2018 IEEE International Symposium on High Performance Computer Architecture

A Spot Capacity Market to Increase Power Infrastructure Utilization in Multi-Tenant Data Centers*

Mohammad A. Islam
UC Riverside

Xiaoqi Ren
Caltech

Shaolei Ren
UC Riverside

Adam Wierman
Caltech

***Abstract*—Despite the common practice of oversubscription, power capacity is largely under-utilized in data centers. A significant factor driving this under-utilization is fluctuation of the aggregate power demand, resulting in unused "spot (power) capacity". In this paper, we tap into spot capacity for improving power infrastructure utilization in multi-tenant data centers, an important but under-explored type of data center where multiple tenants house their own physical servers. We propose a novel market, called SpotDC, to allocate spot capacity to tenants on demand. Specifically, SpotDC extracts tenants' rack-level spot capacity demand through an elastic demand function, based on which the operator sets the market price for spot capacity allocation. We evaluate SpotDC using both testbed experiments and simulations, demonstrating that SpotDC improves power infrastructure utilization and creates a "win-win" situation: the data center operator increases its profit (by nearly 10%), while tenants improve their performance (by 1.2–1.8x on average compared to the no spot capacity case, yet at a marginal cost).**

***Keywords*-Data center, market approach, power management, spot capacity**

I. Introduction

Scaling up power infrastructures to accommodate growing data center demand is one of the biggest challenges faced by data center operators today. To see why, consider that the power infrastructure (e.g., uninterrupted power supply, or UPS), along with the cooling system, incurs a capital expense of US$10-25 per watt of IT critical power delivered to servers, amounting to a multi-million or even billion dollar project to add new data center capacities [1]–[3]. Further, other constraints, such as local grid capacity and long time-to-market cycle, are also limiting the expansion of data center capacities.

Traditionally, when deciding the capacity, data center operators size the power infrastructure in order to support the servers' maximum power demand with a very high availability (often nearly 100%). Nonetheless, this approach incurs a considerable cost, since the power demands of servers rarely peak simultaneously. More recently, data center operators have commonly used *capacity oversubscription* to improve utilization, i.e., by deploying more servers than what the power and/or cooling capacity allows and applying power capping to handle emergencies (e.g., when the aggregate demand exceeds the capacity) [1], [4], [5].

While oversubscription has proven to be effective at increasing capacity utilization, data center power infrastructure is still largely under-utilized today, wasting more than 15% of the capacity on average, even in state-of-the-art data centers like Facebook [1], [6], [7]. This is not due to the lack of capacity demand, as many new data centers are being constructed. Instead, the reason this under-utilization remains is that, regardless of oversubscription, the aggregate server power demand fluctuates and does not always stay at high levels, whereas the infrastructure is provisioned to sustain a high demand in order to avoid frequent emergencies that can compromise data center reliability [1], [5], [8]. Consequently, there exists a varying amount of unused power capacity, which we refer to as *spot (power) capacity* and illustrate in Fig. 2(a) in Section II.

Spot capacity is common and prominent in data centers, and has increasingly received attention. For example, some studies have proposed to dynamically allocate spot capacity to servers/racks for performance boosting via power routing [9] and "soft fuse" [10].

Importantly, all the prior research on exploiting spot capacity has focused on an owner-operated data center, where the operator fully controls the servers. In contrast, *our goal is to develop an approach for exploiting spot capacity in multi-tenant data centers*.

Multi-tenant data centers (also commonly called colocation) are a crucial but under-explored type of data center that hosts physical servers owned by different tenants in a shared facility. Unlike typical cloud providers that offer virtual machines (VMs), the operator of a multi-tenant data center is only responsible for non-IT infrastructure support (like power and cooling), and each tenant manages its own physical servers. Multi-tenant data centers account for five times the energy consumed by Google-type owner-operated data centers altogether [11]. Most tenants are medium/large companies with advanced server management. For example, both Microsoft and Google have recently leased capacities in several multi-tenant data centers for global service expansion, while Apple houses approximately 25% of its servers in multi-tenant data centers [12].

In a multi-tenant data center, power capacity is typically

*An extended abstract of this paper appeared at the ACM SIGMETRICS 2017.

2378-203X/18/$31.00 ©2018 IEEE
DOI 10.1109/HPCA.2018.00071

leased to tenants in advance without runtime flexibility. Traditionally, tenants reserve/subscribe a sufficiently large capacity to meet their maximum demand, but this is very expensive (at US$120-250/kW/month) and results in a low utilization of the reserved capacity. More recently, an increasingly larger number of cost-conscious tenants have begun to reserve capacities that are lower than their peak demand [13], [14]. This is similar to the common practice of under-provisioning power infrastructure (equivalently, oversubscribing a given infrastructure) for cost saving in owner-operated data centers [3], [7]. In fact, even Facebook under-provisions its power infrastructure [1]. Thus, when their demand is high, tenants with insufficient capacity reservation need to cap power (e.g., scaling down CPU [1], [5], [15]), incurring a performance degradation.

Spot capacity complements the traditionally fixed capacity reservation by introducing a runtime flexibility, which is aligned with the industrial trend of provisioning more elastic and flexible power capacities. *Concretely, spot capacity targets a growing class of tenants — cost-conscious tenants with insufficient capacity reservation upfront — and, on a best-effort basis, provides additional power capacities to help them mitigate performance degradation (or equivalently improve performance) during their high demand periods.* More importantly, utilizing spot capacity incurs a negligible cost increase for participating tenants (as low as 0.3%, shown by Fig. 12 in Section V-B). In addition, without power infrastructure expansion, the operator can make extra profit by offering spot capacity on demand. However, exploiting spot capacity is more challenging and requires a significantly different approach in multi-tenant data centers than in owner-operated data centers because the operator has no control over tenants' servers, let alone the knowledge of which tenants need spot capacity and by how much.

Contributions of this work. In this paper, we propose a novel market approach, called Spot Data Center capacity management (SpotDC), which leverages demand bidding and dynamically allocates spot capacity to tenants to mitigate performance degradation. Such flexible capacity provisioning complements the traditional offering of guaranteed capacity, and is aligned with the industrial trend.

Our work is motivated by other spot markets (e.g., cognitive radio [16] and the Amazon cloud [17]). *However, market design for spot power capacity is quite different*, facing a variety of multifaceted challenges. First, the operator does not know when/which racks need spot capacity and by how much. Even without changing workloads, tenants' rack-level power can vary flexibly to achieve different performances [18], and extracting *elastic* spot capacity demand at scale can be very challenging, especially in a large data center with thousands of racks. In addition, practical constraints (e.g., multi-level power capacity) mean that the operator needs a new way to set market prices. Finally, rather than being restricted to only bid the total demand as considered elsewhere (like Amazon [17]), tenants bid for spot capacity differently — *bid a demand vector for their racks which need spot capacity and can jointly affect the workload performance* (Section III-B3).

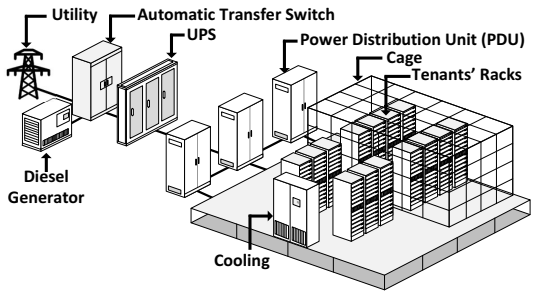


Figure 1. Overview of a multi-tenant data center.

Our design of SpotDC addresses each of these challenges. First, it has a low overhead: only soliciting four bidding parameters for each rack that needs spot capacity. Second, it quickly computes spot capacity allocation under practical constraints, without compromising reliability. In addition, we provide a guideline for tenants' spot capacity bidding to avoid performance degradation (or improve their performance). Finally, as demonstrated by experiments, SpotDC is "win-win": tenants improve performance by 1.2–1.8x (on average) at a marginal cost increase compared to the no spot capacity case, while the operator can increase its profit by 9.7% with any capacity expansion.

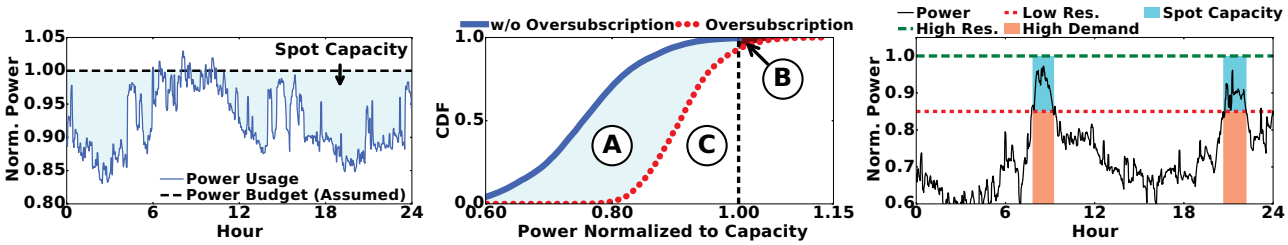
The novelty of our work is that SpotDC is a lightweight market approach to dynamically exploit spot capacity in multi-tenant data centers, complementing fixed capacity reservation. This is in stark contrast with the prior research that has focused on improving power infrastructure utilization in *owner-operated* data centers [2], [9], [10], [19] and maximizing IT resource utilization (e.g., CPU) [20].

