed a key aspect of power workloads in the datacenter: *Actual power needs were nonmonotonic relative to peak needs.* For example, Figure 1 shows an application that had low actual power needs even though it ran on large-nameplate hardware. Such nonmonotonicity produces order inversions, i.e., situations where peak needs order applications differently than actual needs. When two applications are inverted, the application with larger peak power needs will require more provisioned capacity even though it uses less actual power. We found that up to 38% and 28% of PDU pairs in our studied datacenters were inverted based on nameplate ratings and measured peak power respectively. The application 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.

Nonmonotonic peak power makes it hard to predict the actual power draw of applications assigned to a circuit. Assignments with relatively large peak needs can fall well below the median in terms of their actual power needs. The integer programming approach described earlier 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 actual power needs. A simple approach, *smallest peak power first (SPPF)*, totally avoids inversions, producing assignments that increase the actual power draw as circuit capacity increases. However, SPPF assignments can have relatively low actual power needs when applications 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. We used real data from our studied datacenters to compare these approaches. Our approach performed as well or better than integer programming and SPPF in 89% and 98% of our tests. Further, we observed that our approach behaved like SPPF, finding assignments that performed better (i.e., greater actual power draw) as circuit capacity increased.

Our contributions are:

1. We describe methods to study datacenters under strict but widely enforced access policies.

2. We present empirical data showing that actual power needs can be nonmonotonic relative to peak needs for both nameplate ratings and measured peak power.

3. We propose a new provisioning approach that considers diverse power workloads in the datacenter.

The remainder of this paper is as follows. Section 2 describes the access policies enforced in real datacenters and data collection methods used for our study. Section 3 presents our study of actual, measured peak, and nameplate power across whole datacenters. Section 4 makes the case for diversity-aware power provisioning. Section 5 covers related work. Section 6 concludes.

## 2. DATA COLLECTION

The datacenter's floor is its achilles heel; If accessed by the wrong people, a wide range of problems can occur from unplugged cords to hijacked USB ports. Datacenter managers must restrict access to the datacenter floor, even among staff and amicable researchers. These policies make it difficult to measure power workloads. Here, we present techniques to capture PDU-level power needs and namplate ratings without compromising such access policies.

The statement on auditing standards (SAS 70) outlines the following policies as a starting point for protecting the datacenter floor:

> *Data Center Security Staff: These individuals should perform a host of duties on a daily basis, such as monitor intrusion security alarm systems;... monitoring to prevent unauthorized access, such as tailgating; assist all individuals who have authorized access; controlling access to the data center by confirming identity; issue and retrieve access badges;.... Enforcement of no unauthorized photography. [24].*

Almost all production datacenters enforce similar policies at a bare minimum. However, datacenters that host many applications (esp., from untrusted customers) are often more cautious. In the following anonymized email excerpt, a datacenter manager granted our research team access to the datacenter floor only if we agreed to abide by stricter rules:

> *We have asked them to arrange a time to escort you all through the datacenter floor. You can record the Amp[sic] readings and server models. We have asked them not to provide any information about the server names or functions. We have also instructed them that if your group does record server names and locations then you will be asked to leave.*

At first, we thought that strict access policies would severely limit the data that we could collect on power workloads, thwarting our research agenda. But after escorted visits to real datacenters, we realized that labels and LCD displays on the datacenter floor expose a lot of power workload data. For example, most modern PDU display their actual power usage every few seconds. Such data is easily visible to humans, so we spent 3 months at OSU and 3 months at PROD and CSE manually harvesting such data.

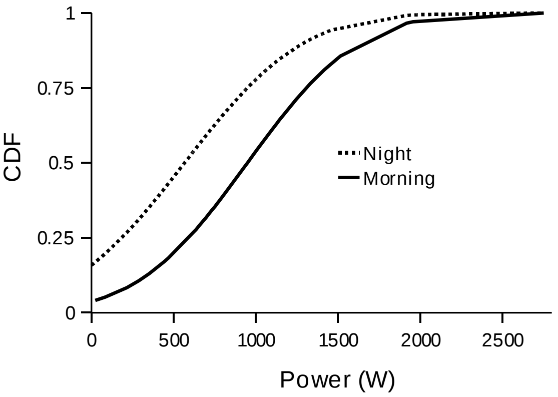

**Figure 2:** **Snapshots of whole-datacenter power needs in PROD. CDF stands for cummulative distribution function. Power needs are taken from the LCD display of each PDU (taken during the morning and night, respectively). Power needs at night were lower than power needs during the day (the CDF rises to 100% more quickly at night than during the day).**

