Characterizing the Impact of the Workload on the Value of Dynamic Resizing in Data Centers

Kai Wang*, Minghong Lin†, Florin Ciucu‡, Adam Wierman† and Chuang Lin§
*Institute of Software, Chinese Academy of Sciences,
†California Institute of Technology, ‡T-Labs / TU Berlin, §Tsinghua University

***Abstract*—Energy consumption imposes a significant cost for data centers; yet much of that energy is used to maintain excess service capacity during periods of predictably low load. Resultantly, there has recently been interest in developing designs that allow the service capacity to be dynamically resized to match the current workload. However, there is still much debate about the value of such approaches in real settings. In this paper, we show that the value of dynamic resizing is highly dependent on statistics of the workload process. In particular, both slow time-scale non-stationarities of the workload (e.g., the peak-to-mean ratio) and the fast time-scale stochasticity (e.g., the burstiness of arrivals) play key roles. To illustrate the impact of these factors, we combine optimization-based modeling of the slow time-scale with stochastic modeling of the fast time scale.**

I. Introduction

Energy costs represent a significant, and growing, fraction of a data center's budget. Hence there is a push to improve the energy efficiency of data centers, both in terms of the components (servers, disks, network [17], [5], [7], [11]) and the algorithms [3], [9], [8], [16]. One specific aspect of data center design that is the focus of this paper is dynamically resizing the service capacity of the data center so that during periods of low load some servers are allowed to enter a power-saving mode (e.g., go to sleep or shut down).

The potential benefits of dynamic resizing have been a point of debate in the community [14], [9], [18]. On one hand, it is clear that, because data centers are far from perfectly energy proportional, significant energy is used to maintain excess capacity during periods of predictably low load when there is a diurnal workload with a high peak-to-mean ratio. On the other hand, there are also significant costs to dynamically adjusting the number of active servers. These costs come in terms of the engineering challenges in making this possible [10], [19], [4], as well as the latency, energy, and wear-and-tear costs of the actual "switching" operations involved [6], [9], [13].

The challenges for dynamic resizing highlighted above have been the subject of significant research. At this point, many of the engineering challenges associated with facilitating dynamic resizing have been resolved, e.g., [10], [19], [4]. Additionally, the algorithmic challenge of deciding, without knowledge of the future workload, whether to incur the significant "switching costs" associated with changing the available service capacity has been studied in depth and a number of promising algorithms have emerged [16], [2], [9], [12].

However, despite this body of work, the question of characterizing the potential benefits of dynamic resizing has still not been properly addressed. Providing new insight into this topic is the goal of the current paper.

The perspective of this paper is that, apart from engineering challenges, the key determinant of whether dynamic resizing is valuable is the workload. In particular, a key observation, which is the starting point for our work, is that there are two factors of the workload which provide dynamic resizing potential savings:

(i) Non-stationarities at a slow time-scale, e.g., diurnal workload variations.
(ii) Stochastic variability at a fast time-scale, e.g., the burstiness of request arrivals.

To this point, we are not aware of any work characterizing the benefits of dynamic resizing that captures both of these features. There is one body of literature which provides algorithms that take advantage of (i), e.g., [9], [8], [16], [2]. This work tends to use an optimization-based approach to develop dynamic resizing algorithms. There is another body of literature which provides algorithms that take advantage of (ii), e.g., [12], [13]. This work tends to assume a stationary queueing model with Poisson arrivals to develop dynamic resizing algorithms.

The first contribution of this paper is to provide an analytic framework that captures both effects (i) and (ii). We accomplish this by using an optimization framework at the slow time-scale (see Section II), which is similar to that of [16], and combining this with stochastic network calculus and large deviations modeling for the fast time-scale (see Section III), which allows us to study a wide variety of underlying arrival processes. We consider both light-tailed models with various degrees of burstiness and heavy-tailed models that exhibit self-similarity.

