Real-Time Power Control of Data Centers for Providing Regulation Service

Hao Chen[1], Ayse K. Coskun[1], and Michael C. Caramanis [2]

[1]Department of Electrical and Computer Engineering, Boston University
[2]Division of Systems Engineering, Boston University
{*haoc, acoskun, mcaraman*}*@bu.edu*

***Abstract*— Data centers account for 3% of the national electricity consumption today. Recent advances in power capping enable data centers to provide Regulation Service (RS) reserves to power markets. RS reserve requirements are expected to increase with the integration of substantial environmentally sustainable, albeit volatile and intermittent, renewable generation. We investigate the ability of server clusters to provide RS reserves while meeting contractual Quality-of-Service (QoS) guarantees for user applications running on the cluster. We propose a dynamic server power regulation policy for processing randomly arriving applications within probabilistic QoS constraints while tracking dynamically broadcasted RS requests by the power market Independent System Operator (ISO). Numerical results demonstrate the ability to select optimal RS reserve and average energy consumption levels that allow data centers to both deliver probabilistic QoS guarantees to customers and track real time ISO RS requests. Finally, we argue that simulated optimal policy performance statistics can enable data centers to bid for energy and to offer RS reserves in the the hour-ahead power markets.**

I. INTRODUCTION

The unabating increase in data center energy consumption is a growing concern because of the associated environmental impacts, the high monetary costs, and the strains imposed on the National Power System Infrastructure. Electricity used by data centers worldwide increased by 56% from 2005 to 2010 [9]. In fact, electricity used in the US data centers currently accounts for around 3% of the total electricity consumption [9]. At the same time, energy and environmental sustainability objectives are likely to result in the integration of ever increasing renewable energy generation into the electricity grid. The volatility and intermittency of renewable generation, however, combined with the lack of reliable large-scale energy storage solutions, creates challenges for grid operators who need to match supply and demand by securing commensurate flexible capacity reserves in forward markets and dispatching them in real time. Data centers offer a unique opportunity as they can provide the necessary flexibility in their energy consumption. Tapping this flexibility can lead to satisfying most of the growth in data center energy consumption from renewables, and also provide additional reserves to accommodate less flexible uses of electricity in our society.

This research has been partially supported by NSF grant 1038230, Sandia National Laboratories, Oracle, and Decision Detective Corporation (SBIR).

The main contribution of this paper is that it provides a credible argument on the ability of data centers to participate profitably in energy markets and provide much needed RS reserves. To do so, data centers must be able to consume power in a manner that closely tracks a dynamic power regulation signal provided by an ISO. Server or data center power capping techniques, where the real time power consumption of computational nodes are controlled not to exceed a given cap, have been proposed recently (e.g., [2], [25], [27]). However, prior work has not investigated the dynamic power tracking by data centers required to participate in the advanced energy markets. Solving the runtime power tracking problem requires specification and optimization of an implementable dynamic policy that guarantees probabilistic user QoS constraints with reasonably close tracking of ISO dictated real time power consumption targets. We demonstrate the optimization of a proposed runtime implementation policy implemented on a commercial multicore server. For typical data center utilization levels, our experiments show that we can achieve up to 30% reduction in the energy costs while satisfying various probabilistic performance requirements for high and low priority jobs running on the server.

The rest of this paper is organized as follows. Section II provides an overview of power markets and reviews the related work on the RS provision as well as data center power management and capping techniques. Section III formulates the overall data center RS provision problem and how it applies to a single server. Section IV describes the simulation methodology and the system model built based on a real-life server. Section V presents and discusses the numerical results elaborating our claim that data centers can richly benefit from the provision of RS through power market participation, and Section VI concludes the paper.

II. BACKGROUND AND RELATED WORK

In most markets for goods and services, uncertainty results in a temporary mismatch of supply and demand that usually does not have overly significant consequences. In electricity markets, however, system stability requires unfailing balancing of supply and demand in real time in order to avoid catastrophic events such as blackouts. As a result, the lack of reliable and economical large-scale storage solutions, despite some progress in battery technology and fly-wheel storage

devices, has elevated reserves with the requisite dynamical characteristics as a key component of modern power markets.

Indeed, the need to match supply with demand in real time has resulted in the adoption of a series of power markets that operate at several different time-scales and clear energy and reserve transactions simultaneously. Focusing on the short-term markets ([10], [15], [16]) that are most relevant for the proposed work we identify: (i) day-ahead markets that close at noon of the previous day and clear energy and reserve bids by market participants for each of the 24 hours of the next day, (ii) hour-ahead adjustment markets that close an hour in advance of each hour allowing participants to adjust their day ahead positions on both energy and reserves at clearing prices that reflect new information, and (iii) 5-minute close-to-real-time economic dispatch markets that determine actual ex post variable marginal cost of energy employed to adjust participant revenues and costs for deviating from the quantities cleared in the previous two markets.

Power markets provide a socially efficient mechanism for pricing and allocating energy, and also for securing the reserves needed for uncertainty contingency planning. Reserves secured in the forward markets include primary (also known as frequency control), secondary (known also as regulation service, or RS) and tertiary reserves which are deployed by commands issued respectively in millisecond, 5 second and 5 minute intervals [21]. We focus here on RS reserves since they usually command high prices [22] and we show that data centers are well qualified for their provision. RS reserves are presently offered primarily by centralized generators, but market rules are already changing to allow the demand side to offer reserves as well. For example, PJM, one of the largest US ISOs, has allowed electricity loads to participate in reserve transactions since 2006 [20] with other ISOs contemplating to follow suit.