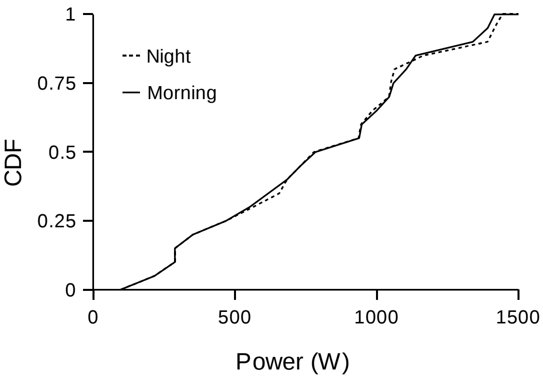

**Figure 3:** **Snapshots of whole-datacenter power needs in CSE.**

## 2.1 Actual Power Needs

Modern PDU not only distribute power, they also report the power usage of their constituent devices. Today, over 60% of the PDU in the TrippLite product line include an LCD display of power usage [17]. Among our studied datacenters, PDU with built-in display were even more popular, accounting for 81% (OSU), 92% (CSE), and 94% (PROD) of the PDU deployed in the last 5 years. We exploited these displays by literally walking around the datacenter floor and writing down their readings.

Figure 2 shows 2 readings of PDU displays across the entire PROD datacenter. One reading was taken in the morning (before 11am) and the other at night (after 6pm). The power needs differed between these times of the day. The median PDU at nighttime used only 60% of the power used by the median morning PDU. We observed several PDU that exhibited diurnal patterns in their power needs, hitting their measured peak in the afternoon and their low in the evening. Such diurnal workloads have been observed many time in prior work [5, 19, 26, 27]. They are often related to social patterns like work schedules. Figure 3 shows that CSE lacked such diurnal cycles; power needs were stable throughout the day. This was expected since the research workload is less affected by social patterns.

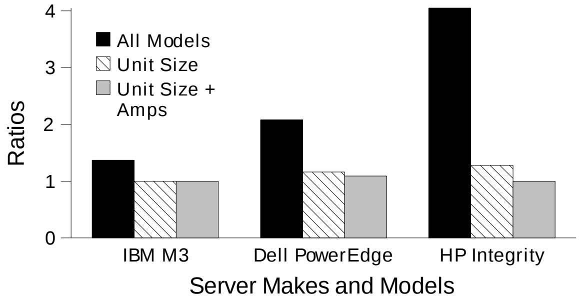

**Figure 4:** **Variance of possible nameplate ratings for HP, Dell and IBM server models. The y-axis shows the ratio of the largest to smallest possible nameplate ratings under the listed constraints.**

## 2.2 Nameplate Ratings

The nameplate rating of a single device is sum of the maximum power draw from its components. For a server, this reflects a workload that fully uses disks, memory, and cores at the same time. For this work, we define a PDU's nameplate rating as the sum of the ratings of its constituent devices plus its own rating. Nameplate ratings are normally easy to get since most hardware companies publish them on public websites [14, 16]. However, published ratings vary depending on the number of cores and disks used by a server and such detailed information is off limits in public datacenters. We recorded easily visible data on the makes, models, sizes, and PDU connections of servers as we walked around the datacenter floor. We found that this data was enough to help us infer nameplate ratings. Specifically, we relied on 2 key observations:

1. *In the datacenter, space is a commodity.* Large servers (e.g., 4U) occupy more units than the smaller sized servers (e.g., 1U) because they are expected to do more work. Configurations that could be applied to 1U servers are rarely actually applied to 4U servers, even if hardware manufacturers allow for such large wimpy nodes.

2. *Circuit breaks are undesirable.* Servers connected to fully occupied PDU normally use less than their share of the circuit breaker's limit. For example, consider 20 systems in a 15A circuit. Most systems will not use more than .75A because managers are cautious.

Figure 4 shows our results for narrowing down nameplate ratings. We started with only make and model information shown on the x-axis. The difference between the largest and smallest possible nameplates for the server model are shown in the all models bar. The effect of using size and PDU ports to narrow down the rating are also shown. We selected 3 server models from HP, Dell, and IBM to illustrate our approach, focusing on the HP and Dell cases below.