Using this modeling framework, we are able to provide both analytic and numerical results that yield new insight into the potential benefits of dynamic resizing (see Section IV). Specifically, we use trace-driven numerical simulations to study (i) the role of burstiness for dynamic resizing, (ii) the role of the peak-to-mean ratio for dynamic resizing, (iii) the role of the SLA for dynamic resizing, and (iv) the interaction between (i), (ii), and (iii). The key realization is that each of these parameters are extremely important for determining the value of dynamic resizing. In particular, for any fixed choices of two of these parameters, the third can be chosen so that dynamic resizing does or does not provide significant cost savings for the data center. Thus, performing a detailed study of the interaction of these factors is important.

In addition to detailed case studies, we provide analytic results that support many of the insights provided by the numerics. The theorems provide monotonicity and scaling results for dynamic resizing in the case of Poisson arrivals and heavy-tailed, self-similar arrivals. Due to page limitation, the theorems are omitted in the paper, and the full verion is available as the technical report [20].

II. Slow Time-scale Model

In this section and the one that follows, we introduce our model. We start with the "slow time-scale model". This model is meant to capture what is happening at the time-scale of the data center control decisions, i.e., at the time-scale which the data center is willing to adjust its service capacity.

A. The Workload

At this time-scale, our goal is to provide a model which can capture the impact of diurnal non-stationarities in the workload. To this end, we consider a discrete-time model such that there is a time interval of interest which is evenly divided into "frames" $k \in \{1, ..., K\}$. In practice, the length of a frame could be on the order of 5-10 minutes, whereas the time interval of interest could be as long as a month/year. The mean arrival rate to the data center in frame k is denoted by λ_k, and non-stationarities are captured by allowing different rates during different frames.

B. The Data Center Cost Model

The model for data center costs focuses on the server costs of the data center, as minimizing server energy consumption also reduces cooling and power distribution costs. We model the cost of a server by the operating costs incurred by an active server, as well as the switching cost incurred to toggle a server into and out of a power-saving model (e.g., off/on or sleeping/waking). Both components can be assumed to include energy cost, delay cost, and wear-and-tear cost. See [16] and [15] for further discussion of the model.

The operating costs are modeled by a convex function $f(\lambda_{i,k})$, which is the same for all the servers, where $\lambda_{i,k}$ denotes the average arrival rate to server i during frame k. The convexity assumption is quite general and captures many common server models. This cost is often modeled using an affine function as follows

$$f(\lambda_{i,k}) = e_0 + e_1\lambda_{i,k} \ , \tag{1}$$

where e_0 and e_1 are constants [1], [5].

The switching cost, denoted by β, models the cost of toggling a server back-and-forth between active and power-saving models. The switching cost includes the costs of the energy used toggling a server, the delay in migrating connections/data when toggling a server, and the increased wear-and-tear cost.

C. The Data Center Optimization

Given the cost model above, the data center has two control decisions at each time: determining n_k, the number of active servers in every time frame, and assigning arriving jobs to servers, i.e., determining $\lambda_{i,k}$ such that $\Sigma_{i=1}^{n_k}\lambda_{i,k} = \lambda_k$. All servers are assumed to be homogeneous with constant rate capacity $\mu > 0$. Modeling heterogeneous servers is also possible, as in [15]; however we limit the discussion in this paper to the homogeneous setting for clarity.

The goal of the data center is to determine n_k and $\lambda_{i,k}$ to minimize the cost incurred during $[0, K]$:

$$\min \sum_{k=1}^{K}\sum_{i=1}^{n_k} f(\lambda_{i,k}) + \beta\sum_{k=1}^{K}(n_k - n_{k-1})^+ \tag{2}$$

$$s.t. \begin{cases} 0 \le \lambda_{i,k} \le \lambda_k \\ \Sigma_{i=1}^{n_k}\lambda_{i,k} = \lambda_k \\ \mathbb{P}(D_k > \bar{D}) \le \bar{\varepsilon} \ , \end{cases} \tag{3}$$

where the final constraint is introduced to capture the SLA of the data center. We use D_k to represent the steady-state delay during frame k, and $(\bar{D}, \bar{\epsilon})$ to represent an SLA of the form "the probability of a delay larger than $\bar{D}$ must be bounded by probability $\bar{\varepsilon}$".

