The Need for Speed and Stability in Data Center Power Capping

Arka A. Bhattacharya, David Culler
University of California, Berkeley
Berkeley, CA, USA
{*arka,culler*}*@eecs.berkeley.edu*

Aman Kansal
Microsoft Research
Redmond, WA, USA
kansal@microsoft.com

Sriram Govindan,Sriram Sankar
Microsoft Corporation
Redmond, WA, USA
{*srgovin,sriram.sankar*}*@microsoft.com*

***Abstract*—Data centers can lower costs significantly by provisioning expensive electrical equipment (such as UPS, diesel generators, and cooling capacity) for the actual peak power consumption rather than server nameplate power ratings. However, it is possible that this under-provisioned power level is exceeded due to software behaviors on rare occasions and could cause the entire data center infrastructure to breach the safety limits. A mechanism to *cap* servers to stay within the provisioned budget is needed, and processor frequency scaling based power capping methods are readily available for this purpose. We show that existing methods, when applied across a large number of servers, are not fast enough to operate correctly under rapid power dynamics observed in data centers. We also show that existing methods when applied to an open system (where demand is independent of service rate) can cause cascading failures in the software service hosted, causing the service performance to fall uncontrollably even when power capping is applied for only a small reduction in power consumption. We discuss the causes for both these short-comings and point out techniques that can yield a safe, fast, and stable power capping solution. Our techniques use admission control to limit power consumption and ensure stability, resulting in orders of magnitude improvement in performance. We also discuss why admission control cannot replace existing power capping methods but must be combined with them.**

***Keywords*-power capping; admission control; frequency scaling;**

I. Introduction

The cost of provisoning power in data centers is a very large fraction of the total cost of operating a data center [1], [2], [3] ranking just next to the cost of the servers themselves. *Provisioning* costs include the cost of infrastructure for sourcing, distribution and backup for the peak power capacity (measured in $/kW). These are higher than the *consumption* costs paid per unit of energy actually consumed (measured in $/kWh) over the life of a data center. Provisioned capacity and related costs can be reduced by minimizing the peak power drawn by the data center. A lower capacity saves on expenses in utility connection charges, diesel generators, backup batteries, and power distribution infrastructure within the data center. Lowering capacity demands is also greener because from the power generation standpoint, the cost and environmental impact for large scale power generation plants such as hydro-electric plants as well as green energy installations such as solar or wind farms, is dominated by the capacity of the plant rather than the actual energy produced. From the utility company perspective, providing peak capacity is expensive due to the operation of 'peaker power plants' which are significantly more expensive to operate and are less environmentally friendly than the base plants. Aside from costs, capacity is now is short supply in dense urban areas, and utilities have started refusing to issue connections to new data centers located in such regions. Reducing the peak power capacity required is hence extremely important.

The need to manage peak power is well understood and most servers ship with mechanisms for power capping [4], [5] that allow limiting the peak consumption to a set threshold. Further capacity waste can be avoided by coordinating the caps across multiple servers. For instance, when servers in one cluster or application are running at lower load, the power left unused could be used by other servers to operate at high power levels than would be allowed by their static cap. Rather than forcing a lower aggregate power level at all times, methods that coordinate the power caps dynamically across multiple servers and applications have been developed [6], [7], [8], [9], [10].

We identify two reasons why existing power capping methods do not adequately meet the challenge of power capping in data centers. The first is *speed*. We show through real world data center power traces that power demand can change at a rate that is too fast for the existing methods. The second is *stability*. We experimentally show that when hosting online applications, the system may become unstable if power capped. A small reduction in power achieved through existing power capping methods can cause the application latency to increase uncontrollably and may even reduce throughput to zero. We focus on the importance of the two necessary properties - *speed* and *stability*, and propose ways of achieving them and discuss the tradeoffs involved. Our observations are generic, and can be integrated into any power capping algorithm.

Specifically, the paper makes the following contributions:

- We quantify the benefit of using power capping to lower power provisioning costs in data centers through the analysis of a real world data center power trace.
- *Speed requirement:* From the same trace, we char-

978-1-4673-2154-9/12/$31.00

acterize the rates at which power changes in a data center. We make a case for one-step power controllers by showing that existing closed-loop techniques for coordinated power capping across a large number of servers may not be fast enough to handle data center power dynamics.
- *Stability requirement:* We show that existing power capping techniques do not explicitly shape demand, and can lead to instability and unexpected failures in online applications.
- We present admission control as a power capping knob. We demonstrate that admission control integrated with existing power capping techniques can achieve desirable stability characteristics, and evaluate the trade-offs involved.

II. Power Costs and Capping Potential

Most new servers ship with power capping mechanisms. System management software, such as Windows Power Budgeting Infrastructure, IBM Systems Director Active Energy Manager, HP Insight Control Power Managment v.2.0, Intel Node Manager, and Dell OpenManage Server Administrator, provide APIs and utilities to take advantage of the capping mechanisms. In this section we discuss why power capping has become a significant feature for data centers.

A. Power Provisioning Costs

The designed peak power consumption of a data center impacts both the capital expense of provisioning that capacity as well as the operating expense of paying for the peak since there is often a charge for peak usage in addition to that for energy consumed.

