The Data Center as a Grid Load Stabilizer

Hao Chen
Electrical and Computer Engineering
Boston University
Boston, MA 02215
haoc@bu.edu

Michael C. Caramanis
Systems Engineering
Boston University
Boston, MA 02215
mcaraman@bu.edu

Ayse K. Coskun
Electrical and Computer Engineering
Boston University
Boston, MA 02215
acoskun@bu.edu

Abstract — To accommodate the increasing presence of volatile and intermittent renewable energy sources in power generation, independent system operators (ISO) offer opportunities for *demand side regulation service (RS)* so as to stabilize the grid load. These power market features allow the demand side to earn monetary credits by modulating its power consumption dynamically following an RS signal broadcast by ISO. This paper studies the capacities and benefits of a major potential demand side, *the data center*, to provide RS. We propose a dynamic control policy that modulates the data center power consumption in response to ISO requests by leveraging server power capping techniques and various server power states. Results demonstrate that using our policy, data centers can provide fast reserves in quantities that are substantial proportions (around 50%) of their average energy consumption, with no major deterioration in quality of service (QoS). By doing so, data centers decrease their energy costs around 50%, while providing the ISOs and the society in general with cost effective demand side reserves that render massive renewable generation adoption affordable.

I. INTRODUCTION

Unlike traditional electric grids, today's smart grids incorporate a larger percentage of intermittent renewable energy sources in power generation. These new volatile energy sources create challenges for grid operators to stabilize the grid load and match the power supply with demand in real time. Therefore, ISOs adopt novel mechanisms in modern power markets to ensure stability. Demand side regulation service (RS) is one such mechanism, where the participant receives monetary benefits upon regulating its power consumption based on ISO requests.

This paper focuses on evaluating the benefits of data center participation in the power market for providing demand side RS reserves. We focus on demand side RS reserves because RS reserve market clearing prices are, on average, as valuable in today's markets as energy clearing prices [1, 2]. More importantly, we focus on RS reserves because on one hand their requirements are expected to increase rapidly with increasing renewable energy integration in the grid [19], while on the other hand data centers have a comparative advantage in offering RS reserves relative to other demand side reserve providers. The ability of data centers to offer RS is indeed significant due to their degrees of freedom in modulating their power consumption and the diversity of jobs that they process ranging from high priority transactional jobs to less sensitive jobs that require a reasonable processing rate on average rather than an immediate response. The investigation of the ability of data centers to offer reserves is quite opportune given their increasing share in power consumption, which is 3% of total US electricity [16].

A considerable body of prior research has introduced techniques to reduce energy consumption of processors, servers, and of entire data centers (e.g., [17, 15, 20]). Rather than minimizing the energy consumption, this paper focuses on optimizing the design and real-time operation of data center power consumption in a way that offers RS reserves to ISOs in advanced power markets while, at the same time, maintaining appropriate levels of QoS to data center loads.

The main contributions of this paper are as follows: (1) We design a dynamic power control policy that enables data center to accurately track the RS signal power cap with no major deterioration in QoS, by leveraging server power capping techniques and various server states. (2) We introduce a method to estimate the average power consumption and regulation reserve amounts for data centers to bid in hour-ahead power markets. (3) We demonstrate the capabilities of data centers to participate in RS provision, and through this participation, the energy costs can be dramatically reduced by around 50%.

The rest of this paper starts with an overview of power markets and RS provision. Section III discusses our dynamic power control policy and how to estimate the average power consumption and the regulation reserves. Section IV describes the simulation methodology and provides the experimental results of data center RS provision. Section V discusses the related work and Section VI concludes the paper.

II. POWER MARKET AND REGULATION SERVICE

Power markets, introduced in the US in 1997 [23], have been widely adopted. Today they serve the majority of high-voltage-connected generators and large consumers. Soon after their introduction, power markets evolved to co-optimize or co-clear energy and capacity reserves (primary for frequency control, secondary for RS, tertiary, etc.), whose system-level requirements reflect contingency planning for uncertainty in energy balance, transmission, and generating capacity availability. Social-welfare contributions of competitive power markets are arguably due to the fact that they enable distributed, yet collaborative, decisions which (i) take advantage of lo-

978-1-4799-2816-3/14/$31.00 ©2014 IEEE

cally known uncertainty and dynamical-response-capability information, and (ii) can respond efficiently to price or other system-wide state sufficient statistics, such as frequency and Area Control Error (ACE) and associated reserve requirement signals. These sufficient statistics enable local decisions to be made efficiently and in a manner that is adaptive to power system requirements.

Synchronized power systems may become unstable when generation and consumption are not carefully balanced in practically real-time. To this end, ISOs solicit and secure sufficient quantities of a mix of reserves with different dynamic delivery properties. Bi-directional reserve contracts are secured at least an hour in advance and promise to respond in real-time to ISO-broadcasted fast changing system requirements. Fast reserves include primary (or frequency control) and secondary (or RS) reserves, and are more valuable than slower reserves such as spinning reserves. Each type of reserves is characterized by (i) the frequency with which the system-wide delivery request signal is updated, and (ii) a response time or speed at which that request must be met by each reserve provider. For example [1, 2], an RS reserve contract agreed upon an hour in advance promising to offer during the hour up to R MW of reserves, is obligated to respond to an ISO signal $z(t)$ re-broadcasted every 4 seconds, i.e., broadcasted at time $t = 0, 4, 8, ..., 3596$ sec and follow a response time of 300 seconds, namely a speed of $1/300$ percent per second. More precisely, $z(t)$ takes values in the interval $[-1, 1]$ setting a consumption target at time t of $P(t+4) = \bar{P} + z(t)R$ where $\bar{P}$ is an average consumption set at the same time that R is offered in the hour ahead market. The demand side provider must change its consumption from $P(t)$ to $P(t+4)$ at a positive or negative rate of change equal to $R/300$ MW/sec. In fact, the values of $z(t)$ are the outputs of an ISO specified integral proportional filter of system frequency and balancing ACE, and, as such, they are unpredictable and unaffected by a single individual market participant's behavior. The statistical behavior of $z(t)$, however, is well known at the beginning of the hour. Its average value over an hour is zero; i.e., the RS signal trajectory encourages energy neutral consumption modulation trajectory.