This model generalizes the data center optimization problem from [16] by accounting for the additional SLA constraint. The specific values in this constraint are determined by the stochastic variability at the fast time-scale. In particular, we derive (for a variety of workload models) a sufficient constraint $n_k \ge \frac{C_k(\bar{D},\bar{\varepsilon})}{\mu}$ such that

$$n_k \ge \frac{C_k(\bar{D}, \bar{\varepsilon})}{\mu} \Longrightarrow \mathbb{P}(D_k > \bar{D}) \le \bar{\varepsilon} \ . \tag{4}$$

Here, μ is the constant rate capacity of each server and $C_k(\bar{D}, \bar{\varepsilon})$ is to be determined for each considered arrival model. One should interpret $C_k(\bar{D}, \bar{\varepsilon})$ as the overall effective capacity/bandwidth needed in the data center such that the SLA delay constraint is satisfied within frame k.

Note that the new constraint is only sufficient for the original SLA constraint. The reason is that $C_k(\bar{D}, \bar{\varepsilon})$ is computed, in the next section, from upper bounds on the distribution of the transient delay within a frame.

With the new constraint, however, the optimization problem in (2) can be considerably simplified. Indeed, note that n_k is fixed during each time frame k and the remaining optimization for $\lambda_{i,k}$ is convex. Thus, we can simplify the form of the optimization problem, and Eqs. (2)-(3) become:

$$\underline{\text{Data Center Optimization Problem}}$$

$$\min \sum_{k=1}^{K} n_k f(\lambda_k/n_k) + \beta\sum_{k=1}^{K}(n_k - n_{k-1})^+ \tag{5}$$

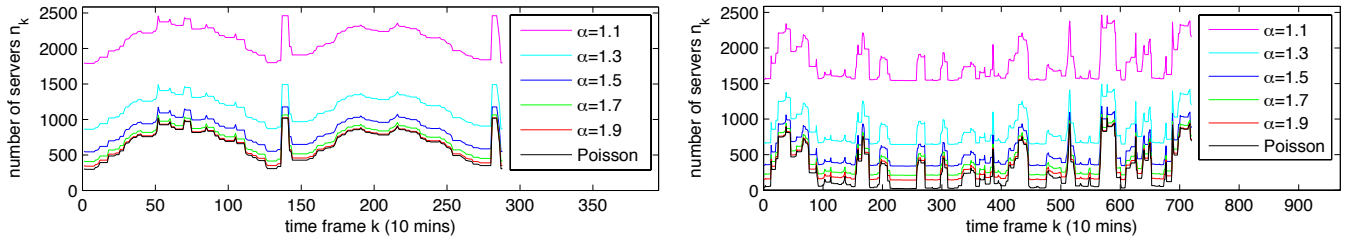
$$\text{s.t. } n_k \ge \frac{C_k(\bar{D}, \bar{\varepsilon})}{\mu} \ .$$

The key difference between the optimization above, and that of [16], is the SLA constraint, which provides a bridge between the slow time-scale and fast time-scale models. Specifically, the fast time-scale model uses large deviations and stochastic network calculus techniques to calculate $C_k(\bar{D}, \bar{\varepsilon})$. Deriving algorithm for this problem is not the goal of the current paper. We use the Lazy Capacity Provisioning (LCP) [16], and the algorithm for our setting is described in [20].

III. Fast Time-scale Model

Given the model of the slow time-scale in the previous section, we now zoom in to give a description for the fast time-scale model. By "fast" time-scale, we mean the time-scale at which requests arrive, as opposed to the "slow" time-scale at which dynamic resizing decisions are made by the data center. To model the fast time-scale, we evenly break each frame from the slow time-scale into "slots" $t \in \{1, \ldots, U\}$, such that frame_length $= U \cdot$ slot_length.