The capital expense (cap-ex) includes power distribution infrastructure as well as the cooling infrastructure to pump out the heat generated from that power, both of which depend directly on the peak capacity provisioned. The cap-ex varies from $10 to $25 per Watt of power provisioned [3]. For example, a 10MW data center spends about $100-250 million in power and cooling infrastructure. Since the power infrastructure lasts longer than the servers, in order to compare this cost as a fraction of the data center expense, we can normalize all costs over the respective lifespans. Amortizing cap-ex over the life of the data center (12-15 years [3], [2]), server costs over the typical server refresh cycles (3-4 years), and other operating expenses at the rates paid, the cap-ex is *over a third* of the overall data center expenses [11], [2]. This huge cost is primarily attributable to the expensive high-wattage electrical equipment, such as UPS batteries, diesel generators, and transformers, and is further exacerbated by the redundancy requirement mandated by data center availability stipulations.

The peak power use affects operating expenses (op-ex) as well. In addition to paying a per unit energy cost (typically quoted in $/kWh), there is an additional fee for the peak capacity drawn, even if that peak is used extremely rarely. Based on current utility tariffs [12] for both average and peak power, the peak consumption can contribute to as much as 40% of the utility bill [13]. Utility companies may also impose severe financial penalties for exceeding contracted peak power limits.

The key implication is that reducing the peak capacity required for a data center, and adhering to it, is highly beneficial.

B. Lower Cost Through Capping

Power capping can help manage peak power capacity in several ways. We describe some of the most common reasons to use it below.

1) Provisioning Lower Than Observed Peak: Probably the most widely deployed use case for power capping is to ensure safety when power is provisioned for the actual data center power consumption rather than based on server *nameplate ratings*. *Nameplate ratings* on servers denotes its maximum possible power consumption, computed as the sum of maximum power consumption of all the server subcomponents and a conservative safety margin.The nameplate rating on servers is typically much higher than the server's actual consumption. Since no workload actually exercises every server subcomponent at its peak rated power, the name plate power is not reached in practice. Data center designers thus provision for the *observed peak* on every server. The observed peak is the maximum power consumption measured on a server when running the hosted application at the highest request rate supported by the server. This observed peak can be exceeded after deployment due to software changes or events such as server reboots that may consume more than the previously measured peak power. Server level power caps can be used to ensure that the provisioned capacity is never exceeded and protect the circuits and power distribution equipment.

Server level caps do not eliminate waste completely. Setting the cap at each server to its observed peak requires provisioning the data center for the *sum of the peaks*, results in wasted capacity since not all servers operate at the peak simultaneously. Instead, it is more efficient to provision for the *peak of the sum* of server power consumptions, or equivalently, the estimated peak power usage of the entire data center. The estimate is based on previously measured data and may sometimes be exceeded. Thus a cap must be enforced at the data center level. Here, the server level caps will change dynamically with workloads. For instance, a server consuming a large amount of power need not be capped when some other server has left its power unused. However the former server may have to be capped when the other server starts using its fair share. Coordinated power capping systems [6], [7], [8], [9], [10] can be used for this.

Additionally, even the observed peak is only reached rarely. To avoid provisioning for capacity that will be left

unused most of the time, data centers may provision for the 99-th percentile of the peak power. Capping would be required for 1% of the time, which may be an acceptable hit on performance in relation to cost savings. If the difference in magnitude of power consumed at the peak and 99-th percentile is high, the savings can be significant. To quantify these savings, we present power consumption data from a section comprising of several thousand servers in one of Microsoft's commercial data centers that host online applications serving millions of users, including indexing and email workloads. The solid line in Figure 1 shows the distribution of power usage, normalized with respect to the peak consumption. If the 99-th percentile of the observed peak is provisioned for, the savings in power capacity can be over 10% of the data center peak. Capacity reduction directly maps to cost reductions.

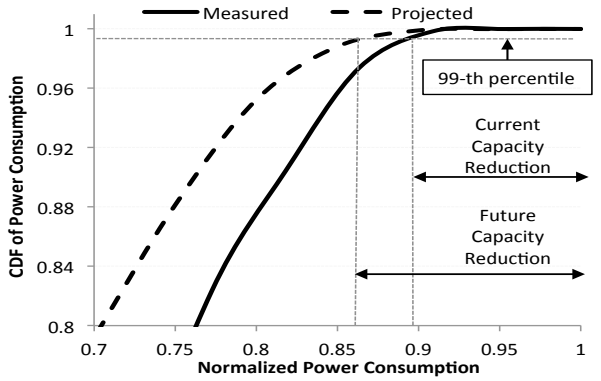


Figure 1. Cumulative distribution function (CDF) of power consumption for a cluster of several thousand servers in one of Microsoft's commercial data centers. Future capacity reduction refers to the power consumed by the same workload if hosted on emerging technology based servers.

Trends in server technology indicate that the margin for savings will increase further. Power characteristics of newer servers accentuate the difference between the peak and typical power (power consumed by a server under average load) usage because of their lower idle power consumption. Power measurement for an advanced development server at different CPU utilizations shows only 35% of peak consumption at idle, much lower than the over 50% measured in current generation servers. Using processor utilizations from the real world servers, we project the power usage of the same workloads on the future generation servers assuming that power scales with processor utilization [14] (the dashed curve in Figure 1). The present day data and technology trends both indicate a significant margin for savings.

2) UPS Charging: Large data centers use battery backups, also referred to as Uninterrupted Power Supplies (UPSs). UPSs provide a few minutes of power during which time the diesel generators may be powered up. After power is restored, the UPS consumes power to re-charge the batteries. This implies that the power capacity provisioned for a data center should not only provide for the servers and cooling equipment but also include an additional margin for battery charging. This additional capacity is almost always left unused since power failures are relatively rare. Even when power failures do happen, they may not occur at the time when data center power consumption is at its peak.