A power market participant in the PJM balancing area who is cleared during hour h to consume on average at $\bar{P}$ kW and provide RS reserves of R kW with market clearing prices for energy and RS reserves Π^E and Π^R respectively, is charged for its average power consumption and credited for the RS reserves as $\Pi^E\bar{P} - \Pi^R R$ [22]. However, the credit for the RS reserves does not come for free. As the hour unfolds, the provider has the obligation to modulate dynamically its power consumption, $P(t)$, so as to track the ISO Regulation Signal, $z(t)$, by ensuring that $P(t) \approx \bar{P} + z(t)R$. Part of the hour ahead RS income $\Pi^R R$ is reduced in proportion to the tracking error [22]. Moreover, if the tracking error exceeds a probabilistic tolerance constraint, the participant may lose its license to participate in the RS reserves market [21]. The ISO determines $z(t)$ through an integral proportional filter of the *Area Control Error* (ACE)[1], and frequency excursions outside of the tolerance interval $[59.980, 60.020Hz]$. The RS signal, $z(t)$, is a zero time average scalar taking values in the interval $[-1, 1]$ and is updated every 4 seconds in increments that do not exceed $\pm R/(\tau/4)$ where τ is 150 seconds for the fast (F) RS and 300 seconds for the slower (S) RS [19]. Typical hour long trajectories of fast (F) and slow (S) RS signals[2], $z^F(t)$ and $z^S(t)$ respectively, are shown in Figure 1, which are indicative of the statistical behavior of $z(t)$.

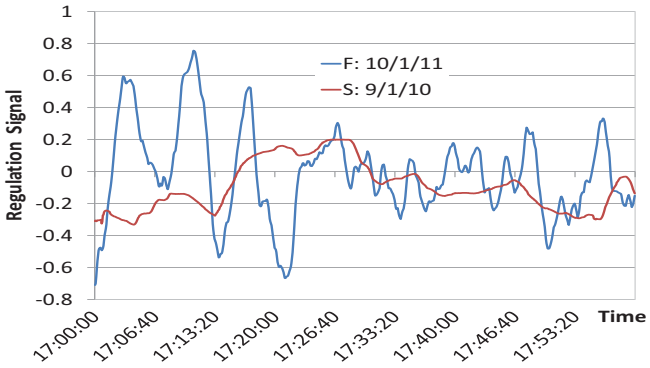

Fig. 1. Typical PJM 150sec ramp rate (F) and 300sec ramp rate (S) regulation signal trajectories.

Recent research has investigated demand side provision of RS. Paschalidis et al. [17] propose a market-based mechanism that enables a building smart microgrid operator to offer RS reserves and meet the associated obligation issued by the ISO, using a dynamic pricing policy. Caramanis et al. [1] study optimal dynamic pricing policies for a single load type and their implications to hour ahead bidding decisions.

As part of the related work on data centers, Rao et al. [24] consider electricity cost minimization subject to QoS guarantees by exploiting spaciotemporal variations in electricity market clearing prices. Mohsenian-Rad et al. [14] propose a grid-aware service request routing design subject to power flow constraints applicable to a smart grid with load balancing and reliability objectives. Ghamkhari et al. [6] investigate the potential profits from data centers' participation in an ancillary service market.

Server power management has been widely studied in recent years. As a result, most current processors have built-in control knobs such as Dynamic Voltage and Frequency Scaling (DVFS) and turning off idle units. Multi-core processors introduce new opportunities for power management as they enable larger degrees of freedom in job scheduling and allocation. Combining DVFS with thread allocation improves the granularity and dynamic range of server power control, and enables better tuning of the performance-power tradeoffs ([23], [3]). It is also possible to enhance the global power management policy on a multi-core processor by including objectives such as prioritization, power balancing, or optimized throughput [7]. Taking processor hardware heterogeneity into account during job scheduling and DVFS expands the applicability and efficiency of power management ([26], [23]). There are other approaches that require hardware support for meeting power budgets through micro-architectural reconfiguration ([13], [8]).

Power capping, where compute nodes operate below a given peak power value, is commonly used today for meeting

[1]ACE measures the difference between actual and scheduled net imports from adjacent balancing areas.

[2]The data is from: http://www.pjm.com/markets-and-operations/ancillary-services/mkt-based-regulation/fast-response-regulation-signal.aspx.

peak power constraints. A software-based power capping strategy meets a given average power budget by inserting idle cycles during execution [4]. Recent research investigates designing control techniques to coordinate multiple levels of capping in a data center [27]. These management techniques use the DVFS and throttling capabilities of the processors. In addition to throttling, power nap modes, in which the system can enter and exit from low-power modes in milliseconds, have been proposed to cope with demand variations in data centers [12]. A recent approach trains power-performance models for a target server and uses thread allocation along with DVFS to improve performance under dynamically changing power caps ([2], [25]).

To the best of our knowledge, our work is the first to combine power RS control and data center power management together. Instead of solely targeting peak power constraints, our goal is to design capping techniques for the data center to be able to first promise a certain flexible stand-by capacity and then modulate this capacity up or down in real-time so as to follow closely the RS signal broadcasted by the ISO. While power management of computers is being addressed at many levels today, none of the prior techniques leverage data center energy management as a grid load stabilizer.

III. PROBLEM FORMULATION

Data center power consumption can be broadly divided into the cooling system power (chillers, air circulation, etc.) and the computational power. Data centers can provide RS by responding to power market outcomes which vary by day, hour and 5 minute periods, and to ISO RS signals broadcasted in 4 second intervals through controlling (1) the *5 minute or longer time-scale* power consumption dynamics of the cooling system, and (2) the practically *real-time* dynamics of the computing systems (i.e., servers). In this section, we first briefly describe a data center level RS provision algorithmic framework and then concentrate on the control of *a single server*, which constitutes the framework's main building block.

A. Data Center Level Framework

As mentioned earlier, dynamic power markets are cascaded. The energy and reserve schedules are co-optimized when the day-ahead market clears, and are subsequently rescheduled at the hour-ahead adjustment market. A load side participant, in our case a data center, who commits in the hour ahead market to consume at the average rate of $\bar{P}$ kW, and to provide R kW of RS reserves, is obligated to modulate the real-time data center consumption, $P(t)$, so as to track an RS signal $z(t)$ broadcasted in 4 second intervals by the ISO. Thus, if the data center has agreed at hour h to $\bar{P}$ and R values for the next hour $[h, h+1]$, it must respond to the dynamics of the ISO RS signal, to consume at the instant power rate $P(t) \approx \bar{P} + z(t)R$.