We consider a variety of models for the workload process at this fast time-scale, including both light-tailed models with



(a) Hotmail (b) MSR

Fig. 1. Impact of burstiness on provisioning n_k for heavy-tailed arrivals.

various degrees of burstiness, as well as heavy-tailed models that exhibit self-similarity. In all cases, our assumption is that the workload is stationary over the slots that make up each time frame.

The goal of this section is to derive the value of $C_k(\bar{D}, \bar{\varepsilon})$ in the constraint $n_k \geq \frac{C_k(\bar{D},\bar{\varepsilon})}{\mu}$ from Eq. (4), and thus enable an interface between the fast and slow time-scales by parameterizing the Data Center Optimization Problem from Eq. (5) for a broad range of workloads. Note that throughout this section we suppress frame's subscript k for n_k, λ_k, C_k, and D_k, and focus on a generic frame.

A. An Aggregation Property

We denote the cumulative arrival (workload) process at the data center's dispatcher by $A(t)$. For each slot $t = 1, \ldots, U$, $A(t)$ counts the total number of jobs arrived in the time interval $[0, t]$. Depending on the total number n of active servers, the arrival process is dispatched into the sub-arrival processes $A_i(t)$ with $i = 1, \ldots, n$. The cumulative response processes from the servers are denoted by $R_i(t)$, whereas the total cumulative response process from the data center is denoted by $R(t) = \sum_i R_i(t)$. All arrival and response processes are assumed to be non-negative, non-decreasing, and left-continuous, and satisfy the initial condition $A(0) = R(0) = 0$. For convenience we use the bivariate extensions $A(s,t) := A(t) - A(s)$ and $R(s,t) := R(t) - R(s)$.

The service provided by a server is modeled in terms of probabilistic lower bounds using the concept of a stochastic service process. This is a bivariate random process $S(s,t)$ which is non-negative, non-decreasing, and left-continuous. Formally, a server is said to guarantee a (stochastic) service process $S(s,t)$ if for *any* arrival process $A(t)$ the corresponding response process $R(t)$ from the server satisfies for all $t \geq 0$

$$R(t) \geq A * S(t) \ , \tag{6}$$

where '$*$' denotes the min-plus convolution operator, i.e., for two (random) processes $A(t)$ and $S(s,t)$,

$$A * S(t) := \inf_{0 \leq s \leq t} \{A(s) + S(s,t)\} \ . \tag{7}$$

We are now ready to state the aggregation property. The proof is deferred to the technical report [20].

Lemma 1. *Consider an arrival process $A(t)$ which is dispatched to n servers. Each server i is work-conserving with constant rate capacity $\mu > 0$. Arrivals are dispatched deterministically across the servers such that each server i receives a fraction $\frac{1}{n}$ of the arrivals. Then, the system has service process $S(s,t) = n\mu(t-s)$, i.e., $R(t) \geq A * S(t)$.*

B. Arrival Processes

The next step in deriving the SLA constraint $n \geq \frac{C(\bar{D},\bar{\varepsilon})}{\mu}$ is to derive a bound on the distribution of the delay at the virtual server with arrival process $A(t)$ and service process $S(s,t) = C(\bar{D}, \bar{\varepsilon})(t-s)$, i.e., $\mathbb{P}(D(t) > \bar{D}) \leq \varepsilon$.

It is importation to observe that the violation probability ε holds for the *transient* delay process $D(t)$, which is defined as $D(t) := \inf\{d : A(t-d) \leq R(t)\}$, and which models the delay spent in the system by the job leaving the system, if any, at time t. However, the violation probability ε is derived so that it is time invariant, which implies that it bounds the distribution of the stead-state delay $D = \lim_{t\to\infty} D(t)$ as well. Therefore, the value of $C(\bar{D}, \bar{\varepsilon})$ can be finally computed by solving the equation $\varepsilon = \bar{\varepsilon}$.