The capacity wasted due to reservation for battery charging can be avoided if the batteries are charged from the allocated server power capacity itself. Should the servers happen to be using their full capacity at recharging time, power capping is needed to reduce the server power consumption by a small amount and free up capacity for recharging batteries at a reasonable rate. Since power failures are rare, the performance impact of this capping is acceptable for many applications. Any data center that uses a battery backup can use power capping to reduce the provisioned power capacity.

3) Total Capital Expenses: Many power management methods are available to reduce server power consumption by turning servers off or using low power modes when unused. Using less energy however does not reduce the cost of the power infrastructure or the servers themselves. The amortized cost of the servers and power infrastructure can be minimized if the servers are kept fully utilized [15]. Workload consolidation can help achieve this. Suppose a data center is designed for a given high priority application and both servers and power are provisioned for the peak usage of that application. The peak workload is served only for a fraction of the day and capacity is left unused at other times. During those times, the infrastructure can be used to host low priority applications.

In this case capping is required on power, as well as other computational resources, at all times to ensure that the low priority application is capped to use only the resources left unused by the high priority applications and up to a level that does not cause performance interference with the high priority tasks. Since power is capped by throttling the computational resources themselves, the implementation may not require an additional control knob for power. However, settings on the throttling knobs should ensure that all resource limits and the power limit are satisfied. The end result is that in situations where low priority workloads are available, power capping can be used in conjunction with resource throttling to lower both power and server capacity requirements.

4) Dynamic Power Availability: There are several situations where power availability changes with time. For instance, if demand response pricing is offered, the data center may wish to reduce its power consumption during peak price hours. If the data center is powered wholly or partly through renewable energy sources such as solar or wind power, the available power capacity will change over time. Power capacity may fall due to brown-outs [16]. In this situation too, a power capping method is required to

track the available power capacity.

The above discussion shows that power capping can help save significant cost for data centers. However, existing power capping methods suffer from speed and stability limitations in certain practical situations. In the next sections we quantitatively investigate these issues and discuss techniques to enhance the existing methods for providing a complete solution.

III. Speed: Power Capping Latency

The actuation latency of power capping mechanisms is an important consideration. Server level power capping mechanisms, typically implemented in server motherboard firmware, change the processor frequency using dynamic voltage and frequency scaling (DVFS) until the power consumption falls below the desired level [4]. These local methods can operate very fast, typically capping power within a few milliseconds. However, capping speed can become an issue for coordinated power capping methods that dynamically adjust server caps across thousands or tens of thousands of servers [9], [8], [10]. To understand this issue in depth, we first study the temporal characteristics of data center power variations from the trace analyzed in Figure 1. We then quantify the required actuation latencies for a power capping mechanism, and compare it to the state-of-the-art.

A. Data Center Power Dynamics

Data center power consumption varies due to workload dynamics such as changes in the volume of requests served, resource intensive activities such as data backup or index updates initiated by the application, background OS tasks such as a disk scrubs or virus scans, or other issues such as simultaneous server restarts. We study the data center power trace previously shown in Figure 1 to quantify the rate of change of power.

Since capping is performed near peak power levels, only power increases that occur near peak usage matter for capping; power changes that are well below the peak, however fast, are not a concern. So we consider power increases that happen when power consumption is greater than the 95th percentile of the peak. We measure the rate of power increase, or *slope*, as the increase in normalized power consumption (if over the 95th percentile) during a 10 second window.

Figure 2 shows the CDF of the *slope*, normalized to the peak power consumption of the cluster. For most 10 second windows, power increases are moderate (less than 2% of the peak cluster power consumption). However, there exists power increases as high as 7% of the peak consumption over a 10 second window. To ensure protection and safety of electrical circuits during such extreme power surges, the power capping mechanism must be agile enough to reduce power consumption within a few seconds.

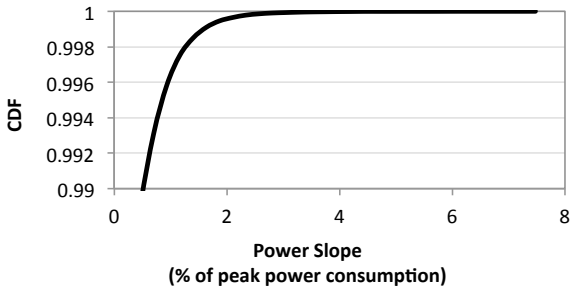


Figure 2. Cumulative distribution function (CDF) of power slope [increase in power conumption of the cluster over a 10 second window]. The slope is normalized to the peak power consumed by the cluster during the period of the study.

B. Power Control Latency

This section experimentally investigates the limits on how fast a power capping mechanism can throttle multiple servers using DVFS. The experiments were performed on three servers with different processors: Intel Xeon L5520 (frequency 2.27GHz, 4 cores), Intel Xeon L5640 (frequency 2.27GHz, dual socket, 12 cores with hyper-threading), and an AMD Opteron 2373EE (frequency 2.10GHz, 8 cores with hyper-threading). All servers under test were running Windows Server 2008 R2. Power was measured at fine time granularity using an Agilent 34411A digital multimeter placed in series with the server. The multimeter recorded direct current values at a frequency of 1000Hz, and root mean square was computed over discrete 20ms intervals where one interval corresponds to 1 cycle of the 50Hz AC power. Since in a practical power capping situation, the cap will likely be enforced when the servers are busy, in our experiment the servers were kept close to 100% utilization by running a multi-threaded synthetic workload. This kept the server near its peak consumption level from where power could be reduced using power capping APIs.