- One datacenter hosted an HP Integrity. This server model comes in 4U to 17U sizes and allows for a wide range of processor configurations. Across all model sizes, we found nameplate ratings that varied from 1kW to 4kW. When we narrowed down to servers in the 4U category

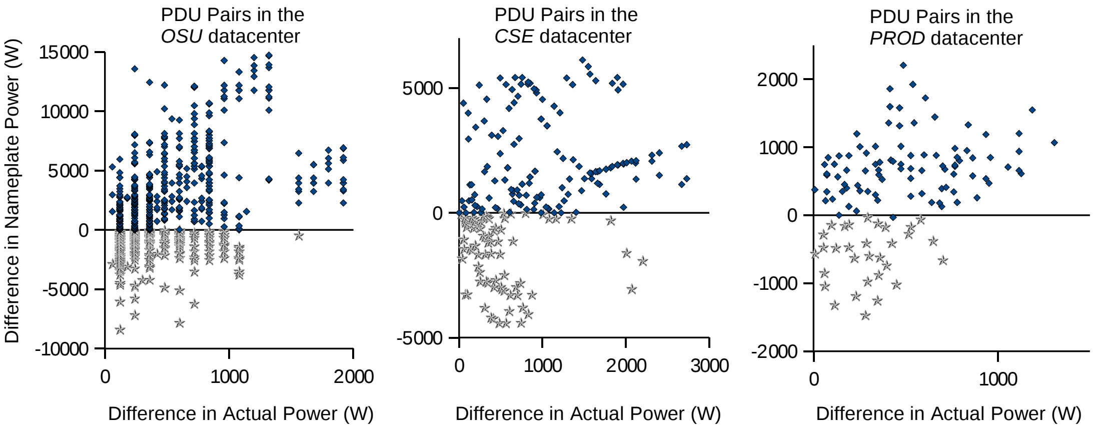

**Figure 5: Nonmonotonic peak power and order inversions. In this figure, peak power is a PDU's nameplate rating. Each point reflects one pair of PDU. All possible pairs are shown. The X axis shows the difference in actual power needs between two PDU in a pair, (i.e., $P_a(i) - P_a(k)$). The Y axis shows the difference in nameplate ratings ($P_{np}(i) - P_{np}(k)$). Stars(points below the 0 on the y-axis) represent PDU pairs that are inverted.**

fewer configurations were available, ranging from 1 Single-Core Intel Itanium 2 Processor to 4 Dual-Core Intel Itanium 2. The largest possible 4U rating was only 1.36kW. Finally, we observed that the particular integrity in question was hosted on a 15A PDU with 12 servers, making it likely that the server in question reflected the minimum 4U configuration [14].

- Another datacenter hosted a Dell PowerEdge server. This server model has 1U to 2U sizes and also allows for various range of processor configurations. Across all model sizes, we found nameplate ratings that varied from 0.75kW to 1.34kW. When we narrowed down to servers in the 1U category, fewer configurations were available, ranging from 2 Quad-Core Intel Xeon 5300 to 1 Dual-Core Intel Low Volt Xeon 5148. The largest possible 1U rating was 0.75kW. Finally, we observed that the particular PowerEdge in question was hosted on a 12A PDU with 8 servers, making it likely that the server in question reflected the minimum 1U configuration [8].

As these examples illustrate above, these observations help us narrow down the nameplate rating range from the information that are given by the manufacturers.

## 3. NONMONOTONIC PEAK POWER

Two applications $i$ and $k$ are order inverted if:

$$P_a(i) < P_a(k) \quad and \quad P_{nr}(i) > P_{nr}(k) \qquad (1)$$
$$or \quad P_a(i) < P_a(k) \quad and \quad P_{mp}(i) > P_{mp}(k) \qquad (2)$$

The $P$ function captures an application's power workload. Parameters $a$, $nr$, and $mp$ stand for actual needs, nameplate rating, and measured peak respectively. When a datacenter has many inverted applications, we say that peak power is nonmonotonic relative to actual power.

Figure 5 plots PDU pairs across our 3 studied datacenters, marking the inverted pairs. In this figure, we used nameplate rating ($P_{nr}$) to describe an application's peak power. These ratings were compared with a snapshot of power usage taken after walking the datacenter floor in the morning (between 7–11am). We found that 27%, 38%, and 23% of PDU pairs were inverted for OSU, CSE, and PROD respectively. Recall, provisioned capacity is often based on peak power. The difference between the peak needs of inverted PDU reflects lost circuit capacity when the PDU with larger peak needs is assigned in place of the PDU with larger actual needs. 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 actual needs in OSU, CSE, and PROD respectively.

