Reducing the Data Center Electricity Costs Through Participation in Smart Grid Programs

Hao Chen
Electrical and Computer Engineering
Boston University
Email: haoc@bu.edu

Michael C. Caramanis
Systems Engineering
Boston University
Email: mcaraman@bu.edu

Ayse K. Coskun
Electrical and Computer Engineering
Boston University
Email: acoskun@bu.edu

***Abstract*—To accommodate the increasing presence of intermittent renewable energy sources in power generation, electricity providers offer incentives for *demand response* so as to stabilize the grid load. This paper argues that data centers have the ability to provide large capacity reserves to emerging smart grid programs, and thus, provide opportunities for sustainable data center growth and cost savings as well as more flexibility for the grid. The paper first reviews the emerging smart grid opportunities, and then proposes policies to deliver data center demand response for peak shaving, regulation services, and frequency control prorgrams. Experimental results indicate substantial cost saving potential compared to solely applying energy management strategies at the data centers.**

I. INTRODUCTION

A 2007 EPA report to Congress [5] stated that the data centers in the US consumed about 1.5% of the US electricity production and projected that this would rise to 3% in 2011. To put this in context, 3% of the US electricity production is about 120 billion kWh or equivalent to the average consumption of a large city with 11.6 million households. A more recent 2013 report [37] puts the *Information and communications Technologies (IT) ecosystem* (including all computing, data centers, internet infrastructure, mobile devices, cellular networks, etc.) consuming about 10% of the world's electricity generation. The IT ecosystem is expected to continue to grow, especially considering the wider-spread adoption of mobile and cloud computing paradigms.

Currently, the vast majority of electricity production comes from fossil fuels, which is long-term unsustainable and has a tremendous environmental impact. Renewable sources amounted to about 12% of the US electricity production in 2012 (based on data from the US Energy Information Administration). Globally, the figures are not much better with the exception of a few European countries. One of the reasons for limited renewable adoption is related to the challenge in real-time matching of supply and demand in the grid. This already significant challenge is tremendously growing because of the intermittent nature of renewable energy generation and the lack of large-scale, green energy storage solutions. As a consequence, the providers have started to offer incentives for the *demand side*, or the electricity consumers, to provide reserves that can be modulated at the request of the providers.

We believe it would be highly appealing if the IT sector, whose growth is contributing to increased electricity demand, could emerge as a major enabler of substantial electricity generation from renewables. This would effectively make the growth of the IT sector sustainable and environmentally neutral, or even beneficial. We envision that the way to achieve this goal is through the participation of primarily data centers, and more broadly computing systems, into the emerging smart grid demand response programs.

In this context, this paper first reviews the energy management mechanisms in data centers as well as the programs and strategies emerging in the smart grid. We then propose policies that enable a data center to regulate its power according to the needs of the program it is participating in. These policies leverage analysis of workload behavior and the available hardware-software control knobs in data centers, and optimize the modulation of these control knobs to achieve the dynamic power consumption targets. We base our evaluation on real-life power and performance data collected from servers, projected to cluster level. Our results demonstrate that participating in smart grid demand response programs, i.e., regulation services and frequency control, can reduce data center electricity costs by up to 59.9% and 68.3%, respectively, in a typical scenario of 50% utilization, while meeting the service level agreements (SLAs) for quality of service (QoS).

II. BACKGROUND AND RELATED WORK

This section first outlines the related work on data center power management techniques. We then introduce the emerging smart grid demand response programs and strategies that are suitable for the data center to participate.

A. Data Center Power Management and Energy Efficiency

Improving energy efficiency helps the data centers reduce their operational costs. Power management and energy efficiency at the server / processor level have been studied broadly. As the number and size of data centers grow throughout the world, especially following the wider adoption of cloud services, a number of researchers start focusing on power management and energy efficiency at the data center level.

A.1. Server / Processor Level Power Management

The majority of the processors today are designed to support various energy-aware operation settings [10]. Widely used control knobs include dynamic voltage-frequency scaling (DVFS) and power gating features to turn off idle units [31]. Multi-core processors offer additional degrees of freedom for managing power through workload allocation [45]. Recently, voltage and frequency islands (VFIs) have been introduced for achieving fine-grained system level power management [41].

Dynamic power management (DPM) at the processor level typically focuses on designing efficient techniques to put idle units into sleep states while minimizing the performance overhead from switching between states [8]. PowerNap is a similar approach at the server level for eliminating the server idle power and reducing the state transition overhead [35]. Isci et al. [29] explore the feasibility of low-latency power states implemented at the server hardware and introduce a power-aware virtualization management policy.

Today's systems also employ power capping mechanisms to prevent the power from exceeding the peak power constraints. DVFS is a popular control knob for capping [20]. For multi-threaded applications, DVFS can be combined with thread allocation and migration to perform finer granularity power capping [17], [44].

As the virtualization technique has advanced significantly in recent years and provides advantages in ease of management and consolidation, a class of power management techniques specifically address virtualized servers. vGreen tries to improve energy efficiency of virtualized servers by linking workload characterization to dynamic virtual machine (VM) scheduling [19]. Some work studies the power management effectiveness of CPU consolidation on virtualized system [28]. Turning CPU resource limits is a recently introduced power management control knob on virtualized server that can achieve finer granularity power consumption compared to DVFS [27].

A.2. Data Center Level Power Management

A data center consists of many servers. In addition to the power management capabilities available within the servers, a data center offers other power management knobs, including power budgeting, job scheduling, and server provisioning.

Some power budgeting approaches consider the heterogenous set of applications and divide total power caps based on application properties [43], [49]. Gandhi et al. [21] develop a queueing model and produce theorems that determine the optimal power allocation under different scenarios including different arrival rates of jobs, power-to-frequency relationships in the processors, etc. The power budgeting problem has also been studied on virtualized systems [39], [40].

Job scheduling impacts the power and performance of data centers and, therefore, has been extensively studied. First-in first-out (FIFO) policy is a widely used strategy today because of its simplicity and fairness. Back-filling is another popular strategy aiming to improve system utilization [38].