We follow the outline above to compute $C(\bar{D}, \bar{\varepsilon})$ for light- and heavy-tailed arrival processes; interested readers may refer to the technical report [20] for details.

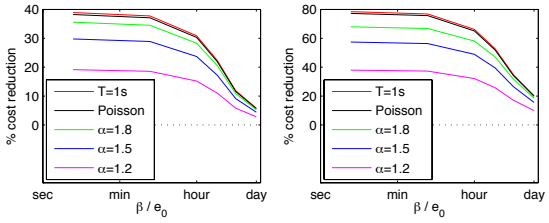
IV. Case Studies

Given the model described in the previous two sections, we are now ready to explore the potential of dynamic resizing in data centers, and how this potential depends on the interaction between non-stationarities at the slow time-scale and burstiness/self-similarity at the fast time-scale.

A. Setup

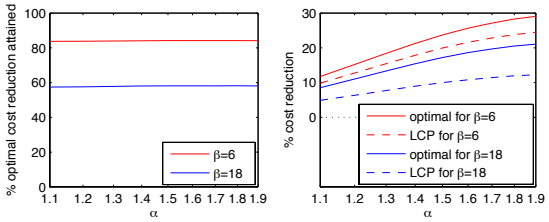
The time frame for adapting the number of servers n_k is assumed to be 10 min, and each time slot is assumed to be 1 s, i.e., $U = 600$. When not otherwise specified, we assume the following parameters for the data center SLA agreement: the delay upper bound $\bar{D} = 200$ms, and the delay violation probability $\bar{\varepsilon} = 10^{-3}$. We choose units such that the fixed energy cost is $e_0 = 1$. The load-dependent energy consumption is set to $e_1 = 0$. Unless otherwise specified, we use the normalized switching cost $\beta = 6$, and fix the burst parameters $T = 1$ and $\alpha = 1.5$. Here, T (the average time for a 2-state Markov-modulated (MM) process to change states twice) and alpha (the tail index of a Pareto r.v.) can be tuned to achieve various degrees of burstiness for the light- and heavy-tailed arrival processes. In addition, we consider a Poisson process with normalized mean; see Figure 1 from [20] for synthetic sample paths of the three arrival processes.

The workloads for these experiments are drawn from two real-world data center traces, i.e., 48-hour traces from Hotmail servers, and 1 week traces from MSR Cambridge. Loads were averaged over disjoint 10 minutes frames. We contrast three designs: (i) the optimal dynamic resizing, (ii) dynamic resizing via LCP, and (iii) the optimal 'static' provisioning. The readers may refer to the technical report [20] for details.



(a) Hotmail (b) MSR

Fig. 2. Impact of burstiness on the cost savings of dynamic resizing for different switching costs, β.



(a) cost savings (b) relative savings of LCP

Fig. 3. Impact of burstiness on the performance of LCP in the Hotmail trace.

B. Results

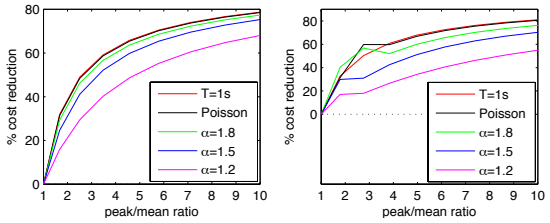
Our experiments are organized to illustrate the impact of a wide variety of parameters on the cost savings attainable via dynamic resizing.

The Role of Burstiness: A priori, one may expect that burstiness can be beneficial for dynamic resizing, since it indicates that there are periods of low load during which energy may be saved. However, this is not actually true since resizing decisions must be made at the slow time-scale while burstiness is a characteristic of the fast time-scale. Thus, burstiness is actually detrimental for dynamic resizing, since it means that the provisioning decisions made on the slow time-scale must be made with the bursts in mind, which results in a larger number of servers needed to be provisioned for the same average workload. This effect can be seen in Figure 1.