II. Opportunities for Spot Capacity

This section highlights that spot capacity is a prominent "win-win" resource in a multi-tenant data center: tenants can utilize spot capacity to mitigate performance degradation on demand at a low cost, while the operator can make an extra profit.

A. Overview of Data Center Infrastructure

Multi-tenant data centers employ a tree-type power hierarchy. As illustrated in Fig. 1, high-voltage grid power first enters the data center through an automatic transfer switch (ATS), which selects between grid power (during normal operation) and standby generation (during utility failures). Then, power is fed into the UPS, which outputs "protected" power to cluster-level power distribution units (PDUs). Each PDU has a IT power capacity of 200-300kW and supports roughly 50-80 racks/cabinets. At the rack level, there is a power strip (also called rack PDU) that directly connects to servers. In a typical (retail) multi-tenant data center, tenants each manage multiple racks and share PDUs.



(a) (b) (c)

Figure 2. **(a)** Illustration of spot capacity in a production PDU [7]. **(b)** CDF of tenants' aggregate power usage. **(c)** A tenant can lease power capacity in three ways: high reservation; low reservation; and low reservation + spot capacity. "Low/high Res." represent low/high reserved capacities.

The capacities at all levels must not be exceeded to ensure reliability. Typically, the UPS and cluster PDUs handle power at a high or medium voltage and hence are very expensive, costing US$10-25 per watt (along with the cooling system and backup generator) [3], [8]. Nonetheless, the rack-level PDU has a lower voltage and is very cheap (e.g., US¢20-50 per watt) [9], [21]. Thus, the capacity bottleneck is at the shared UPSes/PDUs, not at individual tenants' racks. In fact, 20% rack-level capacity margin is already in place [5], and additional over-provisioning is increasingly more common for flexible power distribution to racks (e.g., power routing [9]).

B. Spot Capacity v.s. Oversubscription

Like in owner-operated data centers, power capacity is under-utilized in multi-tenant data centers. Fig. 2(b) plots the cumulative density function (CDF) of measured power at a PDU serving five tenants in a commercial data center over three months. The CDF is normalized to the maximum power and shown as the left-most curve. Suppose that the PDU capacity is provisioned at the maximum power demand. Ideally, if the PDU is always 100% utilized, the power usage CDF would become a vertical line, as shown in Fig. 2(b), which highlights a large gap between the measured CDF and the ideal case.

To improve infrastructure utilization, data center operators commonly oversubscribe the capacity, as tenants typically do not have peak power at the same time. To illustrate oversubscription, we keep the same PDU capacity and add another two tenants, resulting in a new CDF (dotted line in Fig. 2(b)) which is closer to the ideal case than the original CDF. The improved capacity utilization is indicated by the area "A". However, oversubscription may occasionally trigger an emergency when power capacity is exceeded (indicated by the area "B"). This has been well understood, and many power capping solutions have been proposed to handle emergencies [1], [2], [5], [8].

To avoid frequent emergencies, the shared UPS/PDUs must be sized to sustain a high *aggregate* power demand [1], [3], [5], [8]. Consequently, even when some tenants have reached their capacities, other tenants may still have low power usage, possibly resulting in spot capacity at the shared PDU/UPS. The existence of spot capacity is also visualized by the gap (area "C") between the actual and idealized CDFs in Fig. 2(b). Importantly, spot capacity can be allocated to those tenants to improve performances on demand (Section II-C).

Therefore, exploiting spot capacity and power oversubscription are complementary to increasing data center infrastructure utilization: oversubscription is decided over a long timescale and requires power shaving during emergencies [1], [5], [8], whereas spot capacity is dynamically exploited based on the runtime availability to deliver additional power budgets to tenants (with insufficient capacity reservation) for performance improvement.

C. Potential to Exploit Spot Capacity

Even with the same servers/workloads, a tenant's power usage can be *elastic* and can vary significantly depending on power control and/or workload scheduling [18]. Thus, given a server deployment, a tenant's power capacity subscription can vary widely.

Traditionally, each tenant reserves a sufficiently large capacity of the shared PDU with a high availability guarantee (a.k.a. guaranteed capacity) to support its maximum power demand, which is illustrated as "High Res" (high reservation) in Fig. 2(c). The guaranteed capacity subscription, at US$120-250/kW/month, is a major fraction of tenants' cost and can even exceed 1.5 times of the metered energy charge [8], [22]. Furthermore, tenants rarely fully utilize their large guaranteed capacities.

More recently, the shrinking IT budget has placed a growing cost pressure on tenants. A 2016 survey shows that 40% of tenants end up paying more than what they anticipate for their power subscription [14]. Thus, studies on reducing tenants' power costs have been proliferating [13], [23], [24]. Notably, cost-conscious tenants have commonly reserved capacities lower than their maximum power demand to reduce costs [13]. This is illustrated by "Low Res" (low reservation) in Fig. 2(c), and similar to under-provisioning power infrastructures in owner-operated data centers such as Facebook [1], [7]. Then, when their demand is high, tenants with insufficient capacity reservation need to apply power capping, incurring a performance degradation; otherwise, heavy penalties will be applied.

As illustrated in Fig. 2(c), spot capacity helps tenants with insufficient capacity reservation mitigate performance degradation when their power demand is high. Specifically, when a tenant with insufficient capacity reservation has high workloads, the operator can allocate spot capacity, if available, to this tenant's racks as an additional power budget to mitigate performance degradation. The rack-level PDU capacity is not a bottleneck [9], [10], and the operator can dynamically adjust it at runtime, which is already a built-in functionality in many of today's rack-level PDUs [21]. More importantly, utilizing spot capacity incurs a negligible cost for participating tenants (as low as 0.3% and much lower than reserving additional guaranteed capacities, shown in Section V-B).

Spot capacity v.s. guaranteed capacity. Spot capacity is dynamically allocated based on demand function bidding (Section III-B). But, once spot capacity is allocated, it can be utilized over a *pre-determined* time slot (e.g., 1-5 minutes) in the *same* way as guaranteed capacity,[1] with the exception that it may be unavailable in the next time slot. This differs from the Amazon spot market where allocated VMs may be evicted at any time.

In practice, tenants with insufficient capacity reservation often run delay-tolerant workloads (e.g., batch processing), which exhibit large scheduling flexibilities and are run on 50+% servers (with roughly 50% power capacity) in data centers [6], [7]. Note that, for a tech-savvy tenant with advanced power control, insufficient capacity reservation can even apply for delay-sensitive workloads (e.g., web service), as is being done by large companies [1], [3], [7]. In this paper, we use *opportunistic* and *sprinting* tenants to refer to tenants which use spot capacity to mitigate slowing down of delay-tolerant and delay-sensitive workloads, respectively. Thus, a tenant can be both opportunistic and sprinting. In any case, *with the help of spot capacity, a tenant with insufficient capacity reservation can temporarily process its workloads without power capping (or cap power less frequently/aggressively than it would otherwise).*

Importantly, spot capacity targets cost-conscious tenants with insufficient capacity reservation and does not affect the revenue of guaranteed capacity. Even without cost-effective spot capacity, these tenants already choose insufficient capacity reservation; they would not pay the high cost and reserve a sufficiently large amount of guaranteed capacity to meet their maximum power demand. On the other hand, tenants running mission-critical workloads will likely continue reserving a sufficient guaranteed capacity without using intermittent spot capacity.

III. The Design of SpotDC

Our main contribution is a new market approach for exploiting spot capacity, SpotDC, which leverages a new demand function bidding approach to extract tenants' rack-level spot capacity demand elasticity at runtime and reconcile different objectives of tenants and the operator: tenants first bid to express their spot capacity demand, and then the operator sets a market price to allocate spot capacity and maximize its profit. With SpotDC, the operator makes extra profit, while participating tenants mitigate performance degradation (or improve performance) at a low cost.

[1]Section III-C discusses how to guarantee spot capacity for one slot.

A. Problem Formulation

To design SpotDC, we consider a time-slotted model, where each dynamic spot capacity allocation decision is only effective for one time slot. The duration of each time slot can be 1-5 minutes [9].