The objective of a data center decision support framework considered in this paper is to optimize the following hybrid discrete event system: Given discrete probabilistic arrivals of processing requests (jobs) and a well-defined stochastic process describing $z(t)$, determine a dynamic optimal control policy that maps the system state $x(t)$ to action $u(t)$ where:

- $x(t)$ contains (i) the state of servers in the data center: active, idle, asleep, off, in transition, (ii) the jobs waiting in buffer queues or being processed, (iii) the current value of $z(t)$, and (iv) the QoS achieved so far.
- $u(t)$ is a member of the allowable control set containing (i) initiation of server state transitions, (ii) assignment of jobs to servers and to virtual machines (VMs), (iii) rerouting jobs to other data centers, and (iv) taking power and performance management actions (e.g., CPU resource limits control in VMs, dynamic voltage and frequency scaling -DVFS-, etc.) at individual servers.

State dynamics depend on $u(t)$ and evolve with multiple time scale hybrid dynamics responding to discrete control actions (e.g., a server state transition), discrete events (e.g., a job arrival), and continuous retired instruction rates in response to server power management settings.

Recent work [5, 6] on the decision support framework indicates that substantial cost reduction opportunities exist when we regulate the computational power in accordance with ISO RS requests. In this paper we show that after determining $\bar{P}$ and R levels based on workload estimates, physical limits of the servers, and constraints on performance and tracking error, it is possible to design a dynamic policy that maintains the desired QoS level while tracking the ISO signal with a small error. This ability translates into cost savings. Components of the optimization problem (workload estimation, determining $\bar{P}$ and R, dynamic policy, etc.) individually and their interactions introduce interesting yet complex challenges.

Before proceeding with a concrete proposal of the decision support framework used to investigate data center RS reserve offering, We describe the relevant features and pricing rules of RS transactions in the PJM and NYISO Power Markets [1, 2].

Consider a data center that purchases in the hour-ahead market $\bar{P}$ MWh of energy at the clearing price Π^E, and sells RS reserves R MW at the RS reserve clearing price or Π^R per MW traversed by the RS signal $z(t)$. The net cost of this transaction incurred by the data center when the hour ahead market clears is $\Pi^E\bar{P} - \Pi^R R$, provided that the data center tracks the RS signal $z(t)$ perfectly, modulating its power consumption to track the implied obligation $P(t+4) = \bar{P} + z(t)R$ perfectly. However, perfect tracking is practically not possible. Hence at the end of the hour, RS reserve providers are charged an amount that reflects their relative tracking error (RTE). Moreover, if the RTE exceeds a certain threshold[1], the participant loses its qualification to participate in RS reserve transactions and has to repeat a rigorously defined qualification process to re-qualify [1]. More precisely, the RTE is the ratio of the sum, or tracking error per MW or R, over the length of the trajectory traversed by $z(t)$, namely:

$$RTE = \frac{\sum_{t=0,4,8...,3596} |P(t+4) - (\bar{P} + z(t)R)|}{\sum_{t=0,4,8...,3596} |z(t+4) - z(t)|R} \quad (1)$$

At the end of the hour an RS provider is charged an additional cost equal to $\Pi^R R * RTE$. Note that if during the hour the data center observes the RS signal $z(t)$ perfectly and modulates its power consumption to track the implied obligation $P(t+4) = \bar{P} + z(t)R$ perfectly, then $RTE = 0$.

Note also that independent of power market transaction charges discussed above, a data center that provides RS reserves is bound to incur intrinsic costs from the operational level obligation to consume at or close to $P(t+4) = \bar{P} + z(t)R$. These consist of energy consumption efficiency losses associated with power consumption modulation as well as the value of possible reductions in the QoS provided to data center clients during, for instance, low $z(t)$ regulation signal values. Desirable operational level policies discussed in the next section must provide a reasonable tradeoff between market related charges and the above intrinsic costs. Moreover, they should ensure that the probability of exceeding maximum allowed RTE levels as well as client QoS guarantees is within pre-specified confidence intervals.

[1] Usually 30%.

III. DYNAMIC POWER CONTROL POLICY AND REGULATION RESERVES BIDDING

In this section, we first discuss a general data center model and various server states that are useful in power regulation. Then we propose a power control policy that dynamically modulates the data center power consumption $P(t+4)$ at each time interval t to track the RS signal related power capping, $\bar{P} + z(t)R$. We then introduce our estimates of the average power consumption, $\bar{P}$, and the regulation reserves, R, which the data center needs to bid in the hour-ahead power markets.

A. Data Center Model and Server States

A data center consists of computational nodes (servers) and cooling elements (computer room air conditioners, etc.). In this paper, we specifically focus on regulating the computational power. Our technique, however, can be combined with power budgeting techniques [31] that distribute a given total power cap into power caps of the sub-components of the data center.

Each server in the data center is in one of three states: active, idle and sleep [15]. When a server is running a job, it is "active", and its power consumption is P_{active}. P_{active} is composed of the dynamic power, P_{dyn}, and the static power, P_{static}. The dynamic power changes based on the characteristics of the running job, and can be modulated by power management techniques, such as DVFS [17], CPU resource limits [13], etc. The static power is a constant[2], and exists as long as the server is turned on. In our work, we use the CPU resource limits knob to modulate the server dynamic power. CPU resource limits change the resources allocated to a VM on the server, and as a result, adjust the server dynamic power and the throughput. It is a desirable control knob as it can be quickly changed at a very fine granularity [13, 6].