The larger provisioning created by increased burstiness manifests itself in the cost savings attainable through dynamic capacity provisioning as well. This is illustrated in Figure 2, which shows the cost savings of the optimal dynamic provisioning as compared to the optimal static provisioning for varying α and T as a function of the switching cost β.

Interestingly, though Figure 2 shows that the potential of dynamic resizing is limited by increased burstiness, it turns out that the relative performance of LCP is not hurt by burstiness. This is illustrated in Figure 3, which shows the percent of the optimal cost savings that LCP achieves. Importantly, it is nearly perfectly flat as the burstiness is varied.

The Role of the Peak-to-Mean Ratio: The impact of the peak-to-mean ratio on the potential benefits of dynamic resizing is quite intuitive: if the peak-to-mean ratio is high, then there is more opportunity to benefit from dynamically changing capacity. Figure 4 illustrates this well-known effect. The workload for the figure is generated from the traces by scaling λ_k as $\hat{\lambda}_k = c(\lambda_k)^\gamma$, varying γ and adjusting c to keep the mean constant. Figure 4 also highlights that there is a strong interaction between burstiness and the peak-to-mean ratio, where if there is significant burstiness the benefits that come from a high peak-to-mean ratio may be diminished considerably.



(a) Hotmail (b) MSR

Fig. 4. Impact of peak-to-mean ratio on the cost savings of the optimal dynamic resizing.

The Role of the SLA: Figures 5 and 6 highlight the role the violation probability $\bar{\varepsilon}$ has on the provisioning of n_k under the optimal dynamic resizing in the cases of heavy-tailed and MM arrivals. Interestingly, we see that there is a significant difference in the impact of $\bar{\varepsilon}$ depending on the arrival process. As $\bar{\varepsilon}$ gets smaller in the heavy-tailed case the provisioning gets significantly flatter, until there is almost no change in n_k over time. In contrast, no such behavior occurs in the MM case and, in fact, the impact of $\bar{\varepsilon}$ is quite small. This difference is a fundamental effect of the "heaviness" of the tail of the arrivals, i.e., a heavy tail requires significantly more capacity in order to counter a drop in $\bar{\varepsilon}$.

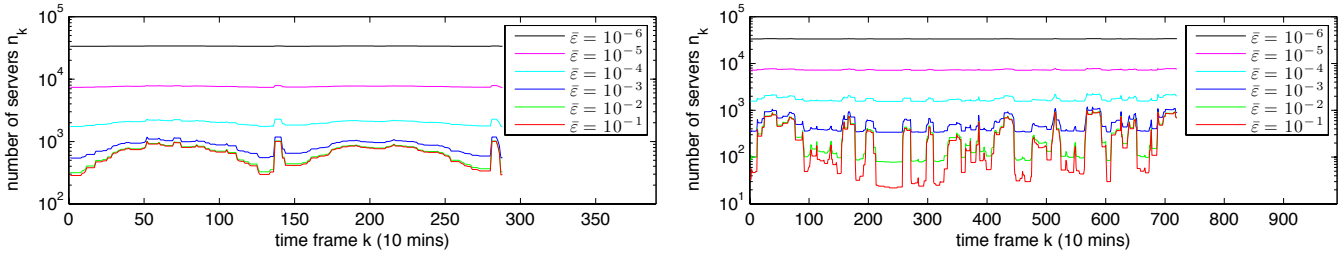
This contrast between heavy- and light-tailed arrivals is also evident in Figure 7, which highlights the cost savings from dynamic resizing in each case as a function of $\bar{\varepsilon}$. Interestingly, the cost savings under light-tailed arrivals is largely independent of $\bar{\varepsilon}$, while under heavy-tailed arrivals the cost savings is monotonically increasing with $\bar{\varepsilon}$.