To estimate the fastest speed at which a data center power capping mechanism can operate, the latency to be considered is the total delay in determining the desired total data center power level, dividing it up into individual server power levels, sending the command to each server, the server executing the power setting command via the relevant API, and the actual power change taking effect (Figure 3). Since we are only interested in the lower limit on latency, we ignore the computational delays in computing the caps. A central power controller is assumed to avoid additional delays due to hierarchical architectures. In the following sections we investigate each of these latency components.

1) Network Latency in a data center: Table I shows the network latency of sending a packet between the controller (hosted within the data center network) and the power capping service at a server, for varying network distances.

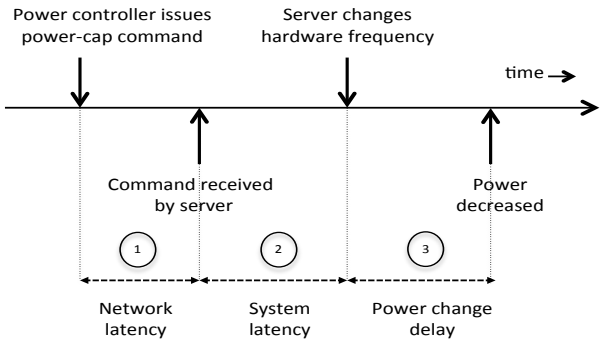


Figure 3. Timeline showing the smallest set of latency components for a coordinated power capping solution. Additional latency components may get added when the cap is enforced in a hierarchical manner such as in [8], [10].

This data was obtained using a Microsoft data center management tool, `PingMesh`, that allows measuring ICMP ping latencies across a data center network. The data shows that the average packet delay on a network is less than a millisecond. This latency component is hence not likely to be a concern for coordinated capping.

Table I
NETWORK LATENCIES IN A DATA CENTER.

Sender and Receiver placement	No. of samples	Avg (ms)	Std. dev (ms)
Within same rack	21	0.331	0.098
Within same aggregation switch	32	0.342	0.030
Under different aggr. switches	61	0.329	0.032

2) System Latency: Once a DVFS setting from the controller reaches a server, it is applied by calling the relevant APIs. In this experiment, the frequency was decreased from the maximum to minimum to obtain the highest resolution power change for measurement of latency. Low level frequency APIs offered through `powerprof.dll` in the Windows OS were used to avoid as much of the software stack delays as possible. The threads for applying and reading the frequency setting were set to higher priority so as to not be delayed due to the server workload. The latency incurred for changing the frequency ranged between 10-50ms for multiple runs on the different servers.

3) Power Change Delay: After the processor frequency changes there is an additional delay before the power drops to the new level at the wall outlet, due to factors such as capacitance in the server and the power supply circuits. This effect requires a fine time granularity power measurement. Figure 4 shows a sample power reading plot measured using the Agilent multimeter. The latency was found to be between 100ms and 300ms, for a frequency change from the maximum to the minimum, across multiple measurements over the three servers. The minimum latency was observed when the frequency was changed between two adjacent DVFS levels requiring a smaller change in power. The smallest latency across all adjacent frequency levels was 60ms. These measurements are similar to the fast capping latency of 125ms reported in commercial product datasheets [17].

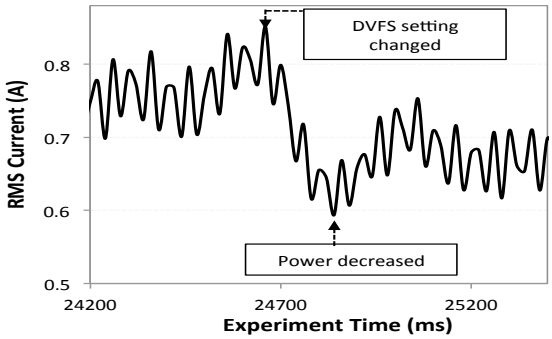


Figure 4. Typical latency between the hardware frequency change and power reduction. The current readings are rms values over discrete 20ms windows. The power decrease latency in this experiment was approximately 200ms.

4) Total Delay: A summary of the latency results is provided in Table II and totals to approximately 110ms to 350ms. This implies that for a feedback based controller, it takes approximately 110ms to 350ms for one iteration of a control loop. Much of this delay is coming from the power change at the server itself rather than the computational overhead or network delay of the coordinated capping algorithm.

Table II
SUMMARY OF ACTUATION LATENCIES FOR POWER CAPPING

Latency type	Approx. Latency
Network	$<$1ms
OS	10-50ms
Wall power change	100-300ms
Total	110-350ms

Implications: An important implication of the above measurements is that a feedback controller using multiple iterations can incur several seconds of delay. Many controllers use a hierarchy to scale to a large number of servers or to logically separate the power division among multiple applications in a data center [8], [10]. When feedback loops operate at multiple levels in the hierarchy, control theoretic stability conditions require that the lower layer control loop must converge before an upper layer loop can move on to the next iteration. Suppose the actuation latency is denoted as l (where $l \approx 110ms - 350ms$ from Table II) and the number of iterations required for convergence at the $i-th$ layer in the control hierarchy is n_i, then the total latency of the capping mechanism using N layers in the hierarchy

becomes:

$$l_{total} = l * \prod_{i=1}^{N} n_i$$

As an example, considering the two layer hierarchy ($N = 2$) with $n_1 = 6$ and $n_2 = 16$ used in [8], and plugging in the measured l value in the above equation, we would get a control latency of 10.56s to 33.6s. For the three layer hierarchy used in [10] and similar number of convergence iterations required, the latency will be even higher. While this latency is not a concern for adapting to the slow changes in workload levels that only cause the power to change every few minutes, these latencies are not acceptable for the fast power changes observed in real data centers (Figure 2).