Model. Consider a data center with one UPS supporting M cluster PDUs indexed by the set $\mathcal{M} = \{m \mid m = 1, \cdots, M\}$. There are R racks indexed by the set $\mathcal{R} = \{r \mid r = 1, \cdots, R\}$, and N tenants indexed by the set $\mathcal{N} = \{n \mid n = 1, \cdots, N\}$. Denote the set of racks connected to PDU m as $\mathcal{R}_m \subset \mathcal{R}$. Note that racks are not shared among tenants in a multi-tenant data center, while a tenant can have multiple racks.

The operator continuously monitors power usage at rack levels [1], [5], [9]. For time slot $t = 1, 2, \cdots$, the predicted available spot capacity at the upper-level UPS is denoted by $P_o(t)$, and the available spot capacity at PDU m is denoted by $P_m(t)$, for $m = 1, 2, \cdots$. How to predict the available spot capacity is discussed in Section III-C. At the rack level, the physical capacity is over-provisioned beyond the guaranteed capacity to support additional power budgets (i.e., spot capacity). The maximum spot capacity supported by rack r is denoted as P_r^R.

The operator sells spot capacity at price $q(t)$, with a unit of \$/kW per time slot. The set of racks that requests spot capacity is denoted by $\mathcal{S}(t) \subseteq \mathcal{R}$, and the actual spot capacity allocated to rack $r \in \mathcal{S}(t)$ is denoted by $D_r(q(t))$.

Objective. The operator incurs no extra operating costs for offering spot capacity, since tenants pay for metered energy usage (and otherwise a reservation price can be set to recoup energy costs). Thus, the operator's profit maximization problem at time t can be formalized as:

$$\underset{q(t)}{\text{maximize}} \quad q(t) \cdot \sum_{r \in \mathcal{S}(t)} D_r(q(t)). \tag{1}$$

Constraints. We list the most important power capacity constraints for spot capacity allocation, from rack to UPS levels, as follows:

$$\underline{\text{Rack}}: \quad D_r(q(t)) \le P_r^R, \ \forall r \in \mathcal{S}(t) \tag{2}$$

$$\underline{\text{PDU}}: \quad \sum_{r \in \mathcal{S}(t) \cap \mathcal{R}_m} D_r(q(t)) \le P_m(t), \ \forall m \in \mathcal{M} \tag{3}$$

$$\underline{\text{UPS}}: \quad \sum_{r \in \mathcal{S}(t)} D_r(q(t)) \le P_o(t) \tag{4}$$

Other constraints, such as heat density (limiting the maximum cooling load, or server power, over an area) and phase balance (ensuring that the power draw of each phase should be similar in three-phase PDUs/UPSes), can also be incorporated into spot capacity allocation following the model in [9], and are omitted for brevity.

In SpotDC, spot capacity allocation is at a rack-level granularity, since tenants manage their own racks while the operator controls upstream infrastructures like PDU and UPS. Note that with a tenant-level spot capacity allocation, the operator would have no knowledge of or control over how a tenant would distribute its received spot capacity among its racks. This can create capacity overloading and/or local hot spots if multiple tenants concentrate their received spot capacity over a few nearby racks served by a single PDU. Finally, rack-level spot capacity allocation does not require a homogeneous rack setup. Different tenants can have different racks with different configurations, and even a single tenant can have diverse rack configurations.

B. Market Design

A key challenge for maximizing the operator's profit is that, due to its lack of control over tenants' servers, the operator does not know tenants' demand function: which tenants need spot capacity and by how much. This is private information of individual tenants. *Prediction* is a natural solution to the challenge — the operator first predicts tenants' responses and then sets a profit-maximizing price for tenants to respond. However, due to the capacity constraints at different levels in Eqns. (2), (3) and (4), prediction needs to be done rack-wise and there can be hundreds or even thousands of racks with dynamic workloads. Most importantly, with prediction-based pricing, spot capacity allocation is decided by the tenants (passively through their responses to the market price set by the operator). This can lead to dangerous capacity overloads in the event of under-predicting tenants' spot capacity demands (i.e., setting a too low price). Thus, prediction-based pricing is not suitable for spot capacity allocation. In contrast, an alternative to prediction is to solicit tenants' demand through bidding, called demand function bidding: each participating tenant first reports its own spot capacity demand to the operator through a bidding process, and then the operator allocates spot capacity by setting a market price while meeting all the capacity constraints.

1) Demand function: The core of demand function bidding is to extract users' demand through a function (called demand function), which can capture how the demand varies as a function of the price. In our context, there can be up to thousands of racks, and even without reducing the workloads, server power can vary to achieve different performances (e.g., with a granularity of watt by Intel's RAPL) [18]. Thus, our goal is to design a demand function that

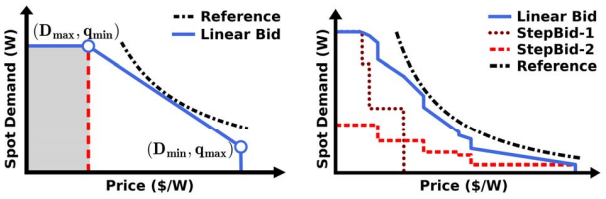


Figure 3. (a) Piece-wise linear demand function. The shaded area represents StepBid. (b) Aggregated demand function for ten racks. StepBid-1 bids $(D_{\max}, q_{\min})$ only, and StepBid-2 bids $(D_{\min}, q_{\max})$ only.

can extract tenants' rack-level elastic spot capacity demand reasonably well yet at a low complexity/overhead.

A straightforward approach is to solicit each tenant's complete rack-level demand curve under all possible prices. We illustrate in Fig. 3(a) an example demand curve (labeled as "Reference," and Section IV-C explains how to derive it). The actual demand curve can be even more complex and multi-dimensional (for multiple racks, as shown in Fig. 4(a)), thus incurring a high overhead to extract. In addition, bidding the complete demand curve is difficult for participating tenants, as they must evaluate their demand under many prices. For these reasons, soliciting the complete demand curve is rarely used in real markets [17], [25].

In practice, *parameterized* demand function bidding is commonly applied when the buyers' demand is unknown to the seller a priori. For example, a step demand function illustrated in the shaded area in Fig. 3(a) is used by Amazon spot VM market [17] and means that a user is willing to pay up to a certain price for a *fixed* amount of requested VMs. We refer to this demand function as StepBid. While it has a low overhead, StepBid can be very different from tenants' actual demand curve ("Reference") shown in Fig. 3(a). Moreover, with StepBid, the operator cannot flexibly allocate spot capacity: a tenant's spot capacity demand can only be either 100% or 0% satisfied. Thus, StepBid cannot capture a tenant's rack-level spot capacity demand elasticity. As illustrated in Fig. 3(b), even at the shared PDU level, StepBid cannot extract the aggregate demand elasticity of multiple racks, thus resulting in a lower profit for the operator (Section V-C). The reason is that, although StepBid can extract the aggregate demand elasticity over a large number of racks, spot capacity allocation is subject to several *localized* constraints (e.g., shared PDU capacity) that each cover only up to a few tens of racks. This is in sharp contrast with Amazon spot market where the unused VMs are pooled together and allocated to a large number of users without restricting one user's demand to any particular rack.

Piece-wise linear demand function. We propose a new parameterized demand function which, as illustrated by "Linear Bid" in Fig. 3(b), *approximates* the actual demand curve using three line segments: first, a horizonal segment: tenant specifies its maximum spot capacity demand for a rack as well as the market price it is willing to pay; second, a linearly decreasing segment: the demand decreases linearly

Algorithm 1 SpotDC— Spot Capacity Management

1: Continuously monitor rack power
2: **for** $t = 0, 1, \cdots$ **do**
3: Each participating tenant analyzes workloads and submits bids
4: Collect bids $\mathbf{b}_r$ and predict spot capacity
5: Decide price $q(t+1)$, send market price and spot capacity allocation to participating tenants, and reset rack capacity via intelligent rack PDU
6: Each tenant manages its power subject to the allocated spot capacity effective for time $t+1$
7: **end for**

as the market price increases; and third, a vertical segment: the last segment indicates tenant's maximum acceptable price and the corresponding minimum demand.

As shown in Fig. 3(a), our linear demand function for rack r is uniquely determined by four parameters:

$$\mathbf{b}_r = \{(D_{\max,r}, q_{\min,r}), (D_{\min,r}, q_{\max,r})\} \tag{5}$$

where $D_{\max,r}$ and $D_{\min,r}$ are the maximum and minimum spot capacity demand, and $q_{\min,r}$ and $q_{\max,r}$ are corresponding prices, respectively. We also allow $D_{\max,r} = D_{\min,r}$ or $q_{\min,r} = q_{\max,r}$, which reduces to StepBid.