**Impact of the Time of the Day:** Section 2 showed that in some datacenters the actual power workload depends on the time of day (diurnal cycles). First, we studied the frequency of order inversions across diurnal periods. We took snapshots of each datacenter in the morning, afternoon, and evening and on the weekends. Each snapshot provided actual 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%. PDU pairs were almost equally inverted at all times of the day for the studied datacenters. 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 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 an application's workload, providing an upper bound that is often closer to actual needs than nameplate ratings [9, 10]. It is quickly becoming the preferred approach to estimate peak power needs [13,21]. We took over 100 samples of the LCD display of each PDU in our study. The largest sample was our measured peak power. We then compared the measured peak to actual morning needs, as in Figure 5. We found that measured peak power significantly reduced the number of inverted PDU pairs for OSU and CSE. Table 1 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 actual needs (like nameplate ratings), explaining the increased number of inverted pairs. Table 1 also shows that measured peak power reduced the potential for inverted PDU to waste circuit capacity across all studied datacenters.

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

**Table 1: 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.

**Other Peak Power Estimators:** Nameplate ratings are an upper bound on an actual power needs. A bound that can be reached by only the workloads that stress all server hardware at the same time. On the other hand, measured peak power provides a tighter and porous bound since data observed in the past may be eclipsed by future peaks. Real peak power falls somewhere between these two estimators. Figure 6 shows 3 inverted PDU pairs in our studied datacenters where the highest measured peak is greater than the lowest nameplate rating. These PDU pairs suggest that some PDU pairs would be inverted for any peak power estimator that is bound by the measured peak and nameplate rating. In other words, the ranking of the peak power estimates of these inverted pairs can not be reversed by using a different peak power estimator. For OSU and PROD, we note that even discounting the measured peak by 5% (95% of original) did not prevent the inversion.

**Power Workload Diversity:** Nonmonotonic peak power is caused by the diverse range power workloads supported in the studied datacenters. Power utilization, defined below, provides a normalized metric to understand an application's power workload.

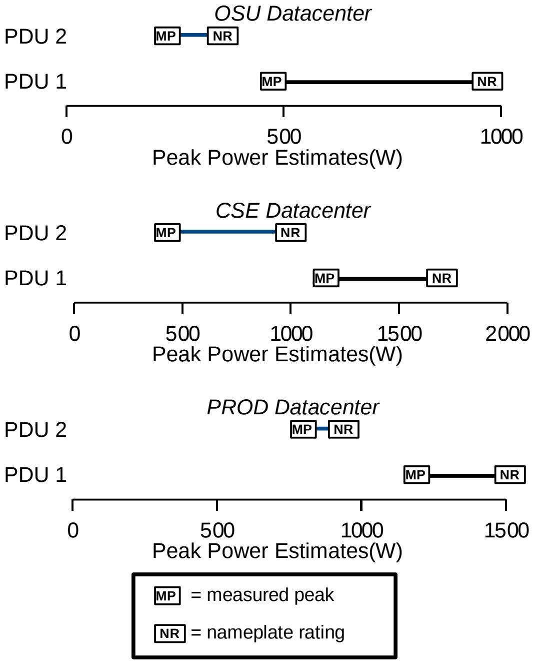

**Figure 6: Unavoidable order inversions for any peak power estimator bound by the nameplate rating and measured peak. The actual power needs (not shown) of PDU 2 are greater than PDU 1. A peak power estimator that uses the measured peak of PDU 1 and the namplate rating for PDU 2 would still invert the PDU.**

$$\rho = \frac{P_a(i)}{P_{nr}(i)} \tag{3}$$

$$or \quad \rho = \frac{P_a(i)}{P_{mp}(i)} \tag{4}$$

Figure 7 shows the diverse power utilization ($\rho$) supported across OSU. This is the source of inverted PDU pairs. When nameplate rating is the denominator, the median and $95^{th}$ percentile power utilization differ by more than 68%. Using the measured peak power, the median and $95^{th}$ percentile differ by 50%.