Regulating the dynamic power affects the server throughput. Previous work has shown a linear relation between P_{dyn} and the server throughput, represented by the retired instructions per second (RIPS), as $P_{dyn} = k * RIPS$ [6]. The server reaches its maximal throughput capacity by running at the peak power consumption rate, P_{peak}, with $P_{peak} = P_{dyn,max} + P_{static}$, where $P_{dyn,max}$ is the maximal dynamic power consumption that the server can achieve when the CPU resource limit is set to maximum.

A server is "idle" if it is turned on but is not running any jobs. An idle server consumes power at a constant rate, P_{idle}, which is equal to P_{static}. In the "sleep" state, the server consumes a very low constant power, P_{sleep}. In general, there are some time delays and energy costs of resuming a server from or suspending it to a "sleep" state. The suspending time delay, t_{susp}, usually is small and can be ignored, while the resuming time delay, t_{res}, is large [11, 15]. During both the suspending and resuming periods, the power consumption are similar, denoted as P_{tran}, which is often close to the peak power, P_{peak} [15]. The energy cost of the resuming period is $E_{loss} = t_{res} * P_{tran}$ and of the suspending period is ignored.

In fact, some servers in data centers can be completely turned off, which indicates a fourth state, "off", with no power consumption. However, the "off" state does not frequently appear due to the very large time delays and energy costs of resuming and suspending process. We do not consider the "off" state in this paper.

We assume there is a FIFO (first in first out) queue for holding the incoming jobs in the data center. Once a job arrives, it is first put into the queue. The job at the front of the queue is scheduled to a server using the policy introduced in Section III-B. In our model, each server can only serve one job a time; thus, we do not consider server consolidation.

We define the data center utilization U as the active time of the whole data center, or the number of active servers at each time interval. For example, $U = 50\%$ means each server is active for half of the whole period, and is in idle or sleep state for the rest of the time. We can also comprehend this as, at each moment, half of total servers in the data center are serving jobs. U is related to the arrival frequency of the workloads.

In this work, we study homogeneous data centers only, where all servers and jobs are of the same type. In fact, a heterogeneous data center with different types of servers and workloads can be split into homogeneous clusters. Also, many high performance computing (HPC) clusters include dedicated, optimized set of servers assigned to specific jobs.

B. Dynamic Power Control Policy

For real-time dynamic power tracking, we need to modulate the data center power consumption rate, $P(t+4)$, to match the dynamic RS signal related power value, $\bar{P} + z(t)R$. At the same time, workload QoS and overall energy waste also need to be considered. Our goals during the tracking process are as follows:

- Reduce the tracking error $|P(t+4) - (\bar{P} + z(t)R)|$;
- Improve the energy efficiency, including reducing the energy waste during the server state transition period, and reducing the static energy waste related to P_{static};
- Reduce the workload QoS performance degradation.

Apparently, there are tradeoffs among these goals. For example, reducing the tracking error prevents the servers from always running at their maximal capacity, which leads to performance degradation. Also, reducing the energy waste during the server state transition period requires reducing the number of server transitions, and reducing the static energy waste requires setting a fewer number of servers in idle and a larger number of servers in sleep mode. Both of these actions might violate the power tracking goal. Hence, our policy aims to optimize among these goals at each time interval by solving the following optimization problem:

$$\min_{u(t)\in U(x(t))} J(x(t),u(t)) = \alpha_1|P(t+4) - (\bar{P} + z(t)R)| + \alpha_2 N_{tran}(t) - \alpha_3 N_{sleep}(t) - \alpha_4 N_{peak}(t) \quad (2)$$

where u(t) is our policy control, $x(t)$ is the dynamic state at t. $x(t) = (z(t), Q(t), S_i(t), P_i(t), J_i(t), R_i(t), t_{idle,i}(t), i = 1, 2..., N_{dc})$. $z(t)$ is the RS signal at time t, $Q(t)$ is the number of jobs waiting in the queue for scheduling. $S_i(t), P_i(t), J_i(t), R_i(t), t_{idle,i}(t)$ are the server state (active, idle, sleep), power consumption, the job in the server ($J_i(t) = 0$: has no job in the server i; $J_i(t) = 1$: has a job), the remaining number of instructions of the job in the server, and the time

[2]The static power, in fact, is temperature dependent. In this work, we assume there is no temperature change.

of being in the idle state, of the server i, respectively. N_{dc} is the number of the servers in the data center, α_1, α_2, α_3 and α_4 are penalty coefficients. $N_{tran}(t)$ is the number of servers at time t that are suspending to or resuming from the sleep state, $N_{sleep}(t)$ is the number of servers in sleep, and $N_{peak}(t)$ is the number of servers running at their peak capacities. We have:

$$P(t) = \sum_{i=1}^{N_{dc}} P_i(t) \tag{3}$$

$$N_{sleep}(t) = \sum_{i=1}^{N_{dc}} \{S_i(t) == \text{“}sleep\text{”}\} \tag{4}$$

$$N_{tran}(t) = |N_{sleep}(t) - N_{sleep}(t-1)| \tag{5}$$

$$N_{peak}(t) = \sum_{i=1}^{N_{dc}} \{P_i(t) == P_{peak}\} \tag{6}$$

$$N_{idle}(t) = \sum_{i=1}^{N_{dc}} \{S_i(t) == \text{“}idle\text{”}\} \tag{7}$$

We include $N_{tran}(t)$ in Eq. (2) in order to reduce the transition energy waste. To reduce the static energy waste, we need to set a fewer number of servers in idle or non-peak active state; i.e., we need to increase the number of servers running at their peak capacity and put more servers in sleep state. Therefore, we include $N_{sleep}(t)$ and $N_{peak}(t)$ in Eq. (2).