Server provisioning, which decides how many servers should be active at a given time, is another essential topic in the data center. Many data centers today leave all the unused servers in idle states as a conservative approach for guaranteeing high performance. Leaving many servers idle, however, causes tremendous waste of energy. Some data center researchers leverage sleep states to improve energy efficiency [11], [36]; however, they typically ignore the wake-up costs from sleep states or use hypothetical server states. Gandhi et al. [22] propose a *SoftReactive* dynamic power management policy, which determines the state of servers in the data center based on the dynamic workload arrival rate, and introduce a timeout-based mechanism to sleep servers.

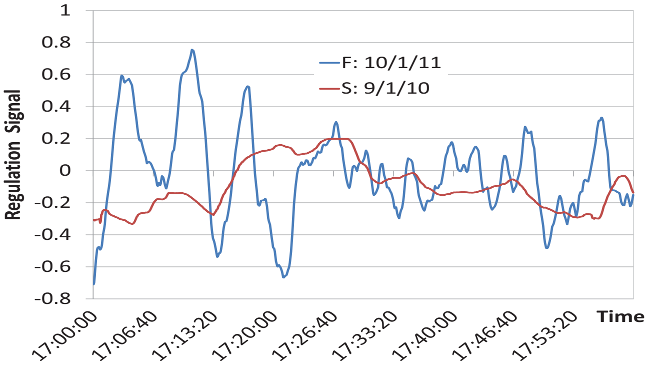


Fig. 1. Typical PJM 150sec ramp rate (F) and 300sec ramp rate (S) regulation signal trajectories.

B. Data Center Demand Response Programs and Strategies

While data center energy costs can be reined in by improvements in energy efficiency, *demand response* to spatio-temporally varying system costs and capacity reserve needs is emerging as an even more effective money saver. We review below both legacy demand response programs as well as new demand response programs that are emerging from the participation of loads in power markets on par with generation side participants. Legacy programs include peak shaving, interruptible load contracts, load shedding and shifting across time, etc. New power market related demand response includes the primary, secondary and tertiary capacity reserves by loads.

B.1. Peak Shaving

Most legacy electricity rates for medium to large commercial and industrial power consumers are binary rates including capacity (i.e., the maximal consumption rate recorded over an agreed upon period) and energy charges. Assuming coincident system and individual consumer demand, electric utilities have relied on a high penalty levied on customer peak kW demand to lower system peak [26]. Data centers are often under Coincident Peak Pricing (CPP) rates that charge a very high cost for usage during the hour that is coincident to the system peak hour [18]. Data Center power capping techniques have been clearly motivated by CPP type rates. In fact, power capping as well as DVFS are both common peak shaving tools. Delaying or speeding up job processing or transferring computational load to geographically distant data centers, to the extent allowed by QoS agreements and available capacity, provide additional opportunities for peak shaving. Finally, storage options involving data center UPS facilities present degrees of freedom for shifting power consumption from on-peak to off-peak periods [26], [7].

B.2. Capacity Reserves

Load side provision of capacity reserves in cascaded day-ahead, hour ahead and five minute markets [42], [4] is picking up traction, with PJM, the largest US Independent System Operator (ISO), pursing it since 2006 [3]. We consider all three types of power capacity reserves under the following notation: primary or frequency control (FC) reserves, R_1, secondary or regulation service (RS) reserves, R_2 and tertiary or operating reserves, R_3. Providers are obliged to modulate their power consumption so as to track a stochastic non-anticipatory dynamic power target, $P_i^{target}(t)$ for $i \in 1, 2, 3$. For primary and secondary reserves, the target varies symmetrically about a fixed average power level $\bar{P}_i$ allowing energy neutral time averaged consumption. Although $P_i^{target}(t)$ dynamics are stochastic and are revealed to reserve providers in short notice, their statistical behavior is well known.

Primary or Frequency Control (FC) Reserves

A primary reserve provider that has offered R_1 in the hour ahead market, must modulate its power consumption $P_1(t)$ in real time to track a target $P_1^{target}(t)$ that is determined as a function of the local (and hence fully distributed) frequency measurement $\omega(t)$. Denoting frequency deviations from 60Hz by $\Delta\omega(t) = \omega(t) - 60$, we have:

$$P_1^{target}(t) = \begin{cases} \bar{P}_1 - R_1, & \Delta\omega(t) \leq -0.2 \\ \bar{P}_1 + \frac{(\Delta\omega(t)+0.02)}{0.2-0.02} R_1, & -0.2 < \Delta\omega(t) < -0.02 \\ \bar{P}_1, & |\Delta\omega(t)| \leq 0.02 \\ \bar{P}_1 + \frac{(\Delta\omega(t)-0.02)}{0.2-0.02} R_1, & 0.02 < \Delta\omega(t) < 0.2 \\ \bar{P}_1 + R_1 & \Delta\omega(t) \geq 0.2 \end{cases}$$

$P_1^{target}(t)$ is a piecewise linear function of $\Delta\omega(t)$, representing the local impact of system-wide supply-demand imbalances. Under most circumstances the statistical behavior of $\Delta\omega(t)$ constitutes a zero mean white noise, whose variance is well known at the beginning of the hour.

In FC, $P_1^{target}(t)$ varies in real time according to $\Delta\dot{\omega}(t)$. We approximate the real time dynamics of $\omega(t)$ by discrete time dynamics with a small time increment of 0.1 seconds. $\Delta\dot{\omega}(t)$ is generally unconstrained, but $\dot{P}_1(t)$ is of constant magnitude. More precisely, $\dot{P}_1(t) = SGN(P_1^{target}(t) - P_1(t))R_1/30$ MW/sec. When $P_1(t) = P_1^{target}(t)$ then and only then $\dot{P}_1(t) = 0$. As such, there is no tracking error allowed in FC reserve power output modulation, as is instead the case with secondary reserves. Although primary reserves are not yet cleared in power markets, and are in fact provided by centralized generation facilities [50], in anticipation of markets evolving in this direction, we assume for purposes of obtaining a reasonable estimate of primary reserve clearing prices, Π_{R1}, that Π_{R_1} equals several times of the value of energy clearing prices Π_E. This is a reasonable assumption given that primary reserves are more valuable than secondary reserves where we assume that Π_{R_2} is of the same order of magnitude as Π_E. In the numerical results reported below we use the relationship $\Pi_E\bar{P}_1 - \Pi_{R_1}R_1$ to evaluate the effective energy cost of a data center that offers FC reserves R_1.