Some of the power distribution components in the data center can handle capacity overages for a few seconds or even minutes [18], [19]. However, when power is changing at a rapid rate, the feedback based controllers cannot meet their stability conditions. The dynamics of the system being controlled must be slower than the convergence time of the controller. The requirement for stability implies that power should not change beyond measurement tolerance within the 10.56s or 33.6s control period. That however is not true since the power can change by as much as 7% of the data center peak power within just 10s, in real data centers (Figure 2).

C. Summary

The latency analysis above implies that feedback based controllers using multiple iterations are not fast enough to operate safely under the data center power dynamics. The design implication for power capping methods is that the system may not have time to iteratively refine its power setting after observing a capacity violation.

> **Observation 1:**
> *A safe power capping system should use a single step actuation to apply a power cap setting, such as using DVFS, that will conservatively bring the power down to well below the allowed limit (say, the lowest DVFS setting)*

The conservative setting is needed to avoid unsafe operation in the presence of model errors. Once power has been quickly reduced to a safe limit, feedback based controllers can be employed to iteratively and gradually increase power to the maximum allowed limit to operate at the best performance feasible within the available power capacity.

IV. Stability: Application Performance with DVFS Based Capping

It is well known that for system stability, the incoming *request rate* should be lower than the sustained *service rate* across the multiple servers hosting a given application [20]. This requirement is often the basis of capacity planning, such as for determining the number of servers required. The service rate is experimentally measured for a variety of requests served by the hosted online application and the number of servers is chosen to match or exceed the maximum expected request rate[1]. As request rate increases, more servers are added to the deployment. [2]

Under normal conditions, the service rate matches the request rate. However, whenever power capping is performed, power consumption of some server resource must be scaled down. Typically the processor power is scaled down using DVFS for practical reasons, though in principle, one could scale down the number of servers or some other resource as well. Regardless of the mechanism used to reduce power, engaging it reduces the service rate. The incoming request rate may or may not change when service rate is reduced. If the system is *closed*, where each user submits a new request only after the previous response is received, the request rate will fall to match the service rate. Batch processing systems such as Map-Reduce, HPC workloads, or large computationally intensive workloads can be closely approximated as a closed system. However, if the system is *open*, where the request rate is not directly affected by the service rate, a decrease in service rate due to power capping may not lead to a equivalent decrease in the request rate. Most web based online applications, such as web search, can be approximated as open systems since the requests are coming from a large number of users and new requests may come from new users who have not yet experienced the reduced service rate. Even users experiencing reduced service rate may not stop submitting new requests. Delays may even lead to rapid abort and retry.

Capping is enforced primarily when the system is at high power consumption. This happens when serving close to the peak demand that the system can support. Hence, the reduced service rate after capping is very likely to be lower than the demand at that time. Queuing theory says that response time shoots up uncontrollably in this situation. We experimentally demonstrate this in an open system.

Experiment: We use a web server hosting Wikipedia pages using MediaWiki[3], serving a 14 GB copy of the English version of Wikipedia, as an example of an open loop system. The front end is hosted on an Intel Nehalem Xeon X5550 based server (16 cores, 48GB RAM) running Apache/2.2.14 and PHP 5.3.5. The database uses mysql 14.14 hosted on another similar server. Both servers run Ubuntu 10.04 as the Operating System. HTTP requests for Apache are generated by a workload generator using the `libevent` library, which has support for generating thread-safe HTTP requests. Seven workload generators were used

[1]Power management methods may be employed to turn off or re-allocate unused servers when request rate is lower than the maximum rate that can be served.

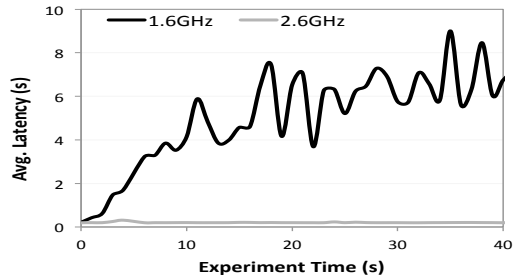
[2]The terms *request rate*, *demand* and *workload* have been used interchangeably in the subsequent sections

[3]http://www.mediawiki.org/

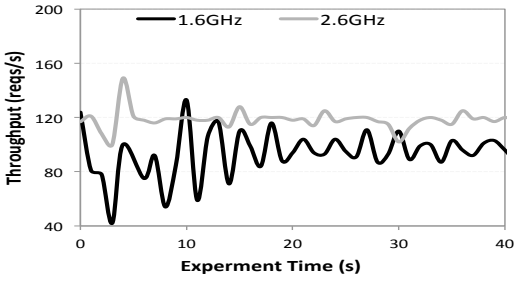
to ensure that the workload generators themselves were not a bottleneck. To avoid crippling disk latencies, and to ensure that all requests were served out of the back-end cache the workload generators repeatedly request the same wiki page. All the workload generator servers have their system time synchronized and log performance (throughput and latency) in separate files that are then aggregated.

We find the maximum service rate that our deployment can support by incrementally increasing the input request rate until performance, measured as the latency of a page retrieval, becomes unacceptable. The rate was 130 requests/s for this deployment, providing a stable latency of approximately 0.2s. We operate the system at a throughput of 120 requests/s, that is below the maximum supported service rate. To measure the effect of power capping we reduce power using DVFS, an existing power capping mechanism.