We set different penalty weights for each goal by changing α_1, α_2, α_3 and α_4. For example, if the power tracking is the most important goal, we can simply assign α_1 much larger than α_2, α_3 and α_4. Then the optimal solution first aims at satisfying the power tracking constraint as accurate as possible at each t. We design some additional constraints and rules as follows:

- If $J_i(t) = 1$, i.e., the server is running a job, then the server must keep active, i.e., $S_i(t) = 1$, until the job is finished. In this way we provide guarantees for workload QoS. Furthermore, we set a lower bound of the minimal power rate P_{min} when serving a job, i.e., $P_i(t) \geq P_{min}$ if $J_i(t) = 1$, which forces the job to be served at a throughput with a lower bound and avoids the job being stalled in the server. P_{min} can be determined by the QoS requirements.
- Once a job is finished, i.e., $R_i(t) = 0$, the server immediately becomes idle.
- When $Q(t) = 0$, i.e., no jobs are waiting in the queue, then no idle server is allowed to be activated[3].
- Transition mechanism: if a server has been in idle longer than a timeout period, t_{tout}, then it automatically sleeps. This timeout mechanism is designed to avoid frequent transitions. We use the timeout value proposed by Gandhi et al. [11]: $t_{tout} = t_{res} * P_{peak}/P_{idle}$. In addition, in order to maximize the number of sleeping servers, we always select the server with smallest current $t_{idle}(t)$ to activate if a job is waiting to be served. Similarly, if we need to force some servers to sleep, we select the servers with the largest $t_{idle}(t)$.

[3]In fact, it is possible to run synthetic workloads to help improve the power tracking performance. In this work we do not consider such loads.

Having these rules, our available control $u(t)$ can be (1) increase/decrease power consumption of active servers by using CPU resource limits; (2) resume sleeping servers; (3) put idle servers to sleep; (4) activate idle servers to run new jobs.

In our work, we assign a large value to α_1 to put power tracking as the high-priority goal. In addition, we set α_2 larger than α_3 and α_4, i.e., we are more reluctant to do the server state transition. The resulting policy from Eq.(2) is:

Case 1- If $P(t) < \bar{P} + z(t)R$, i.e., the power consumption needs to be increased:

- Increase power consumption of some active servers that are not running at maximal capacity to P_{peak};
- If $Q(t) > 0$ and $N_{idle}(t) > 0$, then activate some idle servers and run them at maximal capacity with power consumption at P_{peak};
- Resume sleeping servers following the transition mechanism.

We do the above three steps in order until $P(t+4)$ meets the power cap, $\bar{P} + z(t)R$.

Case 2- If $P(t) > \bar{P} + z(t)R$, i.e., the power consumption needs to be decreased:

- Decrease power consumption of some active servers that are not running at maximal capacity to P_{min};
- Decrease power consumption of some active servers that are working at maximal capacity to P_{min};
- Suspend idle servers following the transition mechanism.

We do the above three steps in order until $P(t+4)$ meets the power cap, $\bar{P} + z(t)R$.

C. Regulation Reserves

Now we estimate the average power consumption $\bar{P}$ and the regulation reserve R that the data center should bid in the power market for the next hour. We assume the arrival of the workloads is a Poisson process with an arrival rate λ (per hour). The value of λ can be controlled by allocating overall load among geographically dispersed data centers to exploit spatiotemporal variations in energy prices [29]. The λ considered here is the one after such allocation. Each job j is composed of a number of instructions, namely, I_j. Since we have homogeneous workloads, all $I_j, j = 1, 2...$ are equal and denoted as I. Finishing a job is equivalent to executing all the instructions.

Having λ, I and the coefficient k between P_{dyn} and RIPS, we are able to estimate the total dynamic energy needed during the hour, E_{dyn}, for finishing all workloads, which is $E_{dyn} = \lambda * kI$. Only active servers can consume dynamic power. As mentioned before, each server has the dynamic power range in (0, $P_{dyn,max}$]. However, our designed policy always tries to force active servers to run at peak capacity, and as a result, most of the active servers consume at the maximal dynamic power. Hence in order to provide sufficient dynamic energy E_{dyn} for serving all workloads in the hour, the average number of servers that should be active, $\bar{N}_{active}$, is:

$$\bar{N}_{active} = \frac{\int_0^{1h} N_{active}(t)dt}{1h} = \frac{E_{dyn}}{P_{dyn,max} * 1h} = \frac{\lambda * kI}{P_{dyn,max} * 1h} \tag{8}$$

While estimating the average power consumption $\bar{P}$ during the hour, the energy waste during transition periods needs to be considered. As introduced before, each resuming process has an energy loss as E_{loss}. We assume the total number of times the servers are resumed in the hour across the data center is N_{res}. Then, the total energy waste during the hour is: $E_{loss,1h} = E_{loss} * N_{res}$.

Next, we estimate the number of times the servers are resumed, N_{res}. As introduced before, the dynamic range of RS signal $z(t)$ is [-1,1]. At $z(t) = -1$, the data center is at the lowest power consumption, P_{low}, and at $z(t) = 1$, the data center is at the highest power consumption, P_{high}. In order to increase the regulation reserve R to gain more monetary savings, we should minimize P_{low} and maximize P_{high}. The minimal P_{low} we can achieve is by setting all servers in sleep, and the maximal P_{high} is by setting all servers actively running at the peak power. Thus, every time RS signal increases from -1 to 1, almost all the servers need to be resumed. On the other hand, our designed policy avoids the situation that a server is resumed and suspended back and forth when tracking minor changes in the RS signal. Therefore, in our policy, resuming servers only happens during the periods of the RS signal with large increases. We denote the number of RS signal periods with large increases during the hour as p_b. Then we have $N_{res} = p_b * N_{dc}$. A good estimate[4] of p_b is $p_b = 2$.