We choose our linear demand function for its simplicity and good extraction of the demand elasticity. It also represents a *midpoint* between StepBid (which is even simpler but cannot extract spot capacity demand elasticity) and soliciting the complete demand curve (which is difficult to bid and rarely used in practice [25]). Moreover, the experiment in Section V-C shows that, using our demand function, the operator's profit is much higher than that using StepBid and also fairly close to the optimal profit when the complete demand curve is solicited, thus further justifying the choice of our demand function.

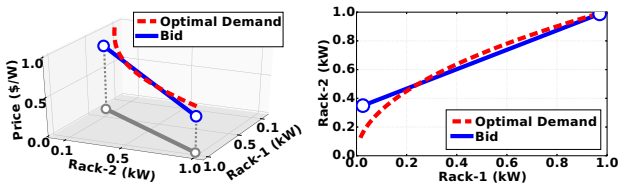
2) Spot capacity allocation: The following three steps describe the spot capacity allocation process, which is also described in Algorithm 1.

Step 1: *Demand function bidding.* Participating tenants, at their own discretion, decide their rack-wise bidding parameters based on their anticipated workloads and needs of spot capacity for the next time slot.

Step 2: *Market clearing.* Upon collecting the bids, the operator sets the market price $q(t)$ to maximize profit, i.e., solving (1) subject to multi-level capacity constraints (2)(3)(4). This can be done very quickly through a simple search over the feasible price range.

Step 3: *Actual spot capacity allocation.* Given the market price $q(t)$ plugged into the demand function, each tenant knows its per-rack spot capacity and can use additional power up to the allocated spot capacity during time t.

3) Tenant's bidding for spot capacity: For a tenant, the power budgets for multiple racks jointly determine the application performance (e.g., latency of a three-tier web service, with each tier housed in one rack). Thus, a key difference from spot VM bidding in Amazon [17] is that, in our context, each participating tenant needs to bid a *bundled* demand for all of its racks that need spot capacity. Nonetheless, the bidding strategy is still at the discretion of tenants in our context, like in Amazon spot market. It can follow a *simple* strategy for each rack: bid the needed extra power (i.e., total power needed minus the reserved capacity) as spot capacity demand with $D_{\max,r} = D_{\min,r}$, and set the amortized guaranteed capacity rate (at US$120-250/kW/month) as maximum price. Tenants routinely evaluate server power under different workloads prior to service deployment [1], [2], [7], and thus can determine their needed power based on estimated workloads at runtime.



(a) (b)

Figure 4. Demand function bidding. (a) Optimal spot capacity demand and bidding curve. (b) 2D view.

On the other hand, advanced tenants with detailed power-performance profiling can also bid holistically for their racks in need of spot capacity. Below, we provide a *guideline* for spot capacity bidding to highlight how advanced tenants may approach this task, although our focus is on the operator's side — setting up a market for spot capacity allocation.

Given each price, there exists an optimal spot capacity demand *vector* for a tenant's racks. The *optimality* can be in the sense of maximizing the tenant's net benefit (i.e., performance gain measured in dollars,[2] minus payment), maximizing performance gain (not lower than payment), or others, which tenants can decide on their own. Consider web service (as described in Section IV-B) on two racks as an example. As illustrated in Fig. 4(a), tenant identifies its demand curve by first evaluating performance gains resulting from spot capacity (Section IV-C) and then finding optimal demand vectors under different prices.

In general, the relation between rack-1 demand and rack-2 demand may be non-linear, as shown in Fig. 4(b). Nonetheless, spot capacity allocated to both racks are determined by the same price and may not follow the optimal demand curve. For example, one rack's spot capacity allocation may change linearly in the other's. Consequently, the tenant needs to *approximate* the optimal demand curve using, e.g., a line shown as "Bid" in Fig. 4(a). We also present the top-down perspective of bidding curve in Fig. 4(b), which indicates the relation between the two racks' actual spot capacity

[2]The monetary value for performance gain [26] is quantified by tenants as described in Section IV-C.

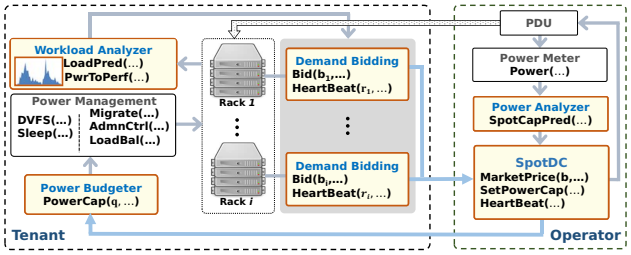


Figure 5. System diagram for SpotDC.

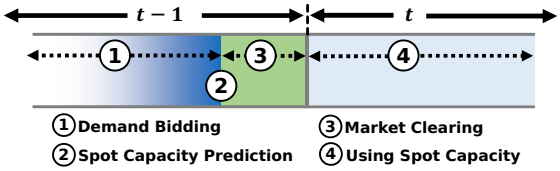


Figure 6. Timing of SpotDC for spot capacity allocation.

demand. In Fig. 4(a), the bidding demand curve includes all the needed parameters: the maximum and minimum demand pairs for the two racks, as well as the corresponding bidding prices (the same $q_{\min}$ and $q_{\max}$ for the two racks).

A tenant can bid similarly if K of its racks need spot capacity: decide the maximum and minimum bidding demand vectors $(D_{\max,1}, \cdots, D_{\max,K})$ and $(D_{\min,1}, \cdots, D_{\min,K})$, which are joined in an affine manner to approximate the optimal K-dimensional demand, and then decide the two corresponding bidding prices.

Finally, it is important to note that tenants can bid freely without their own strategies. Thus, the resulting bidding profile and spot capacity allocation can be significantly different from the theoretical equilibrium point at which each participating tenant's net benefit is maximized (given the other tenants' bids) [25]. In fact, even under a set of simplified assumptions (e.g., concave utility for each tenant, no tenant forecasts the market price, etc.), it is non-trivial to derive the theoretical equilibrium point [25], since a tenant's spot capacity demand involves multiple racks and hence is multi-dimensional. Further, given tenants' strategic behaviors and lack of information about each other, how to reach an equilibrium is a theoretically challenging problem [25]. Thus, we focus on the operator's spot capacity market design and resort to case studies to show the benefit of exploiting spot capacity (Section V), while leaving the theoretical equilibrium bidding analysis as our future work.

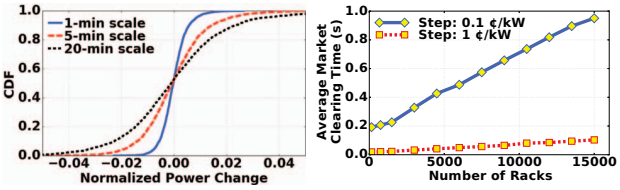
C. Implementation and Discussion

We now illustrate the implementation for SpotDC in Fig. 5, where the application program interfaces (APIs), as highlighted in shaded boxes, facilitate communications between the operator and tenants using a simple network management protocol. In our time-slotted model, the data center operator and participating tenants are synchronized by periodically exchanging `HeartBeat(···)` signals. As suggested by [9], each time slot can be 1-5 minutes in practice. To conclude the design of SpotDC, it is important to discuss a few remaining practical issues.

Timing. We show in Fig. 6 the timing of different stages leading to and during spot capacity allocation in time slot t. For using spot capacity during time slot t, tenants need to submit their demand bids (marked as "1") during time slot $t-1$. The gradient color is to emphasize that most bids are expected to be received closer to time slot t. Then, the operator predicts the available spot capacity (marked as "2") before clearing the market. The market clearing time (marked as "3") is very small (less than a second), and the clearing price is broadcast to the tenants. Finally, from their demand functions, tenants determine their rack-level spot capacity allocation and use it (marked as "4") during time slot t. Note that participating tenants have an entire time slot to decide and send their bids to the operator for using spot capacity in the next time slot. Thus, the communication delay (in the order of hundreds of milliseconds) is insignificant even when tenants submit bids remotely.