Figure 7 highlights two important differences between nameplate ratings and measured peak power. First, utilizations with nameplate ratings are generally low with a few outliers ($75^{th}$ 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 applications according to their power utilization. Power utilization is related to device (e.g., CPU) utilization which is well known to be biased under interactive and throughput-oriented datacenter workloads [10, 19,28]. Figure 8 reports similar results but under a broader definition of application—i.e., the combined software workload of a customer. For each customer, the power utilization is the sum of the actual power needed by its servers divided by the sum of each server's peak power. At this granularity, the median and $95^{th}$ percentile differ by 24% under nameplate power and by

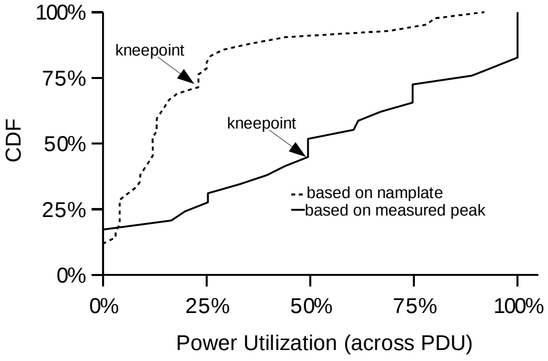

**Figure 7: Power utilization in OSU. Each point reflects a single PDU. The line based on nameplate rating shows $\frac{P_a(i)}{P_{nr}(i)}$. The line based on measured peak shows $\frac{P_a(i)}{P_{mp}(i)}$.**

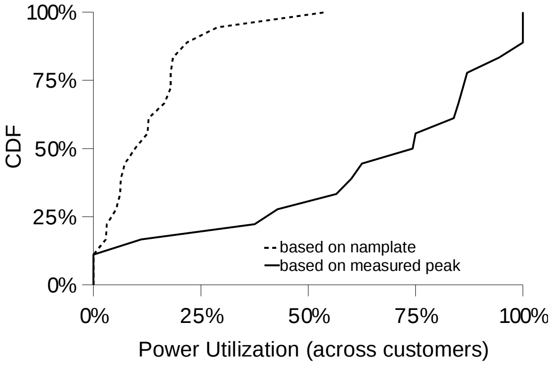

**Figure 8: Power utilization in OSU. Each point reflects the utilization of a single customer.**

| 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 2: 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).**

37% under the measured peak power. This result suggests that the nonmonotonic relationship between actual and peak power persists even when application boundaries are defined differently (here we use customers instead of PDU).

# 4. POWER PROVISIONING

Section 3 showed that peak power was nonmonotonic relative to actual 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* applications from *N* under the following rules:

1. The total peak needs of the selected *k* applications can not exceed preset capacity limits.
2. The goal is for the selected *k* applications is to use as much actual 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 actual available power on any given circuit. To make room, they had to move *k* applications from circuits targeted for the new order. That is, for each targeted circuit, they wanted to migrate applications in a way that moved as much actual power as possible within the capacity limits of other circuits. At another datacenter, managers always considered large future orders when they assigned applications 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.

Our rules also related to provisioning goals outlined in research works. Load unbalancing [22] attempts to fill circuits (i.e., areas in the datacenter) to their capacity for power savings. Recent works have also proposed soft peak limits [2, 13], for instance for evenly spreading power across circuits. After these limits are set, the goal is to fill them.

## 4.1 Provisioning Approaches

We studied the following approaches: integer programming, smallest peak power first, largest peak power first, first come first serve, and our own approach that considers power workload diversity. Throughout this section and in our results, we refer to the PDU listed in Table 2, which shows the workloads of 8 PDU recently added to each studied datacenter, including: their power utilization and measured peak power.

### *Integer Programming.*

When peak needs match actual 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 [15]. Unfortunately in practice, actual needs often fall below peak needs and peak needs are used in provisioning. Here, we used the knapsack integer programming model: 1) measured 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. Recall from Section 1 that we use peak needs because actual needs could lead to circuit breaks.

Section 3 showed that peak needs are nonmonotonic relative to actual needs. Such nonmonotonic peak power means that a subset of applications with high combined peak power may perform poorly in terms of actual power. For example, consider the integer programming approach under a circuit capacity of 960W in OSU (see Table 2). The 960W and 480W PDU are inverted, but knapsack selects the PDU that uses only 38W of actual power— maximizing peak power. The better choice would be the PDU with actual needs of 480W. Note, the integer-programming approach's choice is poor for 2 reasons: First, it does not use as much actual power as possible. Second, it uses less actual 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 actual 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 monotonic results, i.e., an increase in circuit capacity never decreases the actual power draw of selected applications. Recall from Section 3, 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 applications 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 2) provide an example of this scenario. Hosting scientific research-oriented applications, 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 3.