Now we can estimate the average power consumption $\bar{P}$ by the following equations:

$$\bar{P} = \frac{\int_0^{1h}(\bar{P} + Rz(t))dt}{1h} = \bar{N}_{active} * P_{active} + \bar{N}_{idle} * P_{idle} + \bar{N}_{sleep} * P_{sleep} + \frac{E_{loss,1h}}{1h} \quad (9)$$

and

$$N_{dc} = \bar{N}_{active} + \bar{N}_{idle} + \bar{N}_{sleep} \quad (10)$$

We have solved $\bar{N}_{active}$ and $E_{loss,1h}$ before. P_{idle} and P_{sleep} are constants and known. In our policy, most active servers run at peak capacity; hence, in the equation we can simply replace P_{active} with P_{peak}.

Due to the transition mechanism introduced before, $\bar{N}_{idle}$ is not 0. Moreover, for reducing the performance degradation caused by time delays for resuming the sleep servers, we can manually set aside some idle servers as the "performance guarantee slack". These idle servers are prepared for immediately serving coming jobs. Hence we are able to manually tune the $\bar{N}_{idle}$, and as a result, $\bar{P}$ is changed, i.e., $\bar{P}$ is a function of $\bar{N}_{idle}$. In our work, each sleeping server is coupled with an idle server for providing QoS guarantees; i.e., we use $\bar{N}_{idle} = \bar{N}_{sleep}$ in calculating $\bar{P}$.

Next, we estimate the regulation reserve R that we should bid. First, we have the constraints as:

$$\begin{aligned} \bar{P} - Rz(t) &\geq N_{dc}P_{sleep}\ , \\ \bar{P} + Rz(t) &\leq N_{dc}P_{peak},\ \forall t \end{aligned} \quad (11)$$

As we know $z(t) \in [-1, 1]$, we have:

$$R \leq min\{N_{dc}P_{peak} - \bar{P},\ \bar{P} - N_{dc}P_{sleep}\} \quad (12)$$

[4]This estimate is based on our observations. In fact, the experiments indicate that accuracy of p_b estimation is not critical.

Prior results on single server regulation show that the value of R does not notably affect the tracking performance or the QoS degradation [5]. Moreover, the results of the single server experiments show that the optimal R is indeed almost equal to $min\{\bar{P} - P_{idle},\ P_{max} - \bar{P}\}$. Considering data centers provide even more flexibilities in providing RS compared to a single server, we assume a similar result for the data center; i.e., the RS reserve value R reaches the bound in Eq. (12).

IV. EXPERIMENTAL RESULTS

In this section, we first introduce the system simulation methodology, and then evaluate the bidding estimation and power control policy proposed in Section III under various data center scenarios.

A. Methodology

To determine the relationship between P_{dyn} and RIPS, we run each application from the PARSEC-2.1 [7] benchmark suite on a 1U server, which has an AMD Magny Cours (Opteron 6172) processor, with 12 cores on a single chip. The server is virtualized by VMware vSphere 5.1 ESXi hypervisor; hence, we are able to use the CPU resource limits knob to control the power-performance settings. Detailed results on P_{dyn} and RIPS are shown in our prior work [5].

We assume jobs arrive at the data center following a Poisson process. We generate the workload sequences using Monte Carlo simulation [5]. Without loss of generality, we assume a data center server cluster with $N_{dc} = 100$ servers. By default the data center utilization is 50%. We simulate a 1-hour period experiment 10 times and evaluate the power tracking, QoS performance, and the monetary cost. Regarding the sleep state, we assume $t_{res} = 10$s and $P_{sleep} = 10\%P_{peak}$.

B. Data Center RS

We next evaluate the data center level RS tracking performance and QoS degradation, as well as the monetary savings. Then we compare the results to the single server RS proposed in previous work [5, 6].

Fig. 1(a) shows the statistical distribution of the power tracking error, $\epsilon(t) = (P(t+4) - (\bar{P} + z(t)R))/R$, over time t. The result shows that in most of the time, the tracking errors are close to 0 for the data center level RS, while for the single server the tracking errors are mostly around 0.1-0.2. Moreover, the data center RS has smaller deviation in the tracking error. The maximal tracking error of the data center is less than 1 while that of the single server case reaches close to 2.5. Some ISOs have strict limitations on the peak tracking error, thus the data center can perform much better than a single server.

Fig. 1(d) shows the statistical distribution of job servicing time degradation, D_i, for each job i. This degradation is the ratio of the job servicing time T_i when providing RS to the shortest processing time for the job, $T_{i,min}$, which refers to running the job without any power capping restrictions and without any waiting time in the queue. Thus, $D_i = T_i/T_{i,min} - 1$, and $D_i = 0$ means that there is no degradation. Our result shows that most jobs get almost no degradation in data center RS, while for the single server, the jobs suffer degradation with higher probabilities.