Spot capacity prediction. The operator can predict spot capacity by taking the current aggregate power usage as a reference and subtracting it from the physical PDU/UPS capacity. For racks that are currently using spot capacity or request it for the next time slot, the guaranteed rack-level capacity will be used as their reference power usage. Collecting the power readings can be done near instantaneously as a part of the routine power monitoring. The key concern here is how accurate the prediction of spot capacity is. We note that, due to statistical multiplexing, the cluster-level PDU power only changes marginally within a few minutes (e.g., less than $\pm 2.5\%$ within one minute for 99% of the times) [1], [7], [9]. In Fig. 7(a), we show the statistics of PDU-level power variations in our experimental power trace (Section V) and see that it is consistent with the results in [7]: the PDU-level power changes slowly across consecutive time slots. Moreover, in almost all cases, spot capacity is not completely utilized due to the operator's profit-maximizing pricing and multi-level capacity constraints (as seen in Fig. 10). The operator can also conservatively predict (i.e., under-predict) the available spot capacity without noticeably affecting its profit or tenants' performance (Fig. 17). Finally, any unexpected short-term power spike can be handled by circuit breaker tolerance, let alone the power system redundancy in place. Therefore, even with inaccurate spot capacity prediction, the availability of spot capacity can be guaranteed for one time slot almost highly as the normal power capacity provisioning.

Applicability. SpotDC targets a growing class of tenants — cost-conscious tenants with insufficient capacity reservation (even Facebook under-provisions power capacity in its own data center [1]) — and helps them mitigate performance degradation on a best-effort basis. *Utilizing spot capacity is even easier than otherwise: with spot capacity, a tenant caps*



(a) (b)

Figure 7. (a) PDU power variation in our simulation trace. (b) Market clearing time at scale.

power less frequently/aggressively than it would otherwise. Moreover, spot capacity bidding is at the discretion of tenants: it can be either as simple as bidding the needed power at a fixed price, or as sophisticated as holistically bidding for multiple racks in need of spot capacity (Section III-B3). There is no application/workload requirement for tenants to participate in SpotDC, as long as they can control power subject to dynamic spot capacity allocation.

Scalability. The design of SpotDC is highly scalable since only four parameters are solicited for each rack *in need of* spot capacity; no bids are required for racks that do not need extra power demand beyond the reserved capacity. Additionally, our proposed uniform clearing price only requires a scan over feasible prices subject to the infrastructure constraints. Therefore, the market clearing is very fast. We show in Fig. 7(b) the average market clearing time for different numbers of server racks and different search step sizes in our large-scale simulation (Section V-D) on a typical desktop computer. We see that even with 15000 racks, the average clearing time is less than a second for a step size of 0.1 cents/kW. For a step size of 1 cent/kW, the average clearing time is below 100ms. Further, it takes almost no time for the operator to reset rack-level power budgets (e.g., 20+ times per second for our PDU [21] without any timeouts).

Market power and collusion. Tenants with a dominant position may have the power to alter the market price. In theory, tenants might also collude to lower prices. But, this is unlikely in practice, because tenants have no knowledge of the other tenants they are sharing the PDU with, let alone when and where those tenants need spot capacity.

Handling exceptions. In case of any communications losses, SpotDC resume to the default case of "no spot capacity" for affected tenants/racks. In addition, power monitoring at the rack (and even server) level is already implemented for reliability and/or billing purposes [1], [9], [21]. If certain tenants exceed their own assigned power capacity (including spot capacity if applicable), they may be warned and/or face involuntary power cut.

IV. Evaluation Methodology

To evaluate SpotDC we use a combination of testbed and simulation experiments, which we describe below.

Table I
Testbed Configuration.

PDU	Tenant	Type	Alias	Workload	Subscription
#1	Search-1	Sprinting	S-1	Search	145W
	Web	Sprinting	S-2	Web Serving	115W
	Count-1	Opportunistic	O-1	Word Count	125W
	Graph-1	Opportunistic	O-2	Graph Anal.	115W
	Other	—	—	—	250W
#2	Search-2	Sprinting	S-3	Search	145W
	Count-2	Opportunistic	O-3	Word Count	125W
	Sort	Opportunistic	O-4	TeraSort	125W
	Graph-2	Opportunistic	O-5	Graph Anal.	115W
	Other	—	—	—	250W

A. Testbed Configuration

Like in the literature [2], [8], we build a scaled-down testbed with Dell PowerEdge servers connected to two PDUs, labeled as PDU#1 and PDU#2, respectively. In our scaled-down system, each server is considered as a "rack". We show our testbed configuration in Table I, where the subscription amounts (i.e., guaranteed capacity) are based on corresponding tenant's power usage in our experiment. We use two off-the-shelf PDUs (AP8632 from APC [21]) with per-outlet metering capabilities. Each PDU has four participating tenants and one group of "other" tenants representing non-participating tenants. The total leased capacities of PDU#1 and PDU#2 are 750W and 760W, respectively. We assume that the two PDUs have a capacity of 715W and 724W, respectively, to achieve 5% oversubscription (e.g., 750W=715W*105%) [8]. We also consider a common oversubscription by 5% at the upper UPS, and hence the total power usage need to be capped at 1370W ($= \frac{715W+724W}{105\%}$).

B. Workloads

We consider a mixture of workloads in our experiments. Each workload is representative of a particular class of tenants in multi-tenant data centers, and they are typical choices in the prior studies [2], [7], [8].

Search. We implement the web search benchmark from CloudSuite [27] in two servers, each virtualized into three VMs. It is based on a Nutch search engine which benchmarks the indexing process. Our implementation uses one front-end and five index serving VMs.

Web serving. We use the web serving benchmark from CloudSuite [27] that implements a Web 2.0 social-event application using PHP. The front-end is implemented using a Nginx web sever, while the back-end is implemented using MySQL database on a separate server.

Word count and TeraSort. We implement both WordCount and TeraSort benchmarks based on Hadoop 2.6.4 with one master node and seven data nodes, hosted on eight VMs. In our experiment, WordCount processes a 15GB input file, while TeraSort sorts 5GB of data.

Graph analytics. We implement PowerGraph [28] on two servers (16GB memory each). A Twitter data set consisting of 11 million nodes from [29] is used as the input.

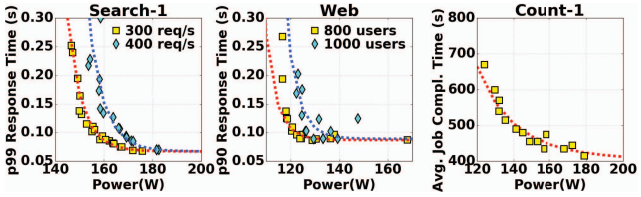


Figure 8. Power-performance relation at different workload levels.

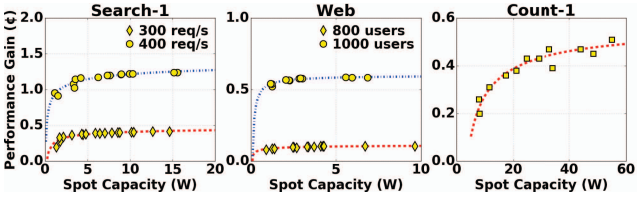


Figure 9. Performance gain versus spot capacity allocation.

C. Power and Performance Model

To participate in SpotDC, a tenant needs to assess the the performance improvement resulting from spot capacity. Towards this end, we first run the workloads at different power levels and workload intensities. Fig. 8 shows the power-performance relation of Search-1, Web and Count-1 for selected workload intensities. The other workloads also exhibit similar power-performance relations and are omitted for brevity.

The power-performance relation gives the potential performance improvement from spot capacity. To determine the bidding parameters, performance improvement needs to be converted into a monetary value. A tenant participating in SpotDC can decide the monetary value at its own discretion without affecting our SpotDC framework. For evaluation purposes, we convert the performance into monetary values following the prior research [8], [26]. Specifically, for sprinting tenants (Search and Web), we consider the following model: $c_{\text{tenant}} = a \cdot d$ if $d \leq d_{\text{th}}$, and $c_{\text{tenant}} = a \cdot d + b \cdot (d - d_{\text{th}})^2$ otherwise, where c_{tenant} measures the equivalent monetary cost per job, a and b are modeling parameters, and d is the actual performance (e.g., 99-percentile, or p99, latency for Search and p90 latency for Web) and d_{th} is the service level objective (SLO, 100ms for all sprinting tenants). The model indicates that the cost increases linearly with latency below the SLO threshold, and quadratically when latency is greater than the SLO to account for penalties of SLO violation. For opportunistic tenants running Hadoop and graph analytics, we use throughput (inverse of job completion time) as the performance metric and employ a linear cost model $c_{\text{tenant}} = \rho \cdot T_{\text{job}}$, where ρ is a scaling parameter and T_{job} is the job completion time.

Tenants can first estimate their performance "costs" with and without spot capacity, respectively, and then the difference is the performance gain (in dollars) brought by spot capacity. In our experiments, the cost parameters are chosen such that spot capacity will not cost more than directly subscribing guaranteed capacity. Further, we assume that Search tenants bid the highest price, Web tenants bid a medium price, and opportunistic tenants bid the lowest price. Fig. 9 shows an example of performance gain in terms of dollars (for using spot capacity per hour) under different spot capacity allocation for the Search-1, Web, and Count-1 tenants, respectively. The monetary values are small due to our scaled-down experimental setup. While tenants can decide bids freely, we consider the guideline in Section III-B3 as the tenants' default bidding approach.