Observations: Figure 5 shows the impact on performance using DVFS based power capping. The gray curve shows the normal operation at 2.6GHz. The black curve shows the operation when the server is operated at a lower frequency but the incoming request rate is not changed. Throughput falls since the computational resource available is lowered. However, latency starts to increase uncontrollably to much higher values than the initial 0.2s, even though the input request rate is constant throughout (at 120 requests/s).



(a) Application Average Response Time



(b) Application Throughput

Figure 5. Effect of DVFS based power capping on throughput and latency in the experimental Wikipedia server.

Performance plummets by orders of magnitude in a relatively short time when operating at the lower frequency. This is expected since several undesirable effects start to manifest in this situation. First, any buffers in the system, such as in the network stack or the web server's application queue for incoming requests will get filled up and they will unnecessarily add to the latency without yielding any advantage on throughput [21]. Second, requests not served will be re-attempted, increasing the total number of requests coming into the system. Since some of the requests served will not be fresh requests but re-attempted ones, the total request service latency will increase. Even with a small reduction in service rate, the number of dropped requests will start piling up and the average latency will continue to rise, leading to the plummeting performance observed. Third, if semantically, each user activity consists of multiple requests (such as a accessing a web page may consist of accessing multiple embedded image and resource URLs from the web server), since some of the requests may have been dropped from each semantic activity, no user activity will have been served. This implies that a small reduction in power can actually render a system *unstable*.

Observation 2:
A stable power capping system when reducing the service capacity of a server through DVFS, should be able to implement a commensurate reduction in incoming application demand (through admission control)

V. Stable Power Capping with Admission Control

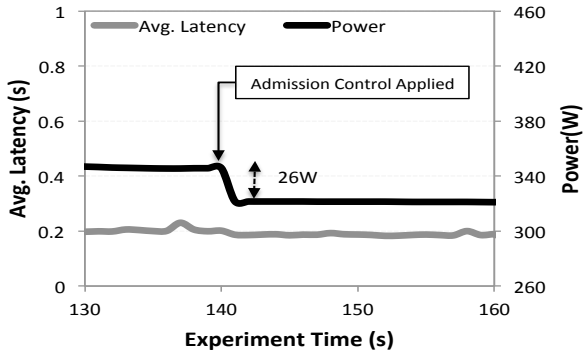
Admission control can be used with power capping mechanisms to reclaim stable behavior.

A. Admission Control and Power

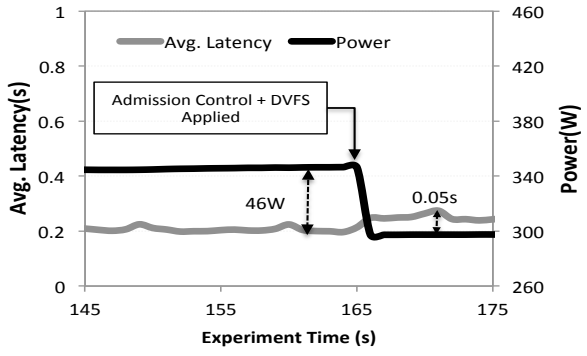
Power capping reduces the service rate, which can make a system unstable. To maintain stability, the input request rate should also be reduced within a modest time window, and admission control is one technique to achieve that. This would result in some users receiving a "request failed" message or a long wait before service, but the system will be able to serve the remaining workload within acceptable performance bounds.

If admission control is applied, the amount of work performed, and correspondingly the amount of computational resource used, is reduced. This implicitly reduces the power consumption since the processor has more idle cycles that it can spend in lower power sleep states. Intuitively, this suggests that admission control can be used as an alternative power capping mechanism. We experimentally verify that this intuition is correct. However, there are practical issues that prevent admission control from directly replacing DVFS based or other existing power capping mechanisms.

Experiment: Using the same experimental testbed as used in Section IV, we measure the power reduction provided using admission control. We implemented admission control using the `iptables` utility and selectively filter out



(a) Power reduction and average request service latency before and after admission control is applied.



(b) Variation of latency and power with time when admission control is used along with DVFS. The small increase in latency is discussed in Section V-C.

Figure 6. Power and average latency variation when using only admission control, and admission control+DVFS

requests from some of the workload generators (based on IP address) to reduce the incoming request rate to the Wikipedia server[4].

As in Figure 5, suppose the server is originally operating at 120 requests/s (at processor frequency 2.6GHz). Suppose the desired power reduction can be achieved using DVFS by lowering the processor frequency to 1.6GHz. The throughput sustained at this lower frequency is measured to be 85 requests/s and the reduction in power is 46W. Keeping the input request rate at 120 requests/s, we enforce admission control to allow only 85 requests/s to be presented to the server. Figure 6(a) shows the impact on power when admission control is applied at the time tick of 140s (approx). As intuitively expected, admission control does reduce power and can be used as a power capping mechanism. However, the reduction in power is only 26W (instead of 46W that was achieved using DVFS for the same reduction in throughput).

B. Practical Issues with Admission Control

Power Efficiency: To investigate the power difference further, we measure the power consumption at varying throughput levels at two different DVFS frequency settings. Figure 7 shows the power measurements. The key take-away is that the same throughput can be served at a lower power level using the lower frequency, though the peak throughput that can be served is lower at the lower frequency. A difference of 20W is apparent. This is because the lower frequency is more energy efficient. As is known from DVFS design, processor power increases with the cube of frequency and even the total server power has been measured to increase super-linearly with frequency [20]. Since the number of processor cycles available for computation increases only linearly with frequency, this makes lower frequencies more energy efficient at a given throughput.