The data center RS provision algorithmic framework consists of a master problem, which interacts with a cooling system sub-problem and server sub-problems, all of which communicate in the 5 minute time-scale as shown in Figure 2. The 5-minute interval is selected here as the cooling power cannot be controlled and regulated as fast as the computing power due to the thermal time constants. Based on an estimate of the stochastic arrival rates of processing jobs of type j during the next hour, $\lambda_h^{DC,j}$, the master problem negotiates the hourly data center average consumption rate, $\bar{P}_h^{DC}$, and maximum RS, R_h^{DC}. At the beginning of each 5 minute period within the hour, denoted by $k \in [1, 2, 3, ..., 11, 12]$:

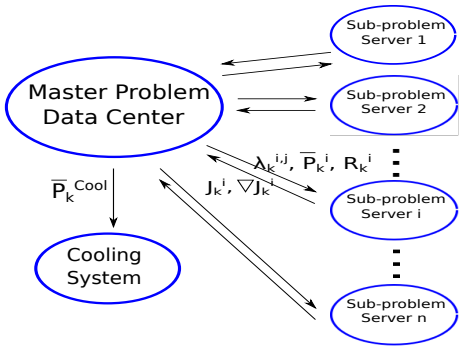

Fig. 2. Data center level power regulation framework.

1) The master problem provides targets to the sub-problems. First, it allocates a power budget to each server $i \in \{1, 2, \ldots, n\}$ and to the cooling system such that $\bar{P}_h^{DC} = \bar{P}_k^{Cool} + \sum_{i=1}^{n} \bar{P}_k^i$. Note that $\bar{P}_h$ is constant across the next hour but the power budgets of the sub-problems, $\bar{P}_k^i$, need to be adjusted every 5 minutes as workload and cooling dynamics may change. The master problem also distributes the hourly RS requirements, $R_h^{DC} = \sum_{i=1}^{n} R_k^i$ and the jobs, $\lambda_h^{DC,j} = \sum_{i=1}^{n} \lambda_k^{i,j}$, $\forall j$, where $\lambda_h^{DC,j}$ is the arrival rate of job type j to the data center. $\bar{P}_k^i$ and R_k^i are related to the type and the rate of the workloads when the servers are identical (in terms of their architecture, configuration, location, etc.). Budgeting algorithms proposed recently (e.g., [28], [5]) can be applied to solve this allocation problem. In this work, we focus solely on regulating the computational (server) power.

2) Each server sub-problem $i = \{1, 2, \ldots, n\}$ then individually performs the RS signal tracking by consuming at the rate $P(t)^i \approx \bar{P}_k^i + z(t)R_k^i$ and evaluates its expected performance during the next five minute period. The server then returns to the master problem its performance, $J_k^i(\lambda_k^{i,j}\ \forall j,\ \bar{P}_k^i,\ R_k^i)$, i.e., how well it expects to be able to perform for the given allocations, as well as the sensitivities of its performance with respect to each of the allocated quantities, $\nabla_{\lambda_k^{i,j}\ \forall j,\ \bar{P}_k^i,\ R_k^i} J_k^i(\lambda_k^{i,j}\ \forall j,\ \bar{P}_k^i,\ R_k^i)$.

3) The master problem uses the sub-problem feedback to determine its overall performance, which is in essence the aggregation of the performance of the individual servers. The objective is to maximize the sum of the benefits achieved by all the servers, and, as such, it uses the sensitivity information to reallocate $\bar{P}_h^{DC}$, R_h^{DC} and λ_h^{DC} among the servers so as to maximize overall data center performance.

In this paper we focus on the individual server sub-problem (the 2^{nd} step above), which is the main building

block in our data center RS provision optimization framework. For notational convenience, in the rest of the paper, we do not denote the time slot (h or k) or the server (i). In addition, jobs are no longer identified by job type j but instead by designation of priority and application type.

B. Single Server problem

The single server problem has two coupled objectives:

1) For given targets $\bar{P}$ and R, it must determine an optimal dynamic policy that maximizes its ability to track the regulation signal $z(t)$ while maintaining acceptable QoS in servicing applications;
2) It must search for the optimal target levels $\bar{P}$ and R that can be bid at the hour ahead power market to minimize the energy cost, provided that the optimal dynamic policy will be used during the hour.

We proceed by defining tracking and QoS performance. The server needs to control its real time power consumption $P(t)$ so as to minimize the *tracking error* defined as $\epsilon(t) = |P(t) - (\bar{P} + z(t)R)|/R$. The control policy at its disposal is a set of rules that (i) select a job in the queue to process and (ii) determine the speed, and hence the power usage rate, at which to process the job. The tradeoff at play is between better tracking, i.e., achieving a small $\epsilon(t)$, and worse QoS by introducing delays in processing jobs and forcing them to wait longer in queue. Categorizing jobs by priority p, (high, medium, low, etc.) and application type a, (e.g., blackscholes, bodytrack, etc.) we define by d_a^p, an instance of the system time, i.e., waiting time plus processing time, of a job of this type expressed as a factor of the shortest possible processing time of that job. Shortest possible processing time is the time if the server works on that job at its highest power consumption level, and hence at the fastest possible service rate. We denote the system time measured in this way by d to underscore the fact that it can be related to the degradation in the job's processing time relative to the situation in which the server is dedicated to work on that job as soon as it arrives. $d_a^p = 1$ means there is no degradation at all, while $d_a^p > 1$ represents various levels of degradation. The performance of a dynamic policy for given $\bar{P}$ and R targets is quantified by the mean and standard deviation of the tracking error $\epsilon(t)$ and the system time of each job type d_a^p, which we denote by $\bar{\epsilon}$, σ_ϵ and $\bar{d_a^p}$, $\sigma_{d_a^p}$.

Observing frequency distributions that are prevalent in the simulated sample trajectories of $\epsilon(t)$ and d_a^p, we notice that they fit the Gamma distribution $\Gamma(k, \theta)$ with shape k: $k_\epsilon = \bar{\epsilon}^2/\sigma_\epsilon^2$, $k_{d_a^p} = \bar{d_a^p}^2/\sigma_{d_a^p}^2$ and scale θ: $\theta_\epsilon = \sigma_\epsilon^2/\bar{\epsilon}$, $\theta_{d_a^p} = \sigma_{d_a^p}^2/\bar{d_a^p}$. We therefore use the Gamma distribution[3] to form probabilistic constraints from estimated mean and variance statistics.