Secondary or Regulation Service (RS) Reserves

A significant difference compared to primary reserves of regulation service (RS) is that each provider is obligated to track the same relative target determined generally by an ISO signal that we denote by $z(t)$. In fact, $z(t)$ is the output of a proportional-integral filter of system wide frequency deviation and Area Control Error (ACE), determined carefully in order to complement primary control reserves. Figure 1 depicts actual historical data trajectories of $z(t)$ corresponding to two different normalized speeds of 1/150 and 1/300 MW/sec [4]. Note that energy neutrality renders $z(t)$ a zero mean random variable over the set $[-1, +1]$. $z(t)$ is centrally determined and broadcasted at 4 second intervals by the ISO. Although $z(t)$ is not known to RS reserve providers in advance, it follows a well behaved two level Markov model whose transition probabilities can be usually calibrated a few hours in advance.

An RS provider who has offered R_2 in the hour ahead market is obliged to modulate its power consumption $P_2(t)$ to track the target $P_2^{target}(t) = \bar{P}_2 + z(t)R_2$ at a constant, albeit slower, speed than FC. With energy and reserve market clearing prices, Π_E and Π_{R2}, that are of the same order of magnitude, a RS reserve provider sees an effective energy cost of $\Pi_E\bar{P}_2 - \Pi_{R2}R_2$. The credit received may be further reduced as a function of the tracking error. A more detailed discussion of data center provided RS reserves can be found in our recent work published in [13], [12].

Recent studies on data canter RS provision include Aikema et al. [6], where multiple ancillary service provision programs, including RS, by data centers are considered. Brocanelli et al. [9] propose to jointly leverage a data center and its employees' PHEVs to provide RS. Our previous work [14], [12] considers a real-life data center model and examines power management techniques that enable the provision of RS.

Tertiary or Operating Reserves

Concluding with tertiary reserve provision we note that it is scheduled in the 5 minute power markets. It involves rescheduling of the provider's consumption from a pre-contingency or pre-congestion level $\bar{P}_3$ to a post contingency or post congestion level that is as much as R_3 lower. That lower level must then be maintained for up to a few hours. Tertiary reserves provide the opportunity to computing loads spread across distant geographic areas to re-route jobs between them. The speed at which tertiary reserves must be offered is far slower than that of primary or secondary reserves, which is $R_3/900sec$, compared to $R_1/30sec$ and $R_2/150sec$. While most flexible loads such as EVs are capable of inter-temporal rescheduling, geographically dispersed but coordinated computing loads, such as data centers with excess pooled capacity, are uniquely qualified for geographic rescheduling. Benefits from reduced computing load cost, as well as power network congestion costs can be significant. For example, PJM locational marginal prices on July 19, 2013, 4:05pm, range from \$30 to over \$500/MWh [1]. This is a case in point of the effectiveness of data transfers over fiber optic lines undertaken in order to compensate for electricity transmission line congestion. Tertiary reserves are often operated with *load migration* discussed further below.

Participation of Data Center in Smart Grid Reserves

It is the aforementioned role of providing capacity reserves we envision for data centers, and more broadly, for computing systems. In today's grid, the FC and RS needed to ensure stability by guaranteeing tolerable ACE and frequency deviation errors amounts to about 0.1% for FC and 1% for RS related to the total electricity load. This amount is in fact comparable to the 2-3% figure attributed to electricity consumption by data centers [30]. Tertiary reserves can be offered effectively by geographically distributed, yet collaborating, data centers relying on excess capacity auctions as those introduced recently by Amazon [2].

Considering today's market conditions, secondary reserves are traded at a price comparable to the price of energy, while, if primary reserves are introduced into power markets, they will probably command higher clearing prices. This implies that a data center able to provide RS reserves equal to 50%, or alternatively FC reserves equal to 10%, of its average energy consumption, may be able to reduce its energy cost by up to 50%. There are, therefore, substantial increasing economic incentives for data center operators to participate in jointly clearing energy and reserve markets. The associated societal and sustainability benefits that may result from greater adoption of renewables enabled by effective data center reserve provision are clearly enormous.

B.3. Load Shedding, Shifting and Migration

Load shedding, shifting and migration constitute data center strategies applicable to various demand response programs, e.g., peak shaving and interruptible contracts. Load shedding refers to a temporary load reduction without a future pay back. As such, load shedding is usually accompanied with QoS degradation. Load shifting, on the other hand, refers to temporary idling or shutting down of servers associated with shifting computing tasks to a future time. Generally speaking, all inter-temporal job rescheduling strategies come under load shifting. Load migration refers to geographic shifting of load to another data center or cluster [24].

Lawrence Berkeley National Laboratory (LBNL) investigates various power management techniques, including load shedding and shifting, and studies degrees of freedom in data center power usage as demand response [24]. Liu et al. [34] compare the capacity of data center demand response, mainly load shedding, with large-scale storage in stabilizing the power network. Ghamkhari et al. [23] examine potential data center benefits associated with participation in a voluntary dynamic load reduction program.

Load migration technique has been broadly studied recently. Chiu et al. [15] propose a low-cost geographic load migration solution to balance the grid. Liu et al. [33] explore the benefits of geographical load balancing. Lin et al. [32] propose an online algorithm for geographical load balancing and study the potential environmental benefits of it. Some work formulates and solves the interactions between power grid and data centers in geographic load balancing as a game theory problem [46], [48]. Wang et al. [47] propose to migrate the workload geographically using VM migration techniques.

In this work, we focus mainly on comparing data center energy management methods against peak shaving and capacity reserves. We leave investigation of geographical load modulation methods to future work.

III. MODELS AND METHODOLOGY

In this section, we introduce the data center model and the methodology we use in the experiments for evaluating benefits of the data center participation in different demand response programs and strategies.

A data center consists of hundreds to thousands of servers, with each of them being able to operate at different power states. There are three common states: active, idle, and sleep [12]. The power consumption of an active server can be modulated by power management techniques, such as DVFS or CPU resource limits [27]. We use the CPU resource limits knob in the hypervisor to modulate the dynamic power of active servers in this work. CPU resource limits change the resources allocated to a VM on the server, and as a result, modulate the server dynamic power and the throughput.