D. Performance Metrics

For the operator, the key metric is the profit obtained through selling spot capacity. For tenants, performance improvement and extra cost for using spot capacity (compared to the no spot capacity case) are the two key metrics.[3]

Specifically, for sprinting tenants running interactive workloads, we consider tail latency: p99 latency for the two search tenants, and p90 latency for the web tenant (as p90 latency is the only metric reported by our load generator). For opportunistic tenants running delay-tolerant workloads, throughput is used as the performance metric: data processing rate for WordCount and TeraSort tenants, and node processing rate for the GraphAnalytics tenant.

V. Evaluation Results

In this section, we present the evaluation results based on our testbed and simulations. Our results highlight that spot capacity can greatly benefit both the operator and tenants: compared to the no spot capacity case, the operator can earn an extra profit by 9.7%, and tenants can improve performance by 1.2–1.8x (on average) while keeping the additional costs low (as low as 0.5%).

A. Execution of SpotDC

For our first experiment, we execute SpotDC in our testbed for 20 minutes divided evenly into 10 time slots. For clarity, we only show the results for tenants served by PDU#1. To show variations of spot capacity availability over the 10 time slots, we create a synthetic trace with a higher volatility for the non-participating tenants' power. Sprinting tenants bid for spot capacity when they would otherwise have SLO violations due to high workloads, while opportunistic tenants process data continuously and would like spot capacity to speed up processing.

1) Spot capacity allocation and market price: Fig. 10 shows the traces of spot capacity allocation (top figure) and market price (bottom figure). As the synthetic power trace is more volatile than the actual usage [1], [7], spot capacity prediction is assumed to be perfect. Later, we will predict spot capacity as presented in Section III-C.

We see that, whenever sprinting tenants participate, they receive most of their requested spot capacity, while opportunistic tenants may be priced out. The reason is that spring

[3]There is no extra server cost for using spot capacity (Section II-C).

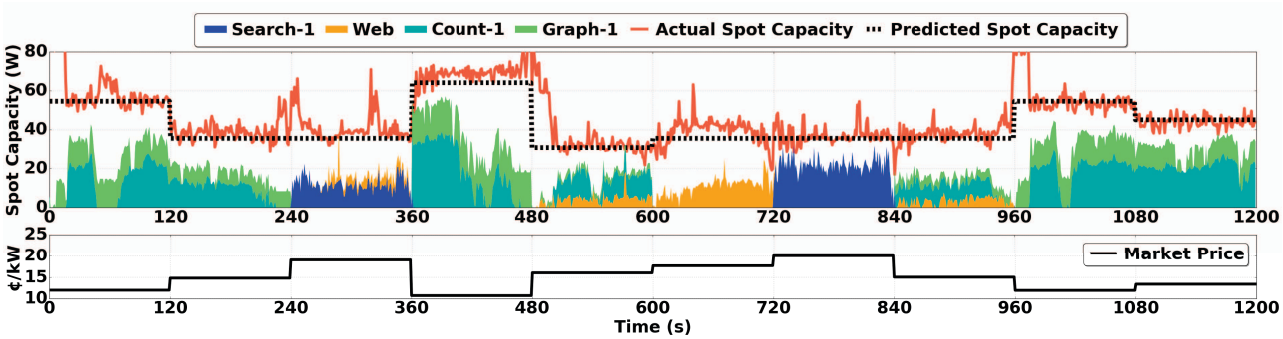


Figure 10. A 20-minute trace of power (at PDU#1) and price. The market price increases when sprinting tenants participate (e.g., starting at 240 and 720 seconds), and decreases when more spot capacity is available (e.g., starting at 360 seconds).

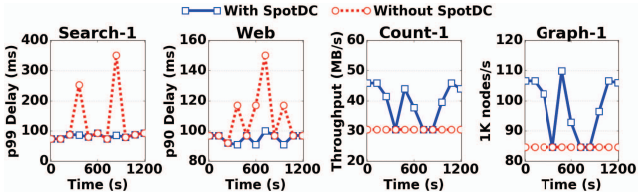


Figure 11. Tenants' performance. Search-1 and Web meet SLO of 100ms, while Count-1 and Graph-1 increase throughput.

tenants need spot capacity more urgently to meet their SLOs and hence bid a higher price. This is also reflected in the market price trace, from which we see that the sprinting tenants' participation drives up the price given the same spot capacity availability (e.g., 120–240 seconds versus 240–360 seconds). In addition, the market price decreases when more spot capacity is available (e.g., 0–120 seconds versus 120–240 seconds). Lastly, we notice that the actual spot capacity allocation is less than the available capacity due to multi-level capacity constraints. This also confirms that, even without conservative prediction, using spot capacity does not introduce additional power emergencies.

2) Tenant performance: We show the performance trace in Fig. 11. We see that the Search-1 and Web tenants can successfully avoid SLO (i.e., 100ms in our experiment) violations by receiving additional power budgets from the spot capacity market. Meanwhile, Count-1 and Graph-1 tenants can also opportunistically improve their throughput (by up to 1.5x).

B. Evaluation over Extended Experiments

Our next set of experiments seek to assess the long-term cost and performance. To do this, we extend our 20-minute experiment to one year via simulations. We use the scaled power trace collected from a large multi-tenant data center as the non-participating tenants' power usage. We also collect and scale the request arrival trace from Google services [30] for sprinting tenants, and back-end data processing trace collected from a university data center for opportunistic tenants (anonymized for review). We consider that the sprinting tenants need spot capacity during high traffic periods for around 15% of the times. Opportunistic tenants only lease guaranteed capacity to keep minimum processing rates, and need spot capacity for speed-up for around 30% of the time slots. We keep an average of approximately 15% of the total guaranteed capacity subscription as spot capacity, while we will vary the settings later. To evaluate SpotDC, we consider comparisons to the following two baselines.

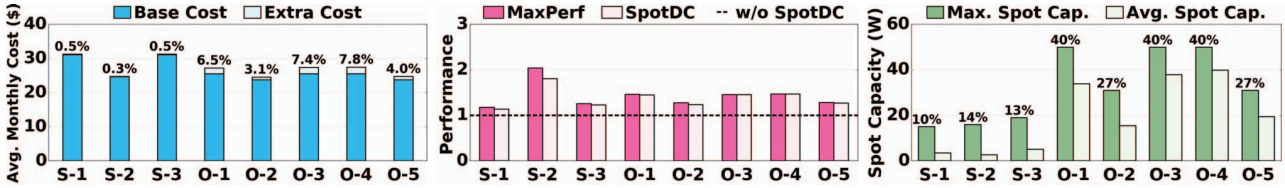
PowerCapped**:** No spot capacity is provisioned, and tenants cap their power below the guaranteed capacity at all times. This is the status quo, and we use it as a reference to normalize cost, profit, and performance.

MaxPerf**:** In this case, the data center operator fully controls all the servers as if in an owner-operated data center, and allocates spot capacity to maximize the total performance gain (as in [9]). There is no payment between the tenants and operator in MaxPerf.

1) Cost and performance: We show in Fig. 12(a) the total cost for tenants (baseline cost under PowerCapped plus extra spot capacity cost), while Fig. 12(b) shows the resulting performance of using spot capacity normalized to that with PowerCapped. Tenants' cost includes spot capacity payment and the increased energy bill. We use inverse of tail latency/job completion time to indicate tenants' performance. The performance is averaged over all the time slots whenever tenants need spot capacity. We see that by using SpotDC, tenants can achieve a performance very close to MaxPerf while the cost increase is only marginal (no more than than 0.5% for sprinting tenants). Opportunistic tenants have a higher percentage of cost increase, because they demand more spot capacity and bid more frequently (30% of the times).

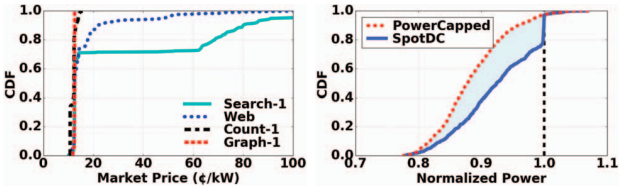
Fig. 12(c) shows each tenant's maximum and average spot capacity usage, in percentage of their guaranteed capacity subscriptions (Table I). In general, sprinting tenants receive less spot capacity (in percentage), because they are more performance-sensitive and hence do not oversubscribe their guaranteed capacity as aggressively as opportunistic tenants. However, if PowerCapped is used without spot capacity, tenants' capacity subscription costs will increase by 10-40% in order to maintain the same performance, because tenants have to reserve enough capacity to support their maximum power usage (e.g., 10% more capacity for Search-1).