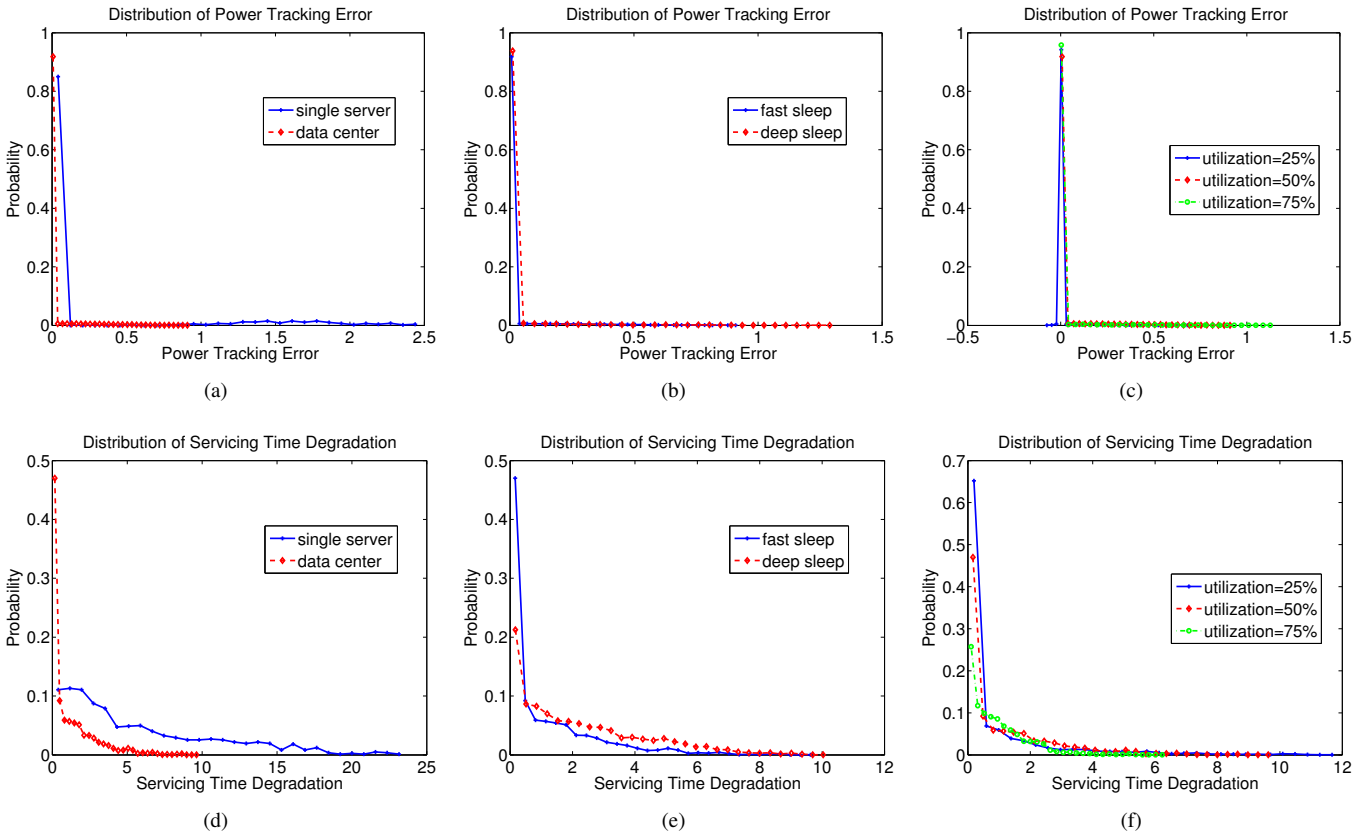

Fig. 1. Statistical distribution of tracking error (a, b, c) and job servicing time degradation (d, e, f). (a) and (d) are results of a single server and the data center both at 50% utilization. (b) and (e) are results of the data center at 50% utilization with a "fast sleep" server state and a "deep sleep" server state. (c) and (f) are results of the data center at various utilizations. All cases are tested with a (homogeneous) set of Blackscholes jobs.

Then, we check the monetary savings in both cases. The net cost of the power for providing RS is $\Pi_E \bar{P} - \Pi_R R$, with $\Pi_E \approx \Pi_R$ [5]. Therefore, $R/\bar{P}$ represents the percentage of the monetary savings. For the single server, we have optimal $R/\bar{P} = 29.7\%$, while in the data center it is 56.8%, which is around 2X improvement.

Thus, providing RS brings dramatic monetary savings (56.8%) to data centers or multi-server clusters, with zero power tracking error for most of the time, and no QoS degradation for most of the jobs. Compared to the single server RS, both QoS and the monetary savings are significantly improved. These results are expected as the multi-server clusters can provide more flexibility and control opportunities to perform power regulation.

C. Fast Sleep and Deep Sleep

Unlike desktops and laptops, in today's data centers, many servers do not have sleep states with fast transitions [15]. These servers are usually put in a deeper sleep mode and rebooted if needed. The time delay and the energy loss in the rebooting process are larger than those of resuming process from the sleep state, whereas servers can save more energy in the deep sleep state compared to the fast sleep state. Based on recent work [15], we assume the time delay of the rebooting process is $t_{reb} = 200$ seconds, and the power consumption in the deep sleep state is $P_{deepS} = 5\% P_{peak}$. The power consumption in the transition process, P_{tran}, is close to P_{peak}. We conduct an experiment with the assumption that the servers in the data center have deep sleep and rebooting instead of fast sleep states.

Fig. 1(b) shows the statistical distribution of power tracking error of the two cases (fast sleep and deep sleep). The power tracking is accurate for both cases. This is because our policy puts power tracking as the highest priority, and unlike the server state, the power consumption rate during the transition process can be immediately changed without any delays, from P_{sleep} or P_{deepS} to P_{tran}, or vice versa. Hence the power tracking accuracy is not sensitive to whether the servers have fast sleep states or not. In addition, the figure shows that using slower deep sleep states leads to larger peak tracking errors.

Fig. 1(e) shows the statistical distribution of job servicing time degradation of two cases. The result shows that data center servers with fast sleep states have smaller degradation and better QoS. This is because the time delay of rebooting a server is very large, which strongly affects the job servicing performance. Overall, even though data centers without fast sleep states have more degradation, most of the degradation is still small, and the QoS is high.

For a data center with faster sleep states, we have monetary savings around $R/\bar{P} = 56.8\%$, while for the slower deep sleep states, the savings are only 36.9%, for the reason that the faster sleep state provides the ability to react more rapidly to ISO requests compared to rebooting. Thus, power RS in both cases can bring significant monetary savings with close to zero power tracking error for most of the time and small QoS degradation for most of the jobs, while having a fast sleep state further improves the monetary savings and QoS.