To acquire the relation between active server dynamic power and server throughput by using the CPU resource limits, we virtualize a 1U server by VMware vSphere 5.1 ESXi hypervisor. Our server has an AMD Magny Cours (Opteron 6172) processor, with 12 cores on a single chip. We run applications from the PARSEC-2.1 [16] benchmark suite under different power-performance settings by leveraging CPU resource limits. Results show a linear relation between the active server power, P_{active}, and the server throughput, represented by the retired instructions per second (RIPS), as $P_{active} = k * RIPS + P_{idle}$. P_{idle} is the power consumption of the server idle states (when no user application is running) and k is a constant that depends on the type of the workload. Detailed results on server dynamic power and RIPS are shown in our prior work [14], [13].

Compared to the "idle" state, a server in "sleep" state consumes significantly lower power. Sleeping servers, however, cannot immediately serve new jobs, as they need to be waken up first. The wake up time delay, T_{up}, ranges from tens of seconds to several minutes, depending on available sleep modes [29]. The delay of sleeping or turning off a server typically is small and can be ignored. Moreover, during the waking up process, servers usually consume power at the maximal rate, P_{max} [29], which lead to additional waking up energy cost $E_{up} = T_{up} * P_{max}$. In this work, we do not consider *completely* turning off servers because of the large delays in booting. In fact, due to this reason, servers in a real-life data center are rarely completely turned off.

We assume a first-in first-out (FIFO) queue for holding incoming jobs that are not served immediately. Such a queuing model is typically used in data centers that mainly aim at serving high performance computing (HPC) type workloads or research/study oriented workloads at universities and academic institutes. Unlike interactive or transactional workloads (i.e., web request, stock exchange, etc.), these workloads are more flexible and tolerable to servicing delays. Therefore, in our model, some workloads may wait in the queue instead of being immediately serviced once they arrive.

We assume jobs arrive following a Poisson process, which is commonly used in data center workload simulation. Then we generate the job queues using Monte Carlo simulation. For each job j, a random number r_j is generated. r_j is used to determine the job arrival time interval; i.e., $\tau_j = -ln(1 - r_j)/\lambda$, where λ is the job arrival rate. In our model, we assume each server only serves one job a time and do not consider server consolidation. We calculate λ based on our target data center utilization, U, which is the average percentage of servers that are active at each time interval. U is determined by the workload arrival rate λ. We simulate a 1-hour period 10 times and evaluate the performance based on the mean and standard deviation statistics. Without loss of generality, we study homogeneous servers and workloads in this work. In fact, a heterogeneous data center with different types of servers and workloads can be split into homogeneous clusters. Also, many HPC clusters include dedicated, optimized set of servers assigned to specific jobs.

IV. PROPOSED POLICIES FOR DATA CENTER PARTICIPATION IN SMART GRID PROGRAMS

In this section, we first propose a power control policy to minimize the energy consumption while guaranteeing the QoS in a data center. We then introduce policies for data center participation in different demand response programs.

When participating in demand response programs, the data center first sends requests to ISO for participation, daily or hourly. The requests can be a provision of peak power reduction, or a bid of RS, etc. Once the requests are approved, a master node in the data center receives requirements (usually peak power caps or regulation signals) from ISO dynamically. It then calculates the power budget for each server and broadcasts the information. Job scheduling is also performed by the master node. Each server then runs workloads within the given power budget, by leveraging DVFS or CPU resource limits control. The QoS of workloads is sent back to the master node by servers periodically. Based on the QoS feedback, the master node alters the decisions.

We evaluate the job QoS by using a probabilistic SLA constraint on the normalized performance. The normalized performance, D_j, is the ratio of the job servicing time T_j under the proposed policy, to the *shortest possible processing time* on our server, i.e., $T_{j,min}$, for the job j. $T_{j,min}$ refers to the time of running the job j without any power capping restrictions and without any waiting time in the queue. Thus, $D_j = T_j/T_{j,min}$ and $D_j = 1$ means that there is no performance degradation. SLA is defined as (d, η), and the SLA is violated if $Probability\{D_j < d\} < \eta$.

A. Minimizing Energy Consumption

Due to the waking up delay and cost, many data centers simply use an *All-on* policy, which never puts any server to sleep. Gandhi et al.'s [22] *SoftReactive* policy puts servers into sleep state to save energy, if they have been idle longer than a timeout threshold. However, *SoftReactive* does not take the job QoS into account while making decisions. It simply wakes up equal number of servers to the number of arrival jobs at every time interval. In fact if a large QoS degradation is tolerable, then energy may be further saved, and the peak power can be reduced. Differently from prior work, our proposed power control policy, *QoS-feedback*, not only leverages power management and server provisioning, but also takes real-time QoS into account while making decisions.

First, in order to avoid frequent transitions between "idle" and "sleep" state, we implement the *timeout mechanism* similar to the policy in prior work [22]: if a server has been in idle longer than a timeout threshold, i.e., T_{tout}, then it automatically sleeps. A good threshold could be: $T_{tout} = \frac{T_{up} * P_{max}}{P_{idle}}$, recall that T_{up} is the server waking up time, P_{max} is the power consumption during waking up period, which equals to the maximum, and P_{idle} is the server idle power. In addition, in order to maximize the number of sleeping servers to save energy, we sort servers based on their cumulative time in the idle state at each moment t, i.e., $T_{idle}(t)$. Servers with the smallest $T_{idle}(t)$ are activated first if some jobs are required to be served at t. Similarly, servers with the largest $T_{idle}(t)$ are put to sleep first if needed.

Second, since we have a linear relation between the active server power, P_{active}, and the server throughput $RIPS$, as $P_{active} = k * RIPS + P_{idle}$, for the energy saving purpose, when we activate a server, setting it up at maximal throughput to reduce the processing time helps minimize the energy waste caused by P_{idle}. Therefore, we run the active servers at their highest throughput and do not modulate their power, when the goal is to minimize energy consumption. In other words, we use server provisioning as the control knob when minimizing energy with QoS constraints.