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. Inverted PDU can cause our approach to make poor choices. On the other hand, our approach can make choices that use PDU with large peaks and large utilization, potentially improving performance beyond SPPF. As shown in Table 3, we set a certainty parameter (*cert* $= 0.1$ in the table). This parameter describes the manager's willingness to risk inversions. As the certainty parameter approaches 0.5, our approach behaves like the integer programming (knapsack) approach. As it approaches 0, our approach favors SPPF. Note, our approach was created to show that diversity-aware provisioning can perform well. We leave the design of an optimal diversity-aware approach to future work.

```
DA_Provsioning (candidates, capacity, utilCDF) {
# candidates –> {$P_{mp}(0),...,P_{mp}(i)$}
# 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 3: Pseudo-code of our provisioning approach.**

## 4.2 Results

Figures 9 and 10 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 actual 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 actual 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 actual power—often performing better than both SPPF and integer programming. Note, the *while* loop in Table 3. Our approach consistently selects the best of many integer programming approaches. Specifically, it matched or outperformed SPPF in 98% of the measured-peak experiments reported in Figure 10. It matched or outperformed the integer programming approach in 89% of measured-peak experiments.

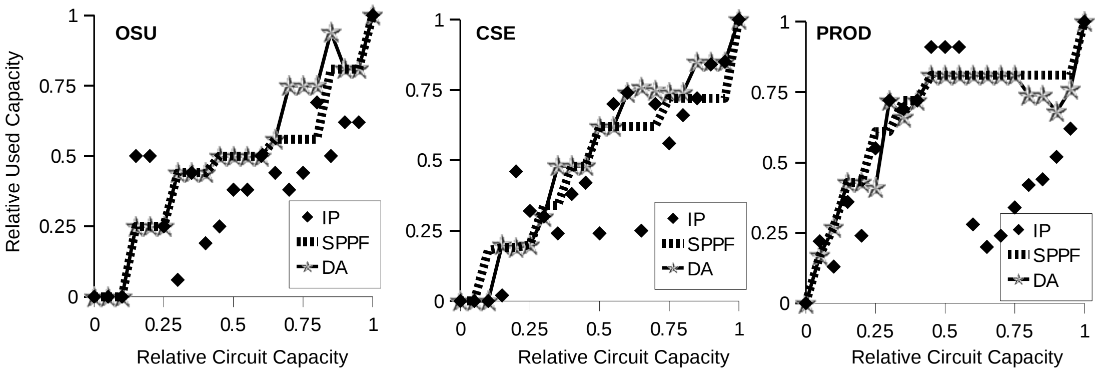

**Figure 9: Comparison of provisioning strategies. X-axis shows the circuit capacity relative to sum of nameplate ratings observed across cadidate PDU (i.e., $\frac{c}{\sum_{n \in N} P_{nr}(n)}$). Y-axis shows the actual power draw on the circuit relative to the sum of actual power across candidate PDUs (i.e., $\frac{\sum_{i \in Assignment} P_a(i)}{\sum_{n \in N} P_a(n)}$). Larger values on x-axis indicate larger circuit capacity. Larger values on the y-axis indicate a better provisioning strategy.**

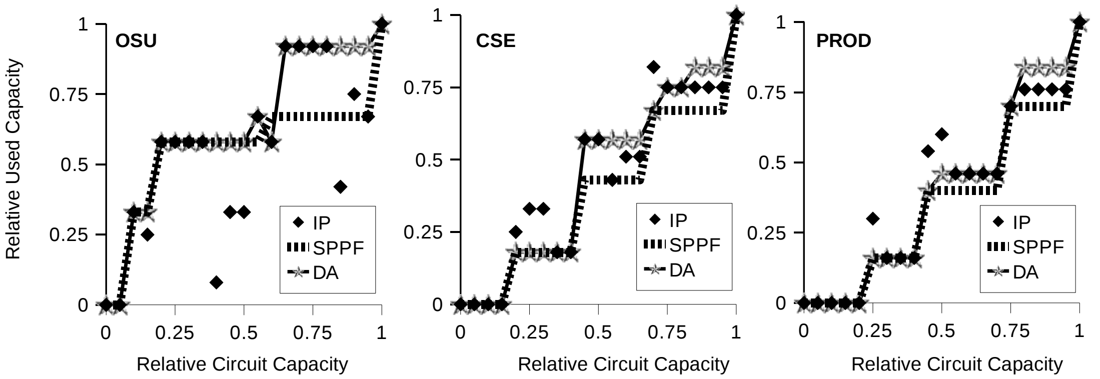

**Figure 10: Comparison of provisioning strategies. X-axis shows the circuit capacity relative to sum of measured peak power observed across cadidate PDU (i.e., $\frac{c}{\sum_{n \in N} P_{mp}(n)}$). Y-axis shows the actual power draw on the circuit relative to the sum of actual power across candidate PDUs.**

Also, it was mostly predictable. Most of the time its assignments improved as circuit capacity increased.

**Impact of Number of Candidates** Figure 11 shows similar qualitative behavior as the number of PDU candidates increases. The integer programming approach under OSU exhibits a sudden unexpected drop in actual power provisioned to the circuit. Our diversity-aware approach performs the best under all tested conditions.

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

## 5. RELATED WORK

Power provisioning for datacenters has been studied by a number of researchers. Our data collection technique addresses the challenges presented by restricted access to the datacenter floor. In our study, we were not allowed to migrate applications to non-production hardware for offline power profiling [6, 13]. We did not have permission to access application logs for request-granularity regression [29]. We could not issue controlled workloads at production servers (recall from Section 2 that some managers hid server functionality), making the non-intrusive Red Pill approach [18] inapplicable.

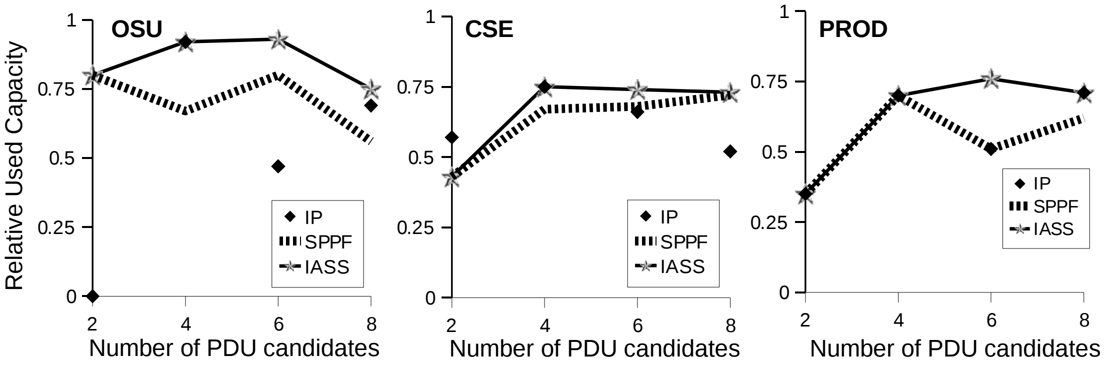