D. Impact of Cluster Utilization

By default we assume the utilization of the data center is 50%. In real life, different clusters, or same clusters at different time, have different utilizations. Next, we evaluate the impact of cluster utilization in providing RS. Fig. 1(c) shows the statistical distribution of power tracking error under various utilizations. Tracking performance in various utilizations is similar, and most of the tracking errors are close to zero. This is because in all cases our policy gives the highest priority to power tracking.

Fig. 1(f) shows the statistical distribution of job servicing time degradation under various utilizations. The result shows that when utilization increases, the performance degradation increases. This is expected, as more jobs need to be processed under higher utilization and more servers are busy, which increases the performance degradation. Overall, the degradation of all three cases is mostly close to 0; hence, the data center can provide RS reserve without significantly influencing the job QoS in various utilization levels.

We compare the monetary savings of different utilization cases. For $U = 25\%, 50\%, 75\%$, we have savings $R/\bar{P} = 78.0\%, 56.8\%, 21.8\%$ correspondingly, which shows that the savings decrease when the utilization increases. This is due to the reason that higher utilization leads to higher average power consumption $\bar{P}$, which limits R. However, even with 75% utilization, we still can have around 22% monetary savings, which shows that providing RS on the data center has cost advantages regardless of the utilization.

E. Impact of Different Workloads

All previous experiments are conducted by using workloads made out of homogeneous Blackscholes jobs. In this part we study the data center RS problem with different types of workloads. Table I shows the experimental results on four different workloads. We list their power tracking statistics, QoS degradation statistics, and monetary savings. $\bar{D}$ and σ_D are the mean and standard deviation of performance degradation, $\bar{\epsilon}$ and σ_ϵ are the mean and standard deviation of the tracking error. The results show that the power tracking performance is not influenced by the workload type, while the performance degradation is. From the table, workloads with longer shortest processing time, i.e., $T_{i,min}$, such as Streamcluster and Facesim (whose shortest processing time is larger than 100 seconds, while Blackscholes and Canneal only have 20-40 seconds), have less QoS performance degradation. This is expected as the waiting time is relatively shorter (compared to the processing time) for longer processing time jobs. As our policy has rules (e.g., P_{min}) to guarantee the job processing time, waiting time becomes the main reason for degradation. Overall, both the performance degradation and the tracking error are quite small. In addition, in all cases, data centers can achieve more than 50% monetary savings. Hence data center level RS is expected to provide small tracking errors and QoS degradation along with dramatic monetary savings for a broad range of workloads.

TABLE I
CLUSTER LEVEL POWER REGULATION ON DIFFERENT WORKLOADS

	Blackscholes	Canneal	Streamcluster	Facesim
$\bar{P}$	$9.75 * 10^3$	$9.71 * 10^3$	$9.84 * 10^3$	$9.84 * 10^3$
R	$5.54 * 10^3$	$4.98 * 10^3$	$5.46 * 10^3$	$5.11 * 10^3$
$\bar{D}$	1.13	1.13	0.21	0.22
σ_D	1.54	0.69	0.26	0.27
$\bar{\epsilon}$	0.03	0.03	0.03	0.03
σ_ϵ	0.10	0.09	0.09	0.09
$R/\bar{P}$	56.8%	51.3%	55.5%	52.0%

[a] $\bar{D}$ and σ_D are mean and standard deviation of performance degradation; $\bar{\epsilon}$ and σ_ϵ are mean and standard deviation of tracking error.

V. RELATED WORK

Some previous work has investigated demand side RS in power market. Caramanis et al. [4] study the RS bidding problem by using optimal dynamic pricing policies. Paschalidis et al. [24] propose a market-based mechanism to enable the smart building to provide RS.

Data center level power management techniques have advanced significantly in the recent years. For the server level power management, DVFS, power gating and multi-core processor workload scheduling and allocation have been investigated [17, 27, 28]. Different power capping techniques, which are used for meeting the peak or average power constraints have also been widely studied [9, 8, 26, 18]. In virtualized servers, some CPU resource management and consolidation techniques have been applied to manage the power consumption [22, 14, 13]. On the data center level, application and server aware power budgeting have been researched [25, 21]. Zhan et al. [31] propose a system profile based energy-efficient data center power budgeting technique. Gandhi et al. [10] investigate the optimal power allocation in server farm by considering different complex situations. Server commitment is another hot topic in the data center level power management. Meisner et al. [20] propose PowerNap technique to eliminate the server idle power. Isci et al. [15] explore the feasibility of low-latency power states and demonstrate a power-aware virtualization management solution leveraging these states. Gandhi et al. [11] study the regime of sleep states that would be advantageous in data centers and propose some dynamic power management policy based on server commitment.

There are a few recent studies that investigated on data center participation in advanced power market. Ghamkhari et al. [12] build an analytical profit model to determine whether participation in an ancillary service market can be beneficial to data centers. Aikema et al. [3] analyze a number of different advanced power market for data centers to participate in potential. Wang et al. [30] propose to migrate the workload between geographically distributed and virtualized data centers situated in multiple regional electrical markets, to maximize the expected payoff. However, none of these work closely consider using data center power management techniques and designing servicing policy for providing RS. Chen et al. [6, 5] propose a data center level power management framework to provide RS, but then the work only focuses on a single server level power management.

Our work is the first to closely investigate the data center level power budgeting and management, the server commit-

ment, as well as the workload scheduling and allocation, to enable the data center to participate in the advance power market. We propose a dynamic power control policy for the data center to provide RS, to achieve dramatic monetary savings while also guarantee no major deterioration in QoS.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a power control policy for data centers to dynamically track RS signal related power capping. We have also introduced an estimation method to calculate the RS provision bidding value. Experimental results show that data centers with our policy and estimation can accurately track the RS signal and achieve more than 50% energy monetary savings, with no major QoS performance degradation, regardless of types of workloads. The results also demonstrate the strong capacity and substantial monetary savings of data centers to provide RS in various scenarios, e.g., under different utilizations and with different server states. In addition, the significant improvement in both monetary savings and QoS of data center level RS provision has been investigated and compared to prior single server results, indicating that data-center-wide control is not only feasible but also more beneficial. Our ongoing work focuses on (i) leveraging heterogeneous workload and server RS provision by advanced power budgeting and job scheduling, and (ii) considering synergies with cooling power consumption.