The main idea of *QoS-feedback* is to determine the minimal number of servers needed at time t, i.e., $N_{min}(t)$, based on the current length of queue and the overall QoS performance till t. Let us assume at time t, there are $S(t)$ jobs running, $Q(t)$ jobs waiting in the queue, and $F(t)$ jobs that have already been finished. The servicing time of finished jobs, i.e., $T_j,\ j = 1, 2, ...F(t)$, are known. For jobs that are currently running, we estimate the finishing time based on the current throughput $RIPS_j(t),\ j = 1, 2, ...S(t)$. Recall that the QoS constraint is defined as (d, η), we calculate the minimal number of jobs in the queue, i.e., $Q_{SLA}(t)$, that are required to meet their SLAs, so that the overall SLA will be met:

$$Q_{SLA}(t) = \eta(S(t) + F(t) + Q(t)) - S_{SLA}(t) - F_{SLA}(t) \tag{1}$$

where $S_{SLA}(t)$ and $F_{SLA}(t)$ are numbers of running and finished jobs that are expected to meet or have met the SLA, i.e., $T_j/T_{j,min} < d$, respectively. Then the estimated minimal number of servers needed for these $Q_{SLA}(t)$ jobs to meet their SLAs is:

$$N_{min}(t) = \frac{Q_{SLA}(t)}{d} \tag{2}$$

Based on current states of all servers in the data center, we calculate the number of additional servers that are required to be waken up to meet the SLA (or the number of spare servers that can be put towards sleep states), i.e., $N_{req}(t)$, as:

$$N_{req}(t) = N_{min}(t) - (N_{active}(t) + N_{waking}(t) + N_{idle}(t)) \tag{3}$$

where $N_{active}(t)$, $N_{waking}(t)$, and $N_{idle}(t)$ are the numbers of active servers, servers in the waking up process, and idle servers at time t, respectively. $N_{req}(t) > 0$ indicates that some servers should be waken up in order to satisfy the overall SLA constraint, while $N_{req}(t) < 0$ represents that there is additional room for putting some servers to sleep to save energy. Then the power management policy is as follows:

(1) If $N_{req}(t) > 0$: the number of servers to wake up is $min(N_{req}(t), N_{sleep}(t))$, where $N_{sleep}(t)$ is the total number of sleeping servers the data center has at t.

(2) if $N_{req}(t) < 0$, then there is no need to wake up any server. Instead, we can put some idle servers to sleep states. However, as we also apply the *timeout mechanism*, rather than immediately putting spare servers into sleep states, we keep them idle and wait for $T_{idle}(t)$ to increase. In this case, the number of idle servers that should be activated at t to serve waiting jobs is:

$$\begin{cases} 0, & \text{if } N_{min}(t) \leq N_{active}(t) \\ min(\ N_{min}(t) - N_{active}(t),\ Q(t),\ N_{idle}(t)\ ), & \text{otherwise.} \end{cases}$$

B. Provision of Regulation Service (RS)

As introduced in Section II, the goal of the *RS policy* in a data center is to track the power signal $P_2^{target}(t) = \bar{P}_2 + z(t)R_2$ accurately, while also guaranteeing the workload QoS (i.e., satisfying the SLA), and improving the energy efficiency if possible. $\bar{P}_2$ and R_2 are estimated average power consumption and capacity reserves that are bid by the data center operator to ISO one hour ahead. In our *RS policy*, we leverage both server provisioning and the CPU resource limits knobs to track the ISO signal. We dynamically monitor power consumption $P_2(t)$ of data center at every second t and tune power based on the tracking performance. The *RS policy* is briefly introduced as follows. More detailed descriptions of the policy and the estimations of ($\bar{P}_2$, R_2) are in our previous work [12].

Case 1- If $P_2(t) < \bar{P}_2 + z(t)R_2$, i.e., the power consumption needs to be increased, we do the following three steps in order until $P_2(t) = \bar{P}_2 + z(t)R_2$:

(1) Increase power consumption of some active servers that are not running at maximal capacity to P_{max};

(2) If there are jobs in the queue and there are idle servers, then activate some idle servers with jobs in the queue and run them at maximal capacity with power consumption at P_{max};

(3) Resume some of sleeping servers.

Case 2- If $P_2(t) > \bar{P}_2 + z(t)R_2$, i.e., the power consumption needs to be decreased, we do the following two steps in order until $P_2(t) = \bar{P}_2 + z(t)R_2$:

(1) Decrease power consumption of some active servers to P_{min}. P_{min} is the minimum power consumption that we set when serving a job to avoid the job being stalled in the server for a long time and to guarantee QoS. P_{min} can be determined by the QoS requirements;

(2) Sleep some idle servers following *timeout mechanism*.

C. Peak Shaving

The goal of peak shaving is to eliminate the peak power so as to reduce the charges, while also guaranteeing the QoS. We propose a *peak shaving policy* that leverages both server power capping (using resource limits) and server provisioning. Assuming the original peak power of the data center is P_{peak}, and a β percent of peak is required to be shaved to, i.e., during the peak shaving time period (either an hour or a month), the data center has a strict power cap, βP_{peak} that cannot be violated. Similar to the RS program, we have a power consumption constraint to obey. However, unlike the ISO power constraint in RS that is dynamically changed, the constraint in peak shaving program is fixed during the time period. Moreover, in RS we track the power signal with some degrees of tolerable tracking error, while in peak shaving, though the power consumption is strictly capped at βP_{peak}, there is no further constraint on power consumption as long as the power is lower than the cap. Therefore, our *peak shaving policy* is a modified version of the *RS policy* introduced before and defined as follows:

Case 1- If $P(t) < \beta P_{peak}$, i.e., the power consumption is lower than the cap, then we do the following three steps in order, and stop once power consumption hits the capping value:

(1) Increase power consumption of some active servers that are not running at maximal capacity to P_{max};

(2) If there are jobs in the queue and there are idle servers, then activate some idle servers with jobs in the queue and run them at maximal capacity with power consumption at P_{max};

(3) If there are still jobs in the queue, but there is no spare servers, then wake up some servers until either the number of waken up servers equals to the number of jobs in the queue, or the total power consumption hits the power cap.

Case 2- If $P(t) > \beta P_{peak}$, i.e., the power consumption exceeds the cap, we do the following two steps in order until $P(t) = \beta P_{peak}$:

(1) Decrease power consumption of some active servers to P_{min}. P_{min} is the minimum power consumption that we set when serving a job to avoid the job being stalled in the server for a long time and to guarantee QoS. P_{min} can be determined by the QoS requirements;

(2) Sleep some idle servers following *timeout mechanism.*

D. Provision of Frequency Control (FC)

Similar to RS, the goal of frequency control (FC) is to consume power dynamically following a signal, while also guaranteeing the workload QoS (i.e., satisfying the SLA), and improving the energy efficiency if possible. However, in contrast to RS, which uses a centralized ISO signal that is broadcast every 4 seconds, the signal of FC is generated based on the local frequency deviation observation, and typically varies continuously, or changes much faster (10x) than the ISO signal in RS. In addition, demand side in FC is required to react immediately and exactly following the dynamics of the signal with its maximal possible capacity, i.e., tracking error is not adjustable. These two requirements result in difficulties for most demand side to participate in the FC program. However, the price of reserves in FC is much higher (5x) than that of reserves in RS, which may lead to more savings.