Mean and standard deviation estimates of $\epsilon(t)$ define the RS signal tracking performance, which on one level affects RS revenues, and on another, if a tolerance is exceeded with an agreed upon probability, the data center is disqualified from participation in the power market. Mean and standard deviation estimates of job system time are used to construct QoS guarantee constraints that are specified in data center client contracts. For example, a high paying customer may require that system time of jobs it sends to the data center does not exceed a certain tolerance upper bound at some agreed upon probability or confidence level, while a low paying customer would have a relatively larger tolerance.

Finally, denoting the known limits on $P(t)$ (i.e., the limits determined by the specific hardware) by P_{max} and P_{idle}, the tolerances agreed upon between the data center and the ISO as ϵ^{tol}, tolerances between the data center and its clients by $d_a^{p,tol}$, and the statistical confidence level by P_{conf}, we formulate the following optimization problem:

$$\begin{aligned} \underset{\bar{P},R,dynamic\ policy}{\text{maximize}} \quad & \Pi^R R - \Pi^E \bar{P} - \Pi^R c[\sigma_\epsilon^2 + \bar{\epsilon}^2] \\ \text{subject to} \quad & \Gamma^{-1}(k_\epsilon, \theta_\epsilon, P_{conf}) > \epsilon^{tol}, \\ & \Gamma^{-1}(k_{d_a^p}, \theta_{d_a^p}, P_{conf}) > d_a^{p,tol}, \\ & \bar{P} + R < P_{max}, \\ & \bar{P} - R > P_{idle}, \\ & \bar{P} \geq 0, R \geq 0. \end{aligned} \tag{1}$$

where Π_R is the hour-ahead clearing price of RS reserves ($/kWh), Π_E is the hour-ahead clearing price of energy ($/kWh), and c is the penalty associated with the second moment of the tracking error.

The problem above is a hard problem that requires the solution of an optimal stochastic dynamic programming problem for each set of targets, $\bar{P}$ and R. In this exploratory paper, we do not attempt to solve for the optimal policy, leaving this task to future work. Instead we select a reasonable real-time tracking policy described by the following rules:

1) Upon completion of servicing a job, the server selects a new job from the highest priority queue that is not empty, using a first-in first-out (FIFO) protocol.
2) The server has a range of power consumption rates for that job as described in detail in the next section. The policy, in real time, selects an allowable consumption rate that minimizes the tracking error $\epsilon(t)$.

Given this simple yet reasonable policy we simulate a 1-hour period multiple times and estimate the mean and standard deviation performance measure statistics described earlier. We perform a search over the target values $\bar{P}$ and R at a sufficiently fine granularity. An alternative approach that we explore is to estimate the sensitivities of the performance measure statistics with respect to the target parameters $\bar{P}$ and R, and use these sensitivities to perform a more structured search. In the next section we describe the server model and our simulation methodology.

IV. MODELS AND METHODOLOGY

In this section, we introduce the models and the system simulation methodology for solving the problem formulated in Section III and describe our experimental infrastructure on a real life server.

[3]We also did experiments by using Gaussian and uniform distribution. The proposed solution produces similar results in all the cases.

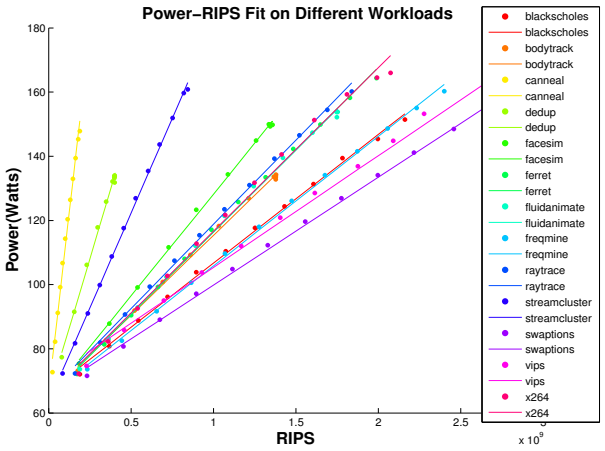

Fig. 3. Power-RIPS models on PARSEC workloads using linear fit. The dots represent the collected data and the lines show the model fit.

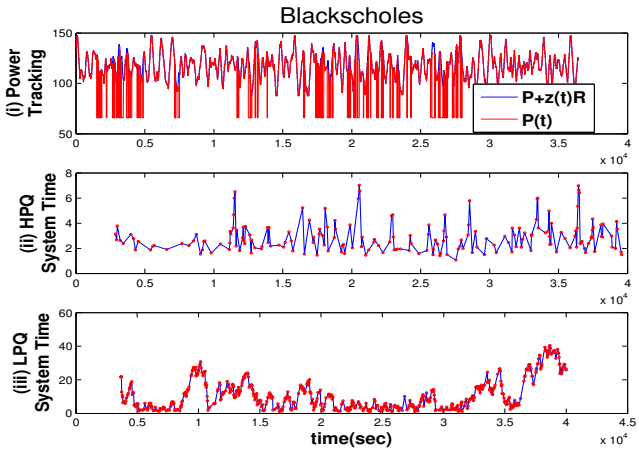

Fig. 4. Power-performance profiles of Blackscholes with $\bar{P} = 117.65W$ and $R = 30W$. (i) $\bar{P} + z(t)R$ and $P(t)$ trajectories (in Watts) over a 11-hour period (10 replications of a 1-hour period, the first hour data is not used because of the warming up process). The tracking error statistics: $\bar{\epsilon} = 0.20$, $\sigma_\epsilon = 0.60$. (ii) HPQ system time degradation (the system time as a factor of the shortest job processing time on our sever) for each job arrival shown as a red dot on the time trajectory. The overall statistics: $\bar{d}^H_{bls} = 2.91$, $\sigma_{d^H_{bls}} = 1.22$. (iii) LPQ system time degradation for each job arrival shown as a red dot on the time trajectory. The overall statistics: $\bar{d}^L_{bls} = 11.27$, $\sigma_{d^L_{bls}} = 9.31$.