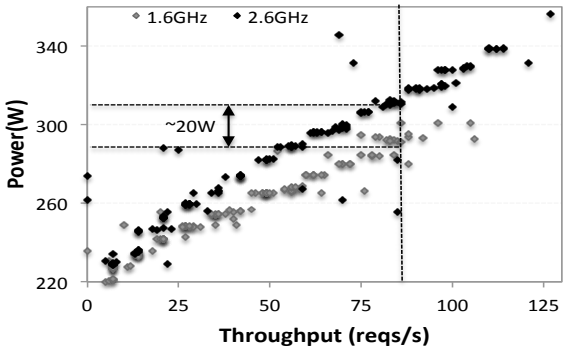


Figure 7. Power vs throughput in the stable region of the Wikipedia front end server at 2.6GHz and 1.6GHz. Note the extra reduction in power available by using DVFS in addition to admission control

The observation above indicates that while admission control is required for stability, DVFS is more efficient from a power perspective. Hence a practical power capping system must use DVFS in combination with admission control to achieve stability without sacrificing efficiency. Figure 6(b) shows the effect on power when both mechanisms are applied simultaneously (around time tick 163s in the figure). The throughput achieved (not shown) is the same in both Figures 6(a) and 6(b) but the power is capped by a greater amount in Figure 6(b). As technology improves and idle power consumption falls further, the above power difference may be reduced since the higher frequency state with more idle cycles will likely become more power efficient as well.

Delay: Another practical issue that requires DVFS is the effect of queuing delays. If the application has a large buffer for incoming requests, then a large number of requests will be served from that queue. Admission control will reduce the

[4]In practice, admission control may be implemented by the application or in the load balancers, among other options. Our purpose in this paper is only to study the effect of admission control on power and performance, and the above implementation suffices.

incoming request rate but the service rate in the servers may remain high while the queues are being emptied, leading to a delay before the power is actually reduced. This is a concern when speed of the capping mechanism is important.

Safety: Admission control reduces the workload offered to the server but does not force the server power to be lowered. While power is expected to fall with reduced workload, in some cases it may not, such as when the server is running a background virus scan or operating system update. With DVFS all computations related to the workload or background tasks will be throttled down simultaneously to reduce power.

C. Application Latency

Another metric worth comparing between Figures 6(a) and 6(b) is the application performance in terms of latency. While throughput reduction is the same and stability is ensured in both cases, the latency shows a small increase when DVFS and admission control are combined. Suppose servicing each request requires an average of n_r processor cycles. Then the latency component attributable to the processor, denoted l_{cpu} can be computed as $l_{cpu} = n_r/f_i$ where f_i is the processor frequency in use at the $i-$th DVFS setting. When DVFS is used to reduce the frequency from a higher value f_0 to a lower value f_1, clearly l_{cpu} will rise. Other latency components such as the network round trip delay, queuing delay, and the latency of accessing the backend storage are not significantly affected by DVFS and the increase in l_{cpu} shows up as a small increase in overall application layer latency.

D. Closed Loop Systems

Admission control is also applicable in a closed application scenario, when the latency increase due to processor based power capping mechanisms (DVFS) is not desirable. Unlike an open system, a closed system remains stable when service rate is reduced. However, both latency and throughput degrade due to the slower clock speed and (if applicable) additional buffering delay. In contrast to DVFS, employing admission control could idle system components to achieve the same power reduction while keeping latency unaffected. Admission control, being less efficient than DVFS in terms of throughput per unit power, would entail a lower throughput. So the latency advantage comes at the price of additional loss of throughput.

E. Summary

From the above analysis , we conclude that using admission control alone leads to a smaller power reduction, higher possible actuation delay, and the possibility of unforseen software events which might cause a power spike.

Observation 3:
Admission control, while necessary, should be used in conjunction with DVFS to increase its effectiveness as a power capping knob.

The design implication is that power capping techniques should coordinate with admission control agents, such as load balancers, to maintain application stability.

VI. Related Work

Server level power capping methods [4] have been developed to throttle processor frequency in response to hardware metered power readings at millisecond granularity. Similar techniques for virtualized servers have been investigated in [22], [23], and use processor utilization capping in addition to frequency control. Since single servers methods do not make efficient use of the overall data center capacity, coordinated power budgeting across multiple servers has also been considered [6], [9], [7], [8], [10]. We build on these methods to address additional challenges. The coordinated methods rely on multiple feedback control iterations that, as we show, may not satisfy convergence conditions under rapid data center power dynamics. Stability concerns with open loop workloads are also not considered in these works. The control of processor frequency in open and closed loop system was considered in [20] but for energy efficiency rather than power capping, and hence the stability issue that arises in capping was not relevant in that context.

Admission control in web servers has also been studied in depth. Admission control methods drop requests to prevent the server from getting overloaded [24], [25], [26] and maintain acceptable performance. Feedback control and queuing theoretic algorithms that carefully trade off the number of dropped requests and performance have also been studied [27], [28]. Processor frequency management to maximize energy efficiency for variable incoming request rates along with admission control have been considered in [29], [30]. Techniques to implement admission control by preventing new TCP connections or selectively blocking requests based on the HTTP headers were presented in [31]. However, the integration of processor frequency management and admission control has not been considered for power capping. We discuss the desirable characteristics from both techniques that are relevant for this problem.

VII. Conclusions and Future Work

The cost of provisioning power and cooling capacity for data centers is a significant fraction of their expense, often exceeding a third of the total capital and operating expense. Power capping is an effective means to reduce the capacity required and also to adapt to changes in available capacity when demand response pricing or renewable energy sources are used. We described why existing methods for power capping lack two desirable properties of speed and stability and showed where these properties can make the

existing power capping mechanisms infeasible to be applied. We also presented an approach based on admission control to ensure stable and efficient operation. While admission control cannot replace existing methods due to multiple practical issues, we showed how it can provide the desirable characteristics in a capping system when used together with existing mechanisms.