To modulate power consumption following a frequency deviation based FC signal, $\omega(t)$, and the correlated power target, $P_1^{target}(t)$ introduced in Section II, in our *FC policy*, the data center only leverages CPU resource limits (or DVFS), and does not apply server provisioning as the control knob, for the reason that the overhead of waking up a server is too large to meet the requirement of FC. Therefore, only active servers can provide reserves. DVFS can be modulated with μs - level overhead, and CPU resource limits can be modulated at *ms* - level in current hypervisors [25], thus, practically at real-time for our purposes. We expect future hypervisors or OS to provide finer granularity, lower overhead resource control options. We first estimate the ($\bar{P}_1$, R_1) for FC. Given the workload information and data center utilization, U, we estimate the number of servers that are needed to be active during the hour as:

$$N_{act}^h = \alpha U N_{dc} \tag{4}$$

recall that N_{dc} is the total number of servers in data center. Here $\alpha > 1$ is the slack provided in order to guarantee job QoS performance, as the data center should be capable of handling the situation when the number of new arrival jobs is larger than the average. In our *FC policy*, these N_{act}^h servers are always turned on, waiting for the incoming jobs and never sleep. We sleep all the rest servers and do not wake up them during the hour, i.e., we have $N^h_{sleep} = N_{dc} - N^h_{act}$ sleeping servers in that hour. A good α should guarantee QoS while also trying to minimize the energy consumption.

Since reserves are provided only from active servers, and it is assumed that all servers in the data center are homogeneous, estimating ($\bar{P}_1$, R_1) for the data center is equivalent to solve ($\bar{P}^{server}_1$, R^{server}_1) of each server. In order to guarantee job QoS and meet the SLA requirement, i.e., (d, η), we set a minimal power consumption for the active server, P_{min} as:

$$P_{min} = \frac{\beta(P_{max} - P_{idle})}{d} + P_{idle} \tag{5}$$

where $\frac{P_{max}-P_{idle}}{d} + P_{idle}$ is the minimal power required to meet the constraint d, without considering job waiting time in the queue. $\beta > 1$ is provided as the slack, considering the job waiting time. One way to determine β is provided in our previous work of the single server RS provision problem [13]. Then for each active server we have $(\bar{P}^{server}_1, R^{server}_1)$ as:

$$\bar{P}^{server}_1 = (P_{max} - P_{min})/2 \tag{6}$$

$$R^{server}_1 = P_{max} - \bar{P}^{server}_1 \tag{7}$$

and thus, $(\bar{P}_1, R_1)$ of the data center are:

$$\bar{P}_1 = \bar{P}^{server}_1 * N^h_{act} + P_{sleep} * N^h_{sleep} \tag{8}$$

$$R_1 = R^{server}_1 * N^h_{act} \tag{9}$$

Since we do not leverage different server states to provide reserves, our *FC policy* is similar to the single server RS provision policy introduced in the previous work [13], with an additional power budgeting strategy for the multiple server scenario. Our *FC policy* is as follows:

(1) At each time t, we equally distribute the power cap $P^{target}_1(t) = \bar{P}_1 + \omega(t)R_1$ to the active servers, hence the power for each server is $P^{target}_{server}(t) = \frac{P^{target}_1(t)}{N_{act}(t)}$, where $N_{act}(t)$ is the number of active servers at t, and $N_{act}(t) \leq N^h_{act}$. The power cap can be equally distributed amongst servers because of the server and workload homogeneity. Other budgeting policies can be applied ([49], [21]).

(2) Each server modulates the power consumption $P_{server}(t)$ with the CPU resource limits as follows, in order to best track the FC signal and the correlated power cap $P^{target}_{server}(t)$:

$$\begin{aligned} P_{server}(t) = argmin|P^{target}_{server}(t) - P_{server}(t)|, \\ P_{idle} < P_{server}(t) < P_{max}. \end{aligned} \tag{10}$$

V. EXPERIMENTAL RESULTS

In this section, we first compare the energy consumption, the peak power and the overall cost of multiple data center strategies proposed in Section IV, under the same data center configuration, with the same workload trace, and the same QoS SLA constraints. Then we modify the scenario by changing system utilization, types of workload, and sleep state characteristics. Finally we discuss how the control knobs of these strategies can be used to obey different SLA requirements.