Finally, we note that spot capacity is provisioned at no



(a) Power subscription cost (b) Normalized performance (c) Spot capacity usage

Figure 12. Comparison with baselines. Tenants' performance is close to MaxPerf with a marginal cost increase.



(a) Market price (b) Aggregate power usage

Figure 13. CDFs of market price and aggregate power. (a) Sprinting tenants bid and also pay higher prices than opportunistic tenants. (b) SpotDC improves power infrastructure utilization.

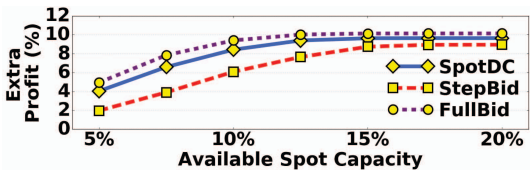
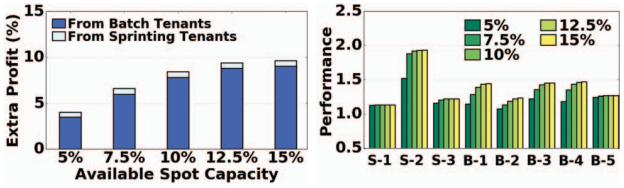


Figure 14. Comparison with other demand functions under different spot capacity availabilities.



(a) Operator's extra profit (b) Tenants' perfromance

Figure 15. Impact of spot capacity availability. With spot capacity, the market price goes down, the operator's profit increases, and tenants have a better performance.

additional cost for the data center operator, except for the negligible capital expense for over-provisioning rack-level capacity to support additional power budgets. In our calculation, we set US$0.4 per watt for rack capacity and amortize it over 15 years [9], [21]. We find that, by using SpotDC, the operator's net profit increases by 9.7% compared to the PowerCapped baseline.

2) Market price and power utilization: Fig. 13(a) shows the CDF of market prices for participating tenants in PDU#1. As expected, opportunistic tenants bid and have lower prices than sprinting tenants, although both types of tenants can avoid high costs of leasing additional guaranteed capacity. In our setting, opportunistic tenants will not bid higher than the amortized cost of guaranteed capacity (around US$0.2/kW/hour), while sprinting tenants are willing to pay more to avoid SLO violations.

In Fig. 13(b), we show the CDF of UPS-level power consumption normalized to the designed UPS capacity. SpotDC can greatly increase the power infrastructure utilization compared to PowerCapped. Since both the PDUs and UPS are oversubscribed in our setting, there exist occasional power emergencies (i.e., exceeding the UPS capacity), but these are handled through separate mechanisms [8] beyond our scope. In any case, spot capacity does not introduce additional emergencies, because it is offered *only* when there is unused capacity at the shared PDUs and UPS.

C. Other Demand Functions

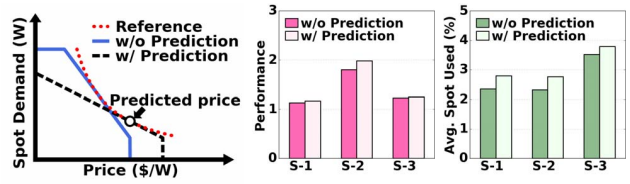
An important design choice in SpotDC is the demand function. To understand its impact, we consider two alternatives: StepBid, where tenants bid a step function for each participating rack, and FullBid, which solicits the complete demand curve for each participating rack. We perform the comparisons using the same setup as in Section V-B, and we also vary the average amount of available spot capacity (measured in percentage of total guaranteed capacity), by keeping the tenants' workloads unchanged and adjusting the shared PDU capacity.

We see from Fig. 14 that SpotDC outperforms StepBid (especially when spot capacity is scarce) and meanwhile is close to FullBid in terms of the operator's profit, justifying the choice of our demand function. The extra profit saturates when the average amount of spot capacity exceeds 15%, because tenants' demands are (almost) all met. By using SpotDC, tenants also receive a better performance than using StepBid, because the operator can partially satisfy their demands whereas StepBid only allows a binary outcome (i.e., either all or zero demand is satisfied). This result is omitted due to space limitations.

D. Sensitivity Study

We now investigate how sensitive SpotDC is against: available spot capacity, tenants' bidding, spot capacity prediction, and system scale.

1) Available spot power: In Fig. 15 we study the impact of amount of available spot capacity. For this, we keep the tenants' setup unchanged, and vary the operator's oversubscription at the PDUs to alter the available spot capacity. The spot capacity availability is measured in percentage of the total subscribed capacity. In Fig. 15(a), we show that the operator's extra profit increases with spot capacity availability, as the operator can get more money by selling



(a) (b) (c)

Figure 16. Impact of bidding strategies. With price prediction, sprinting tenants get more spot capacity and better performance.

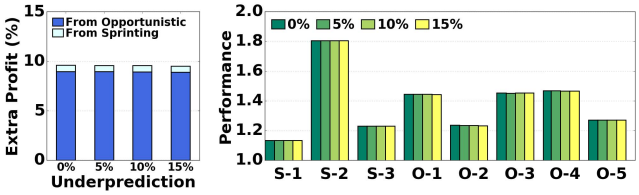


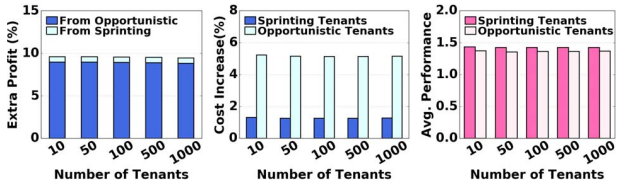
Figure 17. Impact of spot capacity under-prediction.

more spot capacity. Fig. 15(b) shows tenants' performance increases with spot capacity availability.

2) Tenants' bidding strategy: Tenants can bid for spot capacity on demand differently. For example, tenants may predict the price and set their bids accordingly. As illustrated in Fig. 16(a), we assume that sprinting tenants bid with a perfect knowledge of market price. The way opportunistic tenants bid remain the same. We see from Figs. 16(b) and 16(c) that through a more strategic bidding, sprinting tenants gain more spot capacity and increase their performance (without additional costs). Nonetheless, the operator's profit is not considerably affected (within 0.05%), since spot capacity is offered with no extra operating expenses at all. There can be many alternative bidding strategies for tenants, which are beyond our focus.

3) Spot capacity prediction: Perfectly predicting spot capacity is challenging. To avoid power emergencies, the operator can conservatively estimate the available spot capacity (i.e., under-prediction). In Fig. 17, we study the impact of spot capacity under-prediction, by multiplying the spot capacity (at both PDU/UPS levels) with an under-prediction factor. For example, 15% under-prediction means that the operator multiplies the originally predicted spot capacity by 0.85. We see that under-prediction has nearly no impact on the operator's extra profit and tenants' performance. The reason is that even without under-prediction, not all spot capacity is used up under a profit-maximizing price, as shown in Fig. 10, due to practical constraints (e.g., multi-level power capacity).

4) Larger-scale simulation.: We now extend our evaluation to a larger-scale simulation by increasing the number of tenants to up to 1,000 (a hyper-scale data center). We keep the same tenant composition as shown in Table I. Tenants' power subscriptions and the PDU/UPS capacity are both scaled up proportionally to those listed in Table I. For the newly added tenants, we randomly scale up/down workloads and performance cost models by up to 20% to reflect tenant diversity.



(a) (b) (c)

Figure 18. Impact of number of tenants. (a) Operator's profit. (b) Tenants' cost. (c) Tenant's performance.

The results are normalized to those obtained using PowerCapped (without offering spot capacity) and shown in Fig. 18. For clarity, we only show the results averaged over all the participating tenants. We see that as the number of tenants increases, the normalized results are fairly stabilized and consistent with our scaled-down evaluation: compared to PowerCapped, SpotDC increases the operator's profit by 9.7%, while tenants improve performance (by 1.4x on average) at a marginal cost.

VI. Related Work

Data center energy management has received considerable attention in the last decade. For example, numerous techniques have been proposed to improve server energy proportionality [31], [32], to jointly manage IT and non-IT systems [4], [33], and to exploit spatial diversities [34]–[36]. In addition, renewable-powered data centers are also emerging to cut carbon footprint [37], [38].

Maximizing data center infrastructure utilization is another focal point of research. The prior work focuses on power oversubscription e.g., [1], [7], [19], [39]. Other work looks at handling cooling emergencies through geographic load balancing [4] and phase changing materials [33]. Further, recent work also seeks to improve infrastructure utilization through dynamic power routing [9], soft fuse [10], among others.