A. The System Model

A single server in a data center can be abstracted into two parts: the job waiting queues and the processor. When a job arrives, it waits in a FIFO queue until it is scheduled to run on the processor. In our model we assume that the processor runs only one job at a time, while there are other jobs waiting in queues with various priority levels (e.g., a high priority job queue and a low priority queue).

We assume that jobs arrive to the server following an exponential distribution. Jobs with different priorities arrive at different arrival rates; i.e., higher priority jobs have a lower arrival rate, as in general the number of higher priority jobs will be smaller than that of the low priority jobs. Similarly, high priority jobs are more urgent in general, and hence operate under tighter delay constraints. In order to minimize the delay, we design a policy to always serve the high priority job queues before servicing low priority job queues.

We generate the job queues using Monte Carlo simulation. For each job with priority p, a random number r_j is generated. r_j is used to determine the job arrival time interval; i.e., $\tau^p_j = -ln(1 - r_j)/\lambda_p$, where λ_p is the arrival rate of the job with priority p.

B. The Processor Model

A job running on a server is composed of a number of instructions. Finishing a job is equivalent to executing all its instructions. Retired Instructions Per Second (RIPS) is a metric showing the number of instructions finished in each second, and is commonly used for evaluating the performance of the processor. A higher RIPS represents a faster processor service rate.

Our goal is to provide RS without violating performance constraints; therefore, a model between the processor service rate and the corresponding power consumption is needed. In order to design a practical and effective model, we conduct our experiments on a 1U server that has an AMD Magny Cours (Opteron 6172) processor, which has 12 processing cores on a single chip. The server is virtualized by the VMware vSphere 5.1 ESXi hypervisor. We use the *resource limits* control knob in the hypervisor to control the power-performance settings at runtime. Resource limits enable dynamically changing the resources allocated to a virtual machine (VM) quickly and at a fine granularity. For example, cutting the resource limits for a VM to half of the original setting (without limits) cuts the server's power consumption also to half of the original power level while running that VM. Similarly, we can set the performance of the server to any level we need. As more than 50% of the data center servers are virtualized today, controlling the power and performance for the applications through changing the resource limits for the VM is a practical and efficient method.

We run each application from the PARSEC-2.1 [18] benchmark suite on a VM in isolation (by itself, without consolidation) in our experiments and apply regression on the data collected to derive the model between the power consumption and RIPS. We construct the following model with a mean square error of less than 5%:

$$P_a = C_a * RIPS_a + P_{idle} \quad (2)$$

where P_a and $RIPS_a$ are the power consumption and the RIPS of job type a, C_a is a constant which is specific to the job type a, and P_{idle} is the power consumption of the idle states of the processor (when service rate is 0). The data and model fits are shown in Figure 3. Next, we discuss the simulation and experimental results.

V. EXPERIMENTAL RESULTS

This section introduces ours experimental settings and evaluates the proposed policy and method.

A. Experimental Settings

Workloads used in our experiments are from the PARSEC-2.1 [18] benchmark suite. PARSEC-2.1 contains 13 multi-

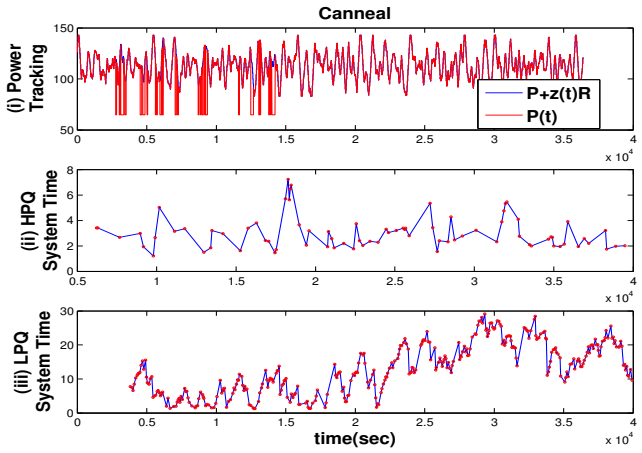

Fig. 5. Power-performance profiles of Canneal with $\bar{P} = 113W$ and $R = 30W$. (i) $\bar{P} + z(t)R$ and $P(t)$ trajectories (in Watts). The tracking error statistics: $\bar{\epsilon} = 0.09$, $\sigma_\epsilon = 0.40$. (ii) HPQ system time degradation trajectory with overall statistics: $\bar{d}^H_{can} = 3.03$, $\sigma_{d^H_{can}} = 1.28$. (iii) LPQ system time degradation trajectory with overall statistics: $\bar{d}^L_{can} = 13.05$, $\sigma_{d^L_{can}} = 7.28$.

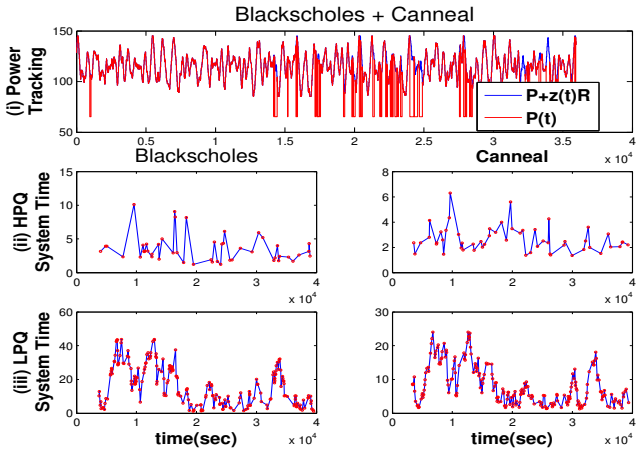

Fig. 6. Power-performance profiles of Blackscholes and Canneal arriving at the server with $\bar{P} = 115.33W$ and $R = 30W$. (i) $\bar{P} + z(t)R$ and $P(t)$ trajectories (in Watts). The tracking error statistics: $\bar{\epsilon} = 0.20$, $\sigma_\epsilon = 0.58$. (ii) HPQ system time degradation trajectory with overall statistics: for Blackscholes $\bar{d}^H_{bls} = 3.53$, $\sigma_{d^H_{bls}} = 2.02$ and for Canneal $\bar{d}^H_{can} = 2.64$, $\sigma_{d^H_{can}} = 1.01$. (iii) LPQ system time degradation trajectory with overall statistics: for Blackscholes $\bar{d}^L_{bls} = 16.84$, $\sigma_{d^L_{bls}} = 11.63$ and for Canneal $\bar{d}^L_{can} = 8.36$, $\sigma_{d^L_{can}} = 5.77$.

threaded applications. We first run each of them on our 12-core AMD Magny Cours based server, record their dynamic power ranges (P_{idle} to P_{max}), shortest possible processing time (used for system time degradation calculation mentioned in Section III) and power consumption values at different RIPS levels (See Section IV). We then simulate the server power regulation using these data and models.