This work illustrates the usefulness of admission control in power capping, but several open challenges remain. These include the design of specific algorithms that control the extent of admission control applied, its implementation in an efficient manner with minimal modifications of deployed applications, and safe integration of multiple control mechanisms. Future work also includes prototyping rapid power capping mechanisms that can quickly reduce power in case of rapid dynamics and then use feedback to iteratively refine the power settings for maximum performance within the safe operating region. We believe that the understanding of relevant issues developed in this work will enable further research towards addressing these challenges.

REFERENCES

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

[2] J. Hamilton, "Cost of power in large-scale data centers," Blog entry dated 11/28/2008 at http://perspectives.mvdirona.com, also in Keynote, at ACM SIGMETRICS 2009.

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

[4] C. Lefurgy, X. Wang, and M. Ware, "Server-level power control," in *Proceedings of the Fourth International Conference on Autonomic Computing*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 4–. [Online]. Available: http://dl.acm.org/citation.cfm?id=1270385.1270763

[5] P. Ranganathan, P. Leech, D. Irwin, J. Chase, and H. Packard, "Ensemble-level power management for dense blade servers," in *In Proceedings of the International Symposium on Computer Architecture (ISCA*, 2006, pp. 66–77.

[6] M. E. Femal and V. W. Freeh, "Boosting data center performance through non-uniform power allocation," in *ICAC*, 2005.

[7] X. Wang and Y. Wang, "Coordinating power control and performance management for virtualized server clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 99, 2010.

[8] X. Wang, M. Chen, C. Lefurgy, and T. W. Keller, "Ship: A scalable hierarchical power control architecture for large-scale data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 1, pp. 168–176, 2012.

[9] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No "power" struggles: coordinated multi-level power management for the data center," in *ASPLOS*, 2008.

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

[11] D. Bhandarkar, "Watt matters in energy efficiency," Server Design Summit, 2010.

[12] "Duke utility bill tariff," http://www.duke-energy.com/pdfs/scscheduleopt.pdf.

[13] S. Govindan, A. Sivasubramaniam, and B. Urgaonkar, "Benefits and limitations of tapping into stored energy for datacenters," in *International Symposium of Computer Architecture (ISCA)*, 2011.

[14] X. Fan, W. dietrich Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *In Proceedings of ISCA*, 2007.

[15] J. Hamilton, "Energy proportional datacenter networks," http://perspectives.mvdirona.com/2010/08/01/EnergyProportionalDatacenterNetworks.aspx, 2010.

[16] A. Verma, P. De, V. Mann, T. Nayak, A. Purohit, G. Dasgupta, and R. Kothari, "Brownmap: Enforcing power budget in shared data centers," in *Middleware(ODP)*, December 2010.

[17] HP, "Power regulator for proliant servers," http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00300430/c00300430.pdf.

[18] X. Fu, X. Wang, and C. Lefurgy, "How much power oversubscription is safe and allowed in data centers?" in *The 8th International Conference on Autonomic Computing*, June 2011.

[19] D. Meisner and T. F. Wenisch, "Peak power modeling for data center servers with switched-mode power supplies," in *Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design*, ser. ISLPED '10, 2010.

[20] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy, "Optimal power allocation in server farms," in *SIGMETRICS*, 2009.

[21] J. Gettys and K. Nichols, "Bufferbloat: dark buffers in the internet," *Commun. ACM*, vol. 55, no. 1, pp. 57–65, Jan. 2012. [Online]. Available: http://doi.acm.org/10.1145/2063176.2063196

[22] K. Rajamani, H. Hanson, J. Rubio, S. Ghiasi, and F. L. R. III, "Application-aware power management," in *IISWC*, 2006.

[23] R. Nathuji, P. England, P. Sharma, and A. Singh, "Feedback driven qos-aware power budgeting for virtualized servers," in *FeBID*, 2009.

[24] L. Cherkasova and P. Phaal, "Session-based admission control: a mechanism for peak load management of commercial web sites," *Computers, IEEE Transactions on*, vol. 51, no. 6, pp. 669 –685, jun 2002.

[25] J. Carlstrom and R. Rom, "Application-aware admission control and scheduling in web servers," in *IEEE INFOCOM*, 2002, pp. 506 – 515.

[26] M. Welsh and D. Culler, "Adaptive overload control for busy internet servers," in *Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems - Volume 4*, ser. USITS'03, 2003.

[27] X. Liu, J. Heo, L. Sha, and X. Zhu, "Adaptive control of multi-tiered web applications using queueing predictor," in *NOMS*, 2006, pp. 106–114.

[28] M. Kihl, A. Robertsson, and B. Wittenmark, "Analysis of admission control mechanisms using non-linear control theory," in *ISCC*, 2003, pp. 1306–1311.

[29] V. Sharma, A. Thomas, T. F. Abdelzaher, K. Skadron, and Z. Lu, "Power-aware qos management in web servers," in *RTSS*, 2003.

[30] C. Poussot-Vassal, M. Tanelli, and M. Lovera, "A control-theoretic approach for the combined management of quality-of-service and energy in service centers," in *Run-time Models for Self-managing Systems and Applications*, ser. Autonomic Systems. Springer Basel, 2010, pp. 73–96.

[31] T. Voigt and P. Gunningberg, "Adaptive resource-based web server admission control," in *ISCC*, 2002, pp. 219–224.