ACKNOWLEDGMENTS

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

REFERENCES

[1] PJM (2013). market-based regulation [online]. *http://pjm.com/markets-and-operations/ancillary-services/mkt -basedregulation.aspx*.

[2] Manual 2: Ancillary services manual, v3.26. *NYISO*, 2013.

[3] D. Aikema, R. Simmonds, and H. Zareipour. Data centres in the ancillary services market. In *Green Computing Conference (IGCC), 2012 International*, pages 1–10. IEEE, 2012.

[4] M. C. Caramanis, I. C. Paschalidis, C. G. Cassandras, E. Bilgin, and E. Ntakou. Provision of regulation service reserves by flexible distributed loads. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 3694–3700. IEEE, 2012.

[5] H. Chen, A. K. Coskun, and M. C. Caramanis. Real-time power control of data centers for providing regulation service. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, 2013.

[6] H. Chen, C. Hankendi, M. C. Caramanis, and A. K. Coskun. Dynamic server power capping for enabling data center participation in power markets. In *Intl. Conf. on Computer-Aided Design*, 2013.

[7] B. Christian. Benchmarking modern multiprocessors. *Ph.D.Thesis. Princeton University*, 2011.

[8] R. Cochran, C. Hankendi, A. K. Coskun, and S. Reda. Pack & cap: adaptive dvfs and thread packing under power caps. In *Proceedings of the 44th annual IEEE/ACM international symposium on microarchitecture*, pages 175–185. ACM, 2011.

[9] H. David, E. Gorbatov, U. R. Hanebutte, R. Khanna, and C. Le. Rapl: memory power estimation and capping. In *Low-Power Electronics and Design (ISLPED), 2010 ACM/IEEE International Symposium on*, pages 189–194. IEEE, 2010.

[10] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy. Optimal power allocation in server farms. In *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, pages 157–168. ACM, 2009.

[11] A. Gandhi, M. Harchol-Balter, and M. A. Kozuch. Are sleep states effective in data centers? In *Green Computing Conference (IGCC), 2012 International*, pages 1–10. IEEE, 2012.

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

[13] C. Hankendi, S. Reda, and A. K. Coskun. vcap: Adaptive power capping for virtualized servers. In *Low Power Electronics and Design (ISLPED), 2013 IEEE International Symposium on*, pages 415–420. IEEE, 2013.

[14] I. Hwang, T. Kam, and M. Pedram. A study of the effectiveness of cpu consolidation in a virtualized multi-core server system. In *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*, pages 339–344. ACM, 2012.

[15] C. Isci, S. McIntosh, J. Kephart, et al. Agile, efficient virtualization power management with low-latency server power states. In *Proceedings of the 40th Annual International Symposium on Computer Architecture*, pages 96–107. ACM, 2013.

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

[17] J. Li and J. F. Martinez. Dynamic power-performance adaptation of parallel computation on chip multiprocessors. In *High-Performance Computer Architecture, 2006. The Twelfth International Symposium on*, pages 77–87. IEEE, 2006.

[18] K. Ma and X. Wang. Pgcapping: exploiting power gating for power capping and core lifetime balancing in cmps. In *Proceedings of the 21st international conference on Parallel architectures and compilation techniques*, pages 13–22. ACM, 2012.

[19] Y. V. Makarov, C. Loutan, J. Ma, and P. de Mello. Operational impacts of wind generation on california power systems. *Power Systems, IEEE Transactions on*, 24(2):1039–1050, 2009.

[20] D. Meisner, B. T. Gold, and T. F. Wenisch. Powernap: eliminating server idle power. In *Sigplan Notices*, volume 44, pages 205–216. ACM, 2009.

[21] R. Nathuji, C. Isci, E. Gorbatov, and K. Schwan. Providing platform heterogeneity-awareness for data center power management. *Cluster Computing*, 11(3):259–271, 2008.

[22] R. Nathuji, K. Schwan, A. Somani, and Y. Joshi. Vpm tokens: virtual machine-aware power budgeting in datacenters. *Cluster computing*, 12(2):189–203, 2009.

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

[24] I. C. Paschalidis, B. Li, and M. C. Caramanis. A market-based mechanism for providing demand-side regulation service reserves. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 21–26. IEEE, 2011.

[25] K. Rajamani, H. Hanson, J. Rubio, S. Ghiasi, and F. Rawson. Application-aware power management. In *Workload Characterization, 2006 IEEE International Symposium on*, pages 39–48. IEEE, 2006.

[26] K. K. Rangan, G.-Y. Wei, and D. Brooks. Thread motion: fine-grained power management for multi-core systems. In *ACM SIGARCH Computer Architecture News*, volume 37, pages 302–313. ACM, 2009.

[27] K. Singh, M. Bhadauria, and S. A. McKee. Real time power estimation and thread scheduling via performance counters. *ACM SIGARCH Computer Architecture News*, 37(2):46–55, 2009.

[28] R. Teodorescu and J. Torrellas. Variation-aware application scheduling and power management for chip multiprocessors. *ACM SIGARCH Computer Architecture News*, 36(3):363–374, 2008.

[29] H. Wang, J. Huang, X. Lin, and H. Mohsenian-Rad. Exploring smart grid and data center interactions for electric power load balancing. In *SIGMETRICS, 2013 ACM Conference on*, 2013.

[30] R. Wang, N. Kandasamy, C. Nwankpa, and D. R. Kaeli. Data centers as controllable load resources in the electricity market. In *Intl. Conf. on Distributed Computing Systems*, 2013.

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