Additionally, computational sprinting is emerging to boost performance. Initially proposed for processors [40], it is also studied at a data center level [41]. More recently, sprinting is extended to a shared rack to coordinate sprinting activities using game theory [42]. It allows the aggregate power demand to temporarily exceed the shared capacity (area "B" in Fig. 2(b)), whereas we exploit spot capacity (area "C" in Fig. 2(b)) based on demand function bidding.

Our work focuses on multi-tenant data centers and significantly differs from the work above. In particular, the key challenge our work addresses is to coordinate spot capacity allocation at scale, leading to a new market approach.

Market-based resource allocation has been studied in other contexts, such as processor design [20], [43], power markets [25], wireless spectrum sharing [16], [44], among others. These studies focus on different contexts with different design goals/constraints than our work (e.g., fairness for server/processor sharing [20], [43]).

Much of the research on multi-tenant data centers focuses on incentive mechanisms for energy cost saving [23], [24], demand response [45], and power capping [8]. In all these works, tenants are incentivized to cut tenant-level power and hence incur a performance loss, whereas we focus on improving performance by exploiting spot capacity.

VII. Conclusion

In this paper, we show how to exploit spot capacity in multi-tenant data centers to complement guaranteed capacity and improve power infrastructure utilization. We propose a novel market, called SpotDC, that leverages demand function bidding to extract tenants' demand elasticity for spot capacity allocation. We evaluate spot capacity based on both testbed experiments and simulations: compared to the no spot capacity case, the operator increases its profit (by 9.7%), while tenants improve performance (by 1.2–1.8x on average, yet at a marginal cost).

Acknowledgment

This work was supported in part by the U.S. NSF under grants CNS-1551661, CNS-1565474, CNS-1518941, CPS-154471, ECCS-1610471, and AitF-1637598.

References

[1] Q. Wu, Q. Deng, L. Ganesh, C.-H. R. Hsu, Y. Jin, S. Kumar, B. Li, J. Meza, and Y. J. Song, "Dynamo: Facebook's data center-wide power management system," in *ISCA*, 2016.

[2] D. Wang, C. Ren, and A. Sivasubramaniam, "Virtualizing power distribution in datacenters," in *ISCA*, 2013.

[3] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *ISCA*, 2007.

[4] I. Manousakis, I. n. Goiri, S. Sankar, T. D. Nguyen, and R. Bianchini, "Coolprovision: Underprovisioning datacenter cooling," in *SoCC*, 2015.

[5] X. Fu, X. Wang, and C. Lefurgy, "How much power oversubscription is safe and allowed in data centers," in *ICAC*, 2011.

[6] L. A. Barroso, J. Clidaras, and U. Hoelzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan & Claypool, 2013.

[7] G. Wang, S. Wang, B. Luo, W. Shi, Y. Zhu, W. Yang, D. Hu, L. Huang, X. Jin, and W. Xu, "Increasing large-scale data center capacity by statistical power control," in *EuroSys*, 2016.

[8] M. A. Islam, X. Ren, S. Ren, A. Wierman, and X. Wang, "A market approach for handling power emergencies in multi-tenant data center," in *HPCA*, 2016.

[9] S. Pelley, D. Meisner, P. Zandevakili, T. F. Wenisch, and J. Underwood, "Power routing: Dynamic power provisioning in the data center," in *ASPLOS*, 2010.

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

[11] NRDC, "Scaling up energy efficiency across the data center industry: Evaluating key drivers and barriers," *Issue Paper*, Aug. 2014.

[12] Apple, "Environmental responsibility report," 2016.

[13] D. S. Palasamudram, R. K. Sitaraman, B. Urgaonkar, and R. Urgaonkar, "Using batteries to reduce the power costs of internet-scale distributed networks," in *SoCC*, 2012.

[14] Uptime Institute, "Data center industry survey," 2016.

[15] Intel, "Rack scale design: Architectural requirement specifications," *Document Number: 332937-003*, Jul. 2016.

[16] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J.Sel. A. Commun.*, vol. 23, no. 2, pp. 201–220, Sep. 2006.

[17] Amazon, "EC2 spot instances," http://aws.amazon.com/ec2/spot-instances/.

[18] D. Lo, L. Cheng, R. Govindaraju, L. A. Barroso, and C. Kozyrakis, "Towards energy proportionality for large-scale latency-critical workloads," in *ISCA*, 2014.

[19] L. Liu, C. Li, H. Sun, Y. Hu, J. Gu, T. Li, J. Xin, and N. Zheng, "Heb: Deploying and managing hybrid energy buffers for improving datacenter efficiency and economy," in *ISCA*, 2015.

[20] M. Guevara, B. Lubin, and B. C. Lee, "Navigating heterogeneous processors with market mechanisms," in *HPCA*, 2013.

[21] APC, "Metered-by-outlet with switching rack PDU," http://www.apc.com/shop/us/en/products/Rack-PDU-2G-Metered-by-Outlet-with-Switching-ZeroU-30A-100-120V-24-5-20R/P-AP8632.

[22] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, Dec. 2008.

[23] C. Wang, N. Nasiriani, G. Kesidis, B. Urgaonkar, Q. Wang, L. Y. Chen, A. Gupta, and R. Birke, "Recouping energy costs from cloud tenants: Tenant demand response aware pricing design," in *eEnergy*, 2015.

[24] M. A. Islam, H. Mahmud, S. Ren, and X. Wang, "Paying to save: Reducing cost of colocation data center via rewards," in *HPCA*, 2015.

[25] R. Johari and J. N. Tsitsiklis, "Parameterized supply function bidding: Equilibrium and efficiency," *Oper. Res.*, vol. 59, no. 5, pp. 1079–1089, Sep. 2011.

[26] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," *SIGCOMM Comput. Commun. Rev.*, 2012.

[27] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, M. Alisafaee, D. Jevdjic, C. Kaynak, A. D. Popescu, A. Ailamaki, and B. Falsafi, "Clearing the clouds: A study of emerging scale-out workloads on modern hardware," in *ASPLOS*, 2012.

[28] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein, "Graphlab: A new parallel framework for machine learning," in *Uncertainty in Artificial Intelligence (UAI)*, 2010.

[29] R. Zafarani and H. Liu, "Social computing data repository at ASU," 2009. [Online]. Available: http://socialcomputing.asu.edu

[30] Google, "Cluster workload traces," https://code.google.com/p/googleclusterdata/.

[31] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," in *INFOCOM*, 2011.

[32] D. Meisner, C. M. Sadler, L. A. Barroso, W.-D. Weber, and T. F. Wenisch, "Power management of online data-intensive services," in *ISCA*, 2011.

[33] M. Skach, M. Arora, C.-H. Hsu, Q. Li, D. Tullsen, L. Tang, and J. Mars, "Thermal time shifting: Leveraging phase change materials to reduce cooling costs in warehouse-scale computers," in *ISCA*, 2015.

[34] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *SIGCOMM*, 2009.

[35] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Greening geographical load balancing," in *SIGMETRICS*, 2011.

[36] S. Lee, R. Urgaonkar, R. Sitaraman, and P. Shenoy, "Cost minimization using renewable cooling and thermal energy storage in CDNs," in *ICAC*, 2015.

[37] C. Li, Y. Hu, L. Liu, J. Gu, M. Song, X. Liang, J. Yuan, and T. Li, "Towards sustainable in-situ server systems in the big data era," in *ISCA*, 2015.

[38] I. Goiri, W. Katsak, K. Le, T. D. Nguyen, and R. Bianchini, "Parasol and greenswitch: managing datacenters powered by renewable energy," in *ASPLOS*, 2013.

[39] D. Wang, C. Ren, A. Sivasubramaniam, B. Urgaonkar, and H. Fathy, "Energy storage in datacenters: what, where, and how much?" in *SIGMETRICS*, 2012.

[40] A. Raghavan, Y. Luo, A. Chandawalla, M. Papaefthymiou, K. P. Pipe, T. F. Wenisch, and M. M. K. Martin, "Computational sprinting," in *HPCA*, 2012.

[41] W. Zheng and X. Wang, "Data center sprinting: Enabling computational sprinting at the data center level," in *ICDCS*, 2015.

[42] S. Fan, S. M. Zahedi, and B. C. Lee, "The computational sprinting game," in *ASPLOS*, 2016.

[43] X. Wang and J. F. Martinez, "ReBudget: Trading off efficiency vs. fairness in market-based multicore resource allocation via runtime budget reassignment," in *ASPLOS*, 2016.

[44] S. Ha, S. Sen, C. Joe-Wong, Y. Im, and M. Chiang, "Tube: time-dependent pricing for mobile data," in *SIGCOMM*, 2012.

[45] L. Zhang, S. Ren, C. Wu, and Z. Li, "A truthful incentive mechanism for emergency demand response in colocation data centers," in *INFOCOM*, 2015.