**Figure 11: Provisioning strategies with different number of PDUs. Relative circuit capacity is set to 0.75. X-axis shows the number of PDU avaialble for an assignment. Y-axis shows the actual power draw on the circuit relative to the sum of actual power across candidate PDU.**

Nathuji et al. [20] exploit the platform heterogeneity of datacenters. They also expose that the variance of power efficiency can be very high when the application workloads are assigned under heterogeneous environments. Such heterogeneity supports our diverse workload observation in the datacenters. Our work is based on this diverse range of power utilization observation, and further we find that there is a nonmonotonic relationship between the actual power and the peak power estimation.

Fan et al. [10] show different peak power needs for different types of applications and they also confirm 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. We also study diverse power utilization as an underlying cause.

In PowerNap, Meisner et al. [19] observe that the average CPU utilization in the datacenter is 20-30%. Datacenter servers spend a lot of time idle. They propose a server architecture that uses very little power when idle. To the extent that CPU and power utilization are related, we observe similar results in 3 real datacenters. In Power Routing, Pelley et al. [21] 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. If these inversions are ignored, capacity is wasted or performance is capped.

Ahmad et al. [2] presents PowerTrade and SurgeGuard that reduce both the power consumption and cooling costs. They use an integer programming method to optimize both the idle and cooling power. We show that diversity-aware power provisioning can improve upon the traditional integer programming solutions when applications with large peak power have low actual power needs. We believe that our provisioning approach should complement PowerTrade on the datacenter floor. Wang et al. [30] and Femal et al. [11] focus on the dynamic infrastructures and dealing with peak power allocation for overprovisioning circuits based on their workloads. Femal et al. [11] allocate peak power while maximizing throughput and balancing load according to service-level requirements. Our work goes beyond these various methods by comparing commonly used power provisioning methods with our approach, we observe that the nonmonotonic relation between the peak and actual power favors a diversity-aware provisioning approach. Here again, we see our contribution as a complement to both Wang et al. [30] and Femal et al. [11].

Gandhi et al. [12] studied the power distribution among servers in a server farm in order to minimize mean response time. They apply a particular power cap on a server and run jobs back-to-back to ensure that the server is fully-utilized, and shift between frequency states to ensure that a server doesn't exceed the maximum power. They observe an relationship between power and frequency within a server for a given workload based on DFS and DVFS. They also experiment with diverse workloads and they use the observed maximum power to be the peak power values in their experiments.

## 6. CONCLUSION

We studied power workloads across 3 real datacenters and uncovered a key result: peak power needs were nonmonotonic relative to actual needs. While prior work has shown that peak power needs overestimate actual needs, our result found that the factor by which actual needs are overestimated (i.e., power utilization) varied across applications. Such diverse power workloads persisted whether we used nameplate ratings or measured peak power. Based on this result, we argued that power provisioning approaches should consider the datacenter's workload diversity. We designed and evaluated a proof-of-concept approach. Under realistic data, our approach outperformed commonly used approaches like integer programming and first come first serve (performing as well or better in 89% and 90% of our tests respectively).