Without loss of generality, we assume that all jobs are classified in two priority levels: high and low. Thus, we have one High Priority Queue (HPQ) and one Low Priority Queue (LPQ). We set job arrival rates to achieve a system utilization around 50%; i.e., the server is processing jobs around 50% of the whole time period, and is in idle state for the rest of time. Such utilization level is typical in today's data centers. We assume that the arrival rate of the low priority jobs is three times larger than that of high priority jobs. Finally, in order to measure statistics described in Section III, we simulate a 1-hour period 10 times to achieve statistical confidence.

B. The Capability Test for Server RS Provision

Generally, three capabilities are going to be tested based on the ISO requirements if a server requests a license to provide RS: keeping power consumption value stable for a period of time; ramping up and down to $\bar{P} + R$ and $\bar{P} - R$ within 2.5-5mins; and regulating the power up and down at each time interval at a sufficiently fine granularity [21]. In addition, a sufficiently large dynamic power range is needed for the RS provision.

Our real-life server experimental result shows: (i) *Dynamic Power Range*: by running all the workloads selected from the PARSEC-2.1 benchmark suite on our server, we achieve a power range from P_{idle} = 66W to P_{max} = 130-170W (maximum value depends on the job). Hence, ideally the maximum regulation power value that can be provided is $(P_{max} - P_{idle})/2$ = 32-52W, which is approximately 25-31% of P_{max}. (ii) *Power Stability*: we fix the resource limit setting at a specific value for a while and observe the fluctuation of power. The standard deviation of the power consumption is 1-3W, which is only 1-2% of $\bar{P}$. (iii) *Ramping up Capability*: from test our server is able to ramp up from 66W to 153W (P_{idle} and P_{max} of Blackscholes) at 1s intervals. (iv) *Granularity of Regulation*: Resource limit settings enable regulating the power at a granularity of a few milliwatts. Overall, we conclude that our server has the sufficient fundamental capabilities in providing RS.

C. Tracking Capability and QoS Evaluation

In general, different types of jobs have different dynamic power ranges, shortest processing time and power-performance models. When we formulate the optimal problem in Eq.(1), we need to separately determine performance constraints for each type of job. In this section, we discuss the tracking capabilities and evaluate QoS by using our policy for both homogeneous and heterogeneous sets of jobs running on the server.

1) The homogeneous Job System: We first investigate the performance under the circumstance that all jobs arriving at the server are of the same type. Figure 4 and Figure 5 show the power tracking along with HPQ and LPQ system time degradations of the jobs *Blackscholes* and *Canneal*, each with a corresponding $(\bar{P}, R)$ setting. We see from the power tracking figures that at the given $(\bar{P}, R)$ setting and under the policy proposed in Section III, our system is able to track the regulation signal with a very small error. Larger errors only appear when both queues are empty, in which case the processor is forced to stay in the idle state with a power consumption of P_{idle} and cannot be regulated. The system time degradation figures show that the degradation of LPQ is much larger than that of HPQ. This is because LPQ has a larger arrival rate and it is always served after the HPQ jobs are served.

TABLE I

THE MONETARY COSTS OF DIFFERENT JOB TYPES FOR VARIOUS ($\bar{P}$, R) SETTINGS

Blackscholes			Bodytrack			Canneal			Facesim			Streamcluster			Blackscholes + Canneal		
$\bar{P}$	R	Cost	$\bar{P}$	R	Cos	$\bar{P}$	R	Cost	$\bar{P}$	R	Cost	$\bar{P}$	R	Cost	$\bar{P}$	R	Cost
117.65	0	117.65	103.63	0	103.63	113	0	113.00	115.04	0	115.04	117.65	0	117.65	115.33	0	115.33
117.65	10	N/A	103.63	10	94.36	113	10	104.36	115.04	10	N/A	117.65	10	N/A	115.33	10	N/A
117.65	20	98.45	103.63	20	83.83	113	20	93.35	115.04	20	95.48	117.65	20	N/A	115.33	20	96.12
117.65	30	88.03	103.63	30	73.73	113	30	83.17	115.04	30	85.24	117.65	30	87.91	115.33	30	85.70
117.65	**35**	**82.94**	**103.63**	**31**	**72.72**	**113**	**33**	**80.14**	**115.04**	**34**	**81.20**	**117.65**	**35**	**82.84**	**115.33**	**34**	**81.63**
117.65	40	N/A	103.63	40	N/A	113	40	N/A	115.04	40	N/A	117.65	40	N/A	115.33	40	N/A
152.95	0	152.95	134.72	0	134.72	146.9	0	146.90	149.55	0	149.55	152.94	0	152.94	149.92	0	149.92

TABLE II

COMPARISON OF OPTIMAL REGULATION PROVISION AND PROVISION WITHOUT REGULATION

		$\bar{d}^H$	σ_{d^H}	$\bar{d}^L$	σ_{d^L}	Cost($/h)
Blackscholes (117.65, 35/0)	Optimal:	3.01	1.40	11.58	9.60	82.94
	Non-Reg:	2.73	0.99	10.92	9.31	117.65
Bodytrack (103.63, 31/0)	Optimal:	3.27	1.27	13.65	7.22	72.72
	Non-Reg:	2.97	0.83	12.37	7.01	103.63
Canneal (113, 33/0)	Optimal:	3.06	1.35	13.17	7.35	80.14
	Non-Reg:	2.64	0.76	12.30	6.97	113.00
Facesim (115.04, 34/0)	Optimal:	2.53	0.70	7.00	3.53	81.20
	Non-Reg:	2.71	0.63	7.11	3.44	115.04
Streamcluster (117.65, 35/0)	Optimal:	2.33	0.62	6.43	3.27	82.84
	Non-Reg:	2.46	0.55	6.52	3.18	117.65
Blackscholes + Canneal (115.33, 34/0)	Optimal, Bls:	3.73	2.20	16.94	11.70	81.63
	Optimal, Can:	2.72	1.05	8.44	5.79	
	Non-Reg, Bls:	3.18	1.16	17.17	12.09	115.33
	Non-Reg, Can:	2.42	0.57	8.59	5.92	

Blackscholes is a CPU-intensive job while Canneal is a memory-intensive job; i.e., they differ in the way they use the processor resources. However, they achieve similar results in both tracking and QoS performances, which implies that providing RS is not constrained by the job type.