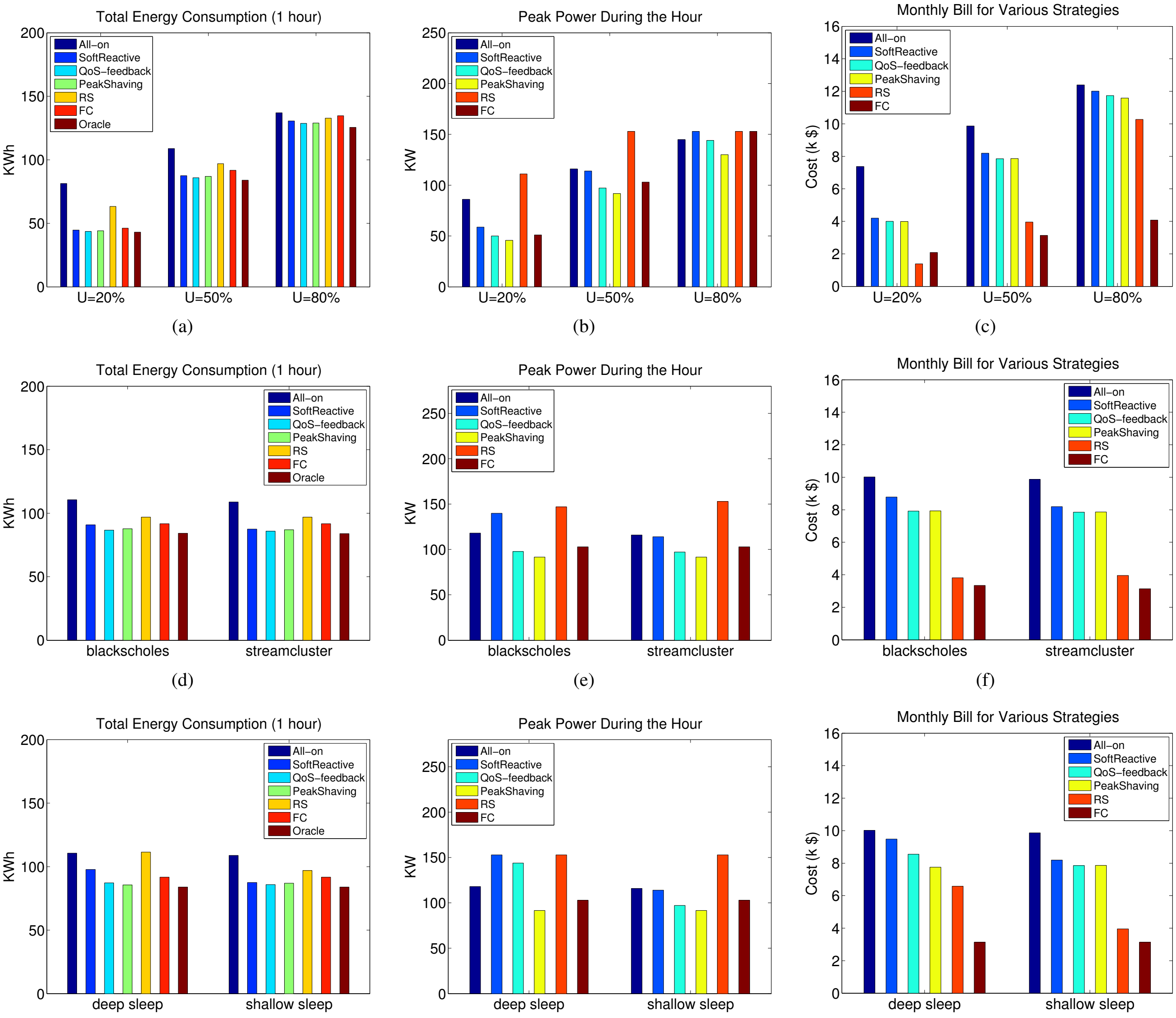
The number of servers in the data center of our experiment is, $N_{dc} = 1000$. Two types of server sleep mode are used for comparison: shallow sleep and deep sleep. For shallow sleep, we have the wake up delay $T_{up} = 10s$, and the power consumption in sleep state $P_{sleep} = 10\%P_{max}$. For deep sleep, $T_{up} = 200s$ and $P_{sleep} = 5\%P_{max}$ [29]. Server idle power is $P_{idle} = 63.0W$, and maximal power is $P_{max} = 152.94W$, measured on our real-life AMD server in the lab. Four workload traces are generated for testing: three of them are of streamcluster workload under different data center utilization levels as 20%, 50% and 80%, and the fourth one is a trace of blackscholes workload for the 50% utilization. Both streamcluster and blackscholes workload are from the PARSEC-2.1 [16] benchmark suite. The *shortest processing time* of streamcluster on our AMD server is $T^{stc}_{min} = 151.24s$, and of blackscholes is $T^{bls}_{min} = 23.5s$. The SLA constraint we use in experiments is $(d, \eta) = (2, 95\%)$, which represents that at least 95% jobs need to have a servicing time less than twice of the *shortest processing time*. Typically, data center operators require that the 95^{th} percentile of servicing time, d_{95}, stays below a certain threshold [22].

In our experiment, the clearing price of the energy consumed is, Π_E =10.7 cents/KWh, and the peak power price is Π_P=12 \$/KW (monthly) [26]. Prices vary in different regions and time. The price information that we use is from Georgia Power [26]. This is a reasonable reference as the world's largest data center, the Google data center, is located in Georgia. For emerging smart grid programs, e.g., the regulation service (RS) and frequency control (FC) in this paper, utilities do not charge peak power separately, and instead, the peak power cost is implicitly included in the cost of energy consumption. In order to make a fair comparison of total costs among different strategies, we calculate the converted energy consumption clearing price of RS and FC programs, Π^{cvt}_E, by taking peak power price into account, as follows:

$$\Pi^{cvt}_E = \frac{\Pi_E * E_m + \Pi_P * P^{peak}_m}{E_m} \tag{11}$$

where E_m is the monthly energy consumption and P^{peak}_m is the peak power in the month. We assume that a monthly power trace is 24*30 repetitions of an hourly power trace, then we have $E_m = 24 * 30 * E^d_h$ and $P^{peak}_m = P^{d,peak}_h$, where E^d_h and $P^{d,peak}_h$ are energy consumption and the peak power of the hourly power trace. The hourly power trace selected to do the price conversion in Eq. (11) is the one generated in the scenario of 50% streamcluster workload with the *All-on policy*. After conversion, Π^{cvt}_E=12.58 cent/KWh. In RS we assume the price of reserves, $\Pi_{R2} = \Pi^{cvt}_E$, and in FC we assume the price of reserves, $\Pi_{R1} = 5\Pi^{cvt}_E$.

The policies evaluated in the experiment include: *All-on, SoftReactive, QoS-feedback, PeakShaving, RS* and *FC*. For *All-on, SoftReactive, QoS-feedback* and *PeakShaving*, the total monetary cost is calculated as the sum of the cost of energy consumed and the penalty cost of peak power, using the price Π_E and Π_P, while for the *RS* and *FC*, the total costs are calculated by $\Pi^{cvt}_E \bar{P}_2 - \Pi_{R2}R_2$ and $\Pi^{cvt}_E \bar{P}_1 - \Pi_{R1}R_1$ respectively, recall that the penalty cost of peak power is included in the price Π^{cvt}_E.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 2. Comparison of energy consumption, peak power and monthly bill for different policies and programs under various scenarios. Figure 2(a), 2(b) and 2(c) are the comparison of policies under different data center utilization U = 20%, 50% and 80%; Figure 2(d), 2(e) and 2(f) are the comparison of policies with different workloads, i.e., blackscholes and streamcluster; Figure 2(g), 2(h) and 2(i) are the comparison of policies with server using "shallow" and "deep" sleep states.