Finally, we also overcame strict but common access policies that limited the data that we could collect from the datacenter floor. Specifically, we found noninvasive ways to collect PDU-level power usage and nameplate ratings. We believe these techniques could help researchers conduct empirical datacenter studies in the future.

## 7. 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 Wharhouse-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] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In *USENIX Symp. on Networked Systems Design and Implementation*, Apr. 2008.

[6] 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.

[7] D. Clark. Power-hungry computers put data centers in bind. The Wall Street Journal Online, 2005.

[8] Dell. Dell poweredge 1950 server product details. http://www.dell.com/us/en/dfb/servers/pedge_1950/pd.aspx?refid=pedge_1950&cs=28&s=dfb.

[9] 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.

[10] X. Fan, W. Weber, and L. Barroso. Power provisioning for a warehouse-sized computer. In *Int'l Symp. on Computer Architecture*, June 2007.

[11] 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.

[12] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy. Optimal power allocation in server farms. In *ACM Int'l Conf. on Measurement and Modeling of Computer Systems*, June 2009.

[13] 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.

[14] HP. Hp integrity rx4640 server. http://h18000.www1.hp.com/products/quickspecs/11847_div/11847_div.HTML.

[15] O. Ibharra and C. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *Journal of the ACM*, 22(4), 1975.

[16] Intel. Intel itanium processor family. http://ark.intel.com/ProductCollection.aspx?familyID=451.

[17] T. Lite. Rack mount and stand-alone power distribution units:tripplite. http://www.tripplite.com.

[18] J. Liu. Automatic server to circuit mapping with the red pills. In *Workshop on Power Aware Computing and Systems(HotPower)*, Oct. 2010.

[19] D. Meisner, B. Gold, and T. Wenisch. Powernap: Eliminating server idle power. In *Conference on Architectural Support for Programming Languages and Operating Systems*, Mar. 2009.

[20] R. Nathuji, C. Isci, and E. Gorbatov. Exploiting platform heterogeneity for power efficient data centers. In *IEEE Int'l Conference on Autonomic Computing*, June 2007.

[21] S. Pelley, D. Meisner, P. Zandevakili, T. Wenisch, 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.

[22] E. Pinheiro, R. Bianchini, E. Carrera, and T. Heath. Load balancing and unbalancing for power and performance in cluster-based systems. In *International Workshop on Compilers and Operating Systems for Low Power*, 2001.

[23] 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.

[24] SAS70.com. Sas 70 overview presentation and audit checklist. http://sas70.com/sas70_overview.html.

[25] 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.

[26] R. Singh, U. Sharma, E. Cecchet, and P. Shenoy. Autonomic mix-aware provisioning for non-stationary data center workloads. In *IEEE Int'l Conference on Autonomic Computing*, June 2010.

[27] C. Stewart, T. Kelly, and A. Zhang. Exploiting nonstationarity for performance prediction. In *EuroSys Conf.*, Mar. 2007.

[28] 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.

[29] 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.

[30] X. Wang and M. Chen. Cluster-level feedback powere control for performance optimization. In *Int'l Symp. on High Performance Computer Architecture*, Feb. 2008.