When is Dynamic Resizing Valuable?: Our goal is to provide a concrete understanding of for which (peak-to-mean, burstiness, SLA) settings the potential savings from dynamic resizing is large enough to warrant implementation. Figure 8 focuses on this question. Our hope is that the figure highlights that a precursor to any debate about the value of dynamic resizing must be a joint understanding of the expected workload characteristics and the desired SLA, since for any fixed choices of two of these parameters (peak-to-mean, burstiness, SLA), the third can be chosen so that dynamic resizing does or does not provide significant cost savings for the data center. Of course, many of the settings of the data center will effect the conclusions illustrated in Figure 8. Two of the most important factors are the switching cost, β, and the SLA, particularly $\bar{\varepsilon}$, which are presented in [20].

V. Conclusion

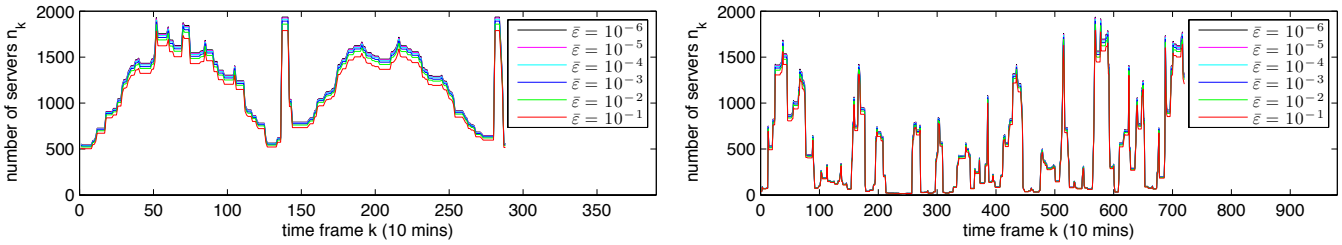
Our goal in this paper is to provide new insight into the debate about the potential of dynamic resizing in data centers. Clearly, there are many facets of this issue relating to the engineering, algorithmic, and reliability challenges involved in dynamic resizing which we have ignored in this paper. These are all important issues when trying to *realize the potential* of dynamic resizing. But, the point we have made in this paper is that when *quantifying the potential* of dynamic resizing it is of primary importance to understand the joint impact of workload and SLA characteristics.

To make this point, we have presented a new model that captures the impact of SLA characteristics in addition to both slow time-scale non-stationarities and fast time-scale burstiness in the workload. This model allows us to provide the



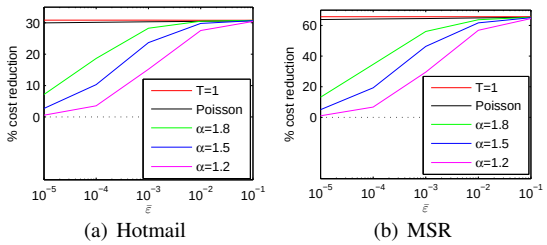
(a) Hotmail (b) MSR

Fig. 5. Impact of $\bar{\varepsilon}$ on provisioning n_k for heavy tailed arrivals.



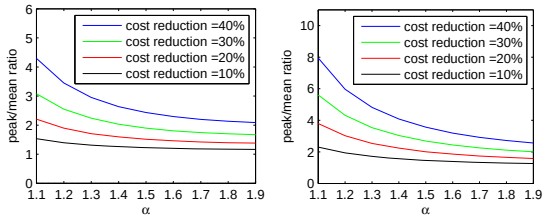
(a) Hotmail (b) MSR

Fig. 6. Impact of $\bar{\varepsilon}$ on provisioning n_k for MM arrivals.



(a) Hotmail (b) MSR

Fig. 7. Impact of $\bar{\varepsilon}$ on the cost savings of dynamic resizing.



(a) Hotmail (b) MSR

Fig. 8. Characterization of burstiness and peak-to-mean ratio necessary for dynamic resizing to achieve different levels of cost reduction.

first study of dynamic resizing that captures both the stochastic burstiness and diurnal non-stationarities of real workloads.