2) The Heterogeneous Job System: We next investigate a heterogeneous case; i.e, jobs arriving at the server are of different types. Without loss of generality, we assume all the jobs are either Blackscholes or Canneal. Figure 6 shows the power tracking and the system time degradations of each job type separately at $\bar{P} = 115.33W$ and $R = 30W$. The result has limited difference compared to the homogeneous case, which implies that power RS can be provided for a set of heterogeneous jobs arriving at the system.

D. The Optimal Solution and the Monetary Savings

Providing RS helps the data center reduce energy costs. The objective function in Eq. (1) is based on the monetary cost for *a single server per hour*, when the server consumes power at an average level $\bar{P}$ W and provides RS of R W. A data center generally contains thousands of servers. Table I shows the monetary costs ($/h) for a data center which has 10^4 servers of the same type running both homogeneous and heterogeneous job cases at various ($\bar{P}$, R) values. The yellow highlighted line is the optimal solution of the Eq. (1) solved by brute force method at a sufficiently fine granularity. In solving Eq. (1), the following parameters are used: $\Pi^R = \Pi^E = 0.1\$/kWh, c = 1, P_{conf} = 0.85, \epsilon^{tol} = 0.2, P_{idle} = 66W$. $d^{H,tol} = 5 and d^{L,tol} = 25$ for all types of jobs. P_{max} changes between 130W and 170W depending on the job type.

The results show that in all cases, the solution is optimal when the regulation power R is around 30% of its corresponding $\bar{P}$, and 23% of the P_{max} (P_{max} of each job type is shown in the bottom row of the table). Such a result also implies that the optimal percentage of RS power provision does not change much among different types of jobs. In addition, comparing the monetary cost under the optimal solution ($\bar{P}$, R) to the one in the first row of the table, which does not have any RS provision (R=0), we can see that the monetary saving is approximately 30%, which is highly promising. Note that 'N/A' in the table means that there is no feasible solution for the corresponding ($\bar{P}$, R) pair according to Eq. (1).

Table II shows the comparison in system time degradation statistics (QoS performance) and monetary costs between the case of optimal regulation provision and the case of provision without regulation (Non-reg.) for different job types. We see that the QoS values in these two cases are very close. Thus, we do not sacrifice much QoS, while we are able to save 30% monetary costs by providing RS.

E. Sensitivity Analysis

In real-life data centers, the QoS performance requirements and the utilization (job arrival rates) of the system frequently change. As a result, the optimal operating point ($\bar{P}$, R) needs to be adjusted. Sensitivity analysis studies how tracking error and system time degradations vary if ($\bar{P}$, R) changes, and provides information on which direction to search for the new optimal point. Thus, sensitivity analysis can highly improve the efficiency of the brute force method. Many approaches have been proposed for performing sensitivity analysis. In this experiment, we focus on the *Finite Difference* [11] method.

Figure 7 shows the changes of tracking error, HPQ and LPQ degradation statistics while either $\bar{P}$ or R is increased by 1% for the homogeneous case with Blackscholes. The results show that while increasing $\bar{P}$ by 1%, first, tracking error increases. This is because higher $\bar{P}$ increases the idle time of the system. Secondly, the LPQ system time degradation highly decreases, but the HPQ system time degradation has no notable change. As expected, increasing $\bar{P}$ leads to performance improvements, especially for LPQ which has a higher number of jobs. However, HPQ jobs are always given priority for execution. Hence, the improvement in HPQ performance when we increase $\bar{P}$ is limited. On the other hand, while increasing R by 1%, neither tracking nor QoS performance have obvious changes. Such results show that both tracking and QoS performance are much more sensitive to $\bar{P}$ rather than R. Therefore, when designing a policy to

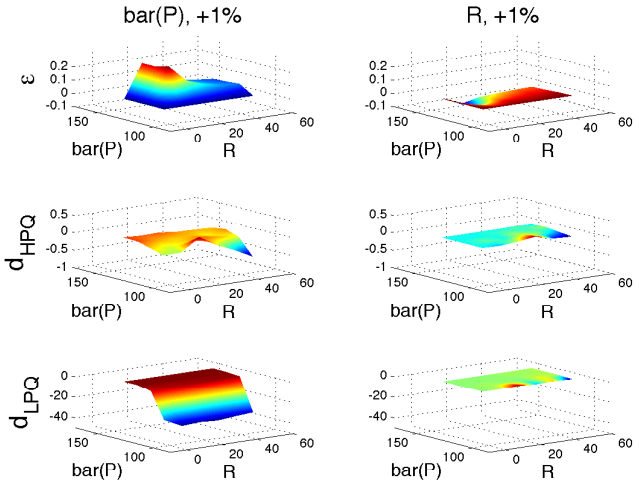

Fig. 7. Sensitivity analysis for Blackscholes jobs. Plots in the left column are results for increasing $\bar{P}$ by 1% and plots in the right column are results for increasing R by 1%. From top to bottom each pair of plots shows the changes in tracking error statistics $\Gamma^{-1}(k_\epsilon, \theta_\epsilon, P_{conf})$, changes in HPQ system time degradation statistics $\Gamma^{-1}(k_{d^H_{Bls}}, \theta_{d^H_{Bls}}, P_{conf})$, and changes in LPQ system time degradation statistics $\Gamma^{-1}(k_{d^L_{Bls}}, \theta_{d^L_{Bls}}, P_{conf})$.

search for the new optimal ($\bar{P}$, R), determining $\bar{P}$ based on new system restrictions and requirements is necessary before selecting R.

VI. CONCLUSIONS AND FUTURE WORK

This paper has investigated the ability of data centers to provide power RS reserves and has quantified associated market participation benefits. It has proposed both a data center-wide optimization framework and a solution to the single-server sub-problem. The proposed RS dynamic tracking policy sub-problem has been simulated over a broad range of policy parameters, namely RS reserves and average power consumption. The optimal values of these parameters have been identified under the objective of maximizing energy cost savings subject to probabilistic QoS guarantees and ISO RS signal tracking error constraints. Numerical evidence indicates that power regulation provision is not constrained by the type of server workload. We conclude that a single server is capable of providing RS and achieving energy cost savings by up to 30%. Finally, the paper has presented a sensitivity analysis, which can be leveraged for designing an efficient real-time multiple server coordination policy.

Our future work will focus on explicitly addressing the data center-wide RS problem and investigating the interaction of computing power consumption with the slower time scale cooling power consumption dynamics.

REFERENCES

[1] M. C. Caramanis, I. C. Paschalidis, C. G. Casandras, E. Bilgin, and E. Ntakou. Provision of Regulation Service Reserves by Flexible Distributed Loads. In *Proceedings, 51st IEEE Conference on Decision and Control*, pages 3694-3700, 2012.

[2] R. Cochran, C. Hankendi, A. Coskun, and S. Reda. Pack & cap: adaptive DVFS and thread packing under power caps. In *ACM/IEEE International Symposium on Microarchitecture*, 2011.

[3] M. Etinski, J. Corbalan, J. Labarta, and M. Valero. Optimizing job performance under a given power constraint in hpc centers. In *Proceedings of the International Conference on Green Computing*, pages 257–267, 2010.

[4] A. Gandhi, R. Das, J. Kephart, M. Harchol-Balter, and C. Lefurgy. Power capping via forced idleness. In *Proceedings of Workshop on Energy-Efficient Design*, 2009.

[5] A. Gandhi, M. Harchol-Balter, R. Das and C. Lefurgy. Optimal power allocation in server farms. In *Proceedings of the Joint Conference on Measurement and Modeling of Computer Systems*, pp. 157-168, 2009.

[6] M. Ghamkhari and H. Mohsenian-Rad. Data centers to offer ancillary services. In *IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pp. 436-441, 2012.

[7] C. Isci, A. Buyuktosunoglu, C.-Y. Cher, P. Bose, and M. Martonosi. An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*, pages 347–358, 2006.

[8] V. Kontorinis, A. Shayan, D. M. Tullsen, and R. Kumar. Reducing peak power with a table-driven adaptive processor core. In *Proceedings of MICRO*, pages 189–200, 2009.

[9] J. Koomey. Growth in data center electricity use 2005 to 2010. Oakland, CA: Analytics Press. August, 1, 2010.

[10] B. Kranz, R. Pike, and E. Hirst. Integrated electricity markets in New York. *The Electricity Journal*, 16(2):54 – 65, 2003.

[11] T. Maly and L. R. Petzold. Numerical methods and software for sensitivity analysis of differential-algebraic systems. *Applied Numerical Mathematics*, 20.1: 57-79, 1996.

[12] D. Meisner, B. Gold, and T. Wenisch. PowerNap: Eliminating server idle power. In *Proceedings of Architectural Support for Programming Languages and Operating Systems*, pages 205–216, 2009.

[13] K. Meng, R. Joseph, and R. P. Dick. Multi-optimization power management for chip multiprocessors. In *International Conference on Parallel Architectures and Compilation Techniques*, pages 177–186, 2008.

[14] A. Mohsenian-Rad and A. Leon-Garcia. Coordination of cloud computing and smart power grids. In *First IEEE International Conference on SmartGridComm*, pp. 368-372, 2010.

[15] *NYISO Day-Ahead Scheduling Manual 11, www.nyiso.com*, June 2001.

[16] A. L. Ott. Experience with PJM market operation, system design, and implementation. *IEEE Transactions on Power Systems*, 18(2):528–534, 2003.

[17] I. C. Paschalidis, M. C. Caramanis, and B. Li. A market-based mechanism for providing demand-side regulation service reserves. *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 2011.

[18] B. Christian. Benchmarking Modern Multiprocessors. Ph.D. Thesis. Princeton University, January 2011.

[19] PJM. *Description of Regulation Signals, www.pjm.com,* Dec. 2011.

[20] PJM. *White Paper on Integrating Demand and Response into the PJM Ancillary Service Markets*, Febuary 2005.

[21] *PJM Manual 12: Balancing Operations, www.pjm.com,* Dec. 2012.

[22] *PJM Manual 28: Operating Agreement Accounting, www.pjm.com,* June. 2013.

[23] K. K. Rangan, G.-Y. Wei, and D. Brooks. Thread motion: fine-grained power management for multi-core systems. In *International Symposium on Computer Architecture*, pages 302–313, 2009.

[24] L. Rao, X. Liu, L. Xie, and W. Liu. Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment. In *Proceedings of IEEE INFOCOM*, pages 1-9, 2010.

[25] S. Reda, R. Cochran, and A. K. Coskun. Adaptive power capping for servers with multi-threaded workloads. *IEEE Micro*, vol.32, no.5, pp.64-75, Sept.-Oct. 2012.

[26] R. Teodorescu and J. Torrellas. Variation-aware application scheduling and power management for chip multiprocessors. In *International Symposium on High-Performance Computer Architecture*, pages 363–374, 2008.

[27] X. Wang, M. Chen, C. Lefurgy, and T. Keller. SHIP: A scalable hierarchical power control architecture for large-scale data centers. *Parallel and Distributed Systems, IEEE Transactions on*, 23(1):168 –176, January 2012.

[28] X. Zhan and S. Reda. Techniques for energy-efficient power budgeting in data centers. In *Proceedings of the 50th Annual Design Automation Conference (DAC)*, p.176, 2013.