ACKNOWLEDGMENT

This research is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDA06010600), the 973 Program of China (No. 2010CB328105), the NSF grant of China (No. 61020106002), and NSF grant CNS 0846025 and DoE grant DE-EE0002890.

REFERENCES

[1] SPEC power data on SPEC website at http://www.spec.org.

[2] F. Ahmad and T. N. Vijaykumar. Joint optimization of idle and cooling power in data centers while maintaining response time. In *Proc. of the 15th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2010.

[3] S. Albers. Energy-efficient algorithms. *Communications of the ACM*, 53(5):86–96, May 2010.

[4] H. Amur, J. Cipar, V. Gupta, G. R. Ganger, M. A. Kozuch, and K. Schwan. Robust and flexible power-proportional storage. In *Proc. of the 1st ACM Symposium on Cloud Computing (SoCC)*, 2010.

[5] L. A. Barroso and U. Hölzle. The case for energy-proportional computing. *Computer*, 40(12):33–37, December 2007.

[6] P. Bodik, M. P. Armbrust, K. Canini, A. Fox, M. Jordan, and D. A. Patterson. A case for adaptive datacenters to conserve energy and improve reliability. Technical Report UCB/EECS-2008-127, University of California at Berkeley, 2008.

[7] E. V. Carrera, E. Pinheiro, and R. Bianchini. Conserving disk energy in network servers. In *Proc. of the 17th annual International Conference on Supercomputing (ICS)*, pages 86–97. ACM, 2003.

[8] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle. Managing energy and server resources in hosting centers. In *Proc. of the 8th ACM symposium on Operating systems principles (SOSP)*, pages 103–116, 2001.

[9] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In *Proc. of the 5th USENIX Symposium on Networked Systems Design & Implementation (NSDI)*, 2008.

[10] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In *Proc. of the 2nd USENIX Symposium on Networked Systems Design & Implementation (NSDI)*, pages 273–286, 2005.

[11] B. Diniz, D. Guedes, W. Meira, Jr., and R. Bianchini. Limiting the power consumption of main memory. In *Proc. of the 34th annual International Symposium on Computer Architecture (ISCA)*, pages 290–301, 2007.

[12] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch. Optimality analysis of energy-performance trade-off for server farm management. *Performance Evaluation*, 67(11):1155–1171, November 2010.

[13] A. Gandhi, M. Harchol-Balter, and M. A. Kozuch. The case for sleep states in servers. In *Proc. of the 4th Workshop on Power-Aware Computing and Systems (HotPower)*, pages 2:1–2:5, 2011.

[14] 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.*, 39:68–73, December 2008.

[15] M. Lin, Z. Liu, A. Wierman, and L. L. Andrew. Online algorithms for geographical load balancing. In *Proc. of the 3rd International Green Computing Conference (IGCC)*, 2012.

[16] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In *Proc. of the 30th IEEE International Conference on Computer Communications (Infocom)*, 2011.

[17] D. Meisner, B. T. Gol, and T. F. Wenisch. PowerNap: Eliminating server idle power. *ACM SIGPLAN Notices*, 44(3):205–216, Mar. 2009.

[18] D. Meisner, C. M. Sadler, L. A. Barroso, W.-D. Weber, and T. F. Wenisch. Power management of online data-intensive services. In *Proc. of the 38th annual International Symposium on Computer Architecture (ISCA)*, 2011.

[19] E. Thereska, A. Donnelly, and D. Narayanan. Sierra: a power-proportional, distributed storage system. Technical Report MSR-TR-2009-153, Microsoft Research, 2009.

[20] K. Wang, M. Lin, F. Ciucu, A. Wierman, and C. Lin. Characterizing the impact of the workload on the value of dynamic resizing in data centers. Technical Report ISCAS-2012-GD1, available at http://arxiv.org/abs/1207.6295, 2012.