A. Energy Efficiency and Monetary Costs Comparison

Figure 2(a) to 2(c) show the results of energy consumption in an hour, peak power and total monthly bill cost for different policies of the streamcluster trace under different utilization. In all these scenarios, the shallow sleep state is used when sleeping servers. We also include the Oracle energy consumption into comparison in Figure 2(a). Oracle energy consumption is the minimal energy required to finish all jobs. It theoretically sums up the energy needed for all jobs with the energy consumed in sleep state by all the spare servers.

Figure 2(a) shows that *QoS-feedback policy* always achieves the lowest energy consumption. It saves 8% - 46.4% comparing to the*All-on policy* and is only 1.2% - 2.4% greater than the Oracle. When increasing the data center utilization, the differences in energy consumption of various policies get smaller, because the flexibility of energy consumption of the data center with a high utilization is low. In addition, the energy consumption of *SoftReactive* and *QoS-feedback* are similar regardless of utilization. This is because that both of them present close results to the Oracle, hence, the room for further improvement is limited. One interesting result is, the *PeakShaving policy* also has low energy consumption.

Figure 2(b) shows that *PeakShaving policy* always achieves the lowest peak power in all scenarios as expected. It reduces the peak power around 10.3% -46.8% comparing to the *All-on policy*. It is also shown that lower utilization increases room for *PeakShaving* to reduce the peak power. In addition, *RS policy* always achieves the highest peak power, for the reason that in order to maximize the capacity reserve R_2 so as to increase the monetary saving, *RS policy* tries to reach to a very large dynamic power range. For *FC policy*, however, as we put spare servers always in sleep state and only regulate those active servers, the peak power is lower than that of the *RS policy*.

Figure 2(c) shows the monthly bill of different policies. The monthly bill is calculated based on the costs of the test hour multiplied by 720 hours/per month. The results show that comparing to the *All-on policy*, all the other policies save money. Among them RS and FC save the most. RS saves from 17.1% to 81.2% and FC saves from 67.1% to 71.7%. When utilization is low, RS saves the most, and when utilization is high, FC saves the most. This is because *RS policy* leverages different server states to provide reserves, and when utilization is lower, there is more room for RS to enlarge reserves. However, FC policy does not apply server states to provide reserves, hence the savings from FC are not sensitive to utilization.

Figure 2(d) to 2(f) compare the power consumption, peak power and monthly bill of policies with two different types of workloads, blackscholes and streamcluster, at 50% utilization. We compare between this two workloads because their processing time are quite different. Processing time of blackscholes is about 6 times shorter than that of streamcluster. There is no notable difference between these two cases shown from the results. Thus, the energy consumption, peak power and costs of these policies are not sensitive to types of workloads.

Figure 2(g) to 2(i) show the results of using different sleep state characteristics: shallow sleep and deep sleep. One notable change is, the cost of RS increases around 66.4% when using deep sleep instead of shallow sleep, while the cost of FC is unaffected. In addition, there are notable increases in peak power in both *SoftReactive* and *QoS-feedback policies*. This is because that waking up servers from deep sleep state takes longer time (200 sec), during which servers are at the maximal power. Thus, servers have higher probabilities of staying in a high power consumption state, which potentially leads to higher peak power of the data center.

Overall, all the polices help data center eliminate the monetary costs. Participating in emerging smart grid programs such as RS and FC can achieve even higher savings. When the utilization of data center is high or the sleep state is “deep”, FC outperforms RS. When the utilization is low, RS can provide more savings. Savings of all programs are insensitive to types of workloads.

B. Control Knobs for Different QoS Constraints

Table I lists the control knobs that are able to be leveraged in policies to satisfy different QoS constraints. There is no adjustable control knob for *All-on* and *SoftReactive* policies, as these two policies simply run workloads as soon as they arrive and as fast as possible. In the *QoS-feedback policy*, QoS constraints are guaranteed by the policy, as the policy always makes decisions dynamically based on the QoS measurement. However, the total energy consumed of *QoS-feedback policy* is not sensitive to the QoS requirement, as it has already been close to the Oracle bound and the room for further improvement is limited. In peak shaving, QoS is adjustable by changing the peak power cap, i.e., P_{cap}. A lower cap typically leads to poorer QoS. In both RS and FC programs, the QoS is adjustable by tuning the bidding values $\bar{P}_i$ and R_i. In our previous work we have discussed the estimation of $\bar{P}$ and R based on the QoS requirement [13], [12]. For example, the result of a sensitive analysis shows that QoS performance is much more sensitive to $\bar{P}$ rather than R [13]. Hence we first do the coarse tuning by changing $\bar{P}$, then the finer grain tuning by changing R. There is an additional control knob in FC that is able to impact the QoS, i.e., the percentage of servers put in sleep state. The larger number of servers are in sleep mode, the poorer the QoS will be. Currently, we determine the number of sleeping servers based on the data center estimated utilization and some fixed slack. A smarter algorithm could be designed to determine the percentage of sleeping servers based on both the data center utilization and the QoS requirement.

TABLE I. QoS TUNING KNOBS FOR VARIOUS POLICIES

Policies	*QoS Tuning Knobs*
All-on	N/A
SoftReactive	N/A
QoS-feedback	QoS feedback settings
PeakShaving	Peak power capping value P_{cap}
RS	$\bar{P}$ and R
FC	Percentage of sleeping servers, $\bar{P}$ and R

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a QoS based data center level dynamic power management policy, *QoS-feedback*, aiming at energy consumption minimization. We have also proposed policies for the data center to participate in various smart grid demand response programs, including peak shaving, RS and FC. We have compared the energy consumption, peak power and overall costs of different policies. Experimental results show that with a prerequisite to meet the QoS SLA requirement, in a typical 50% utilization data center, our *QoS-feedback policy* reduces the energy consumption up to 21.2%. Our *PeakShaving policy* reduces the peak power up to 21.0%. The proposed *RS policy* can reduce monetary costs up to 59.9%, and the proposed *FC policy* can save money up to 68.3%, compared to the *All-on policy* used in today’s data center, regardless of types of workload. Overall, it has been shown that participation in smart grid demand response programs, especially the emerging RS and FC programs, can help a data center achieve tremendous monetary savings, while also providing the ISOs and the society in general with cost effective demand side reserves that render massive renewable generation adoption affordable.

REFERENCES

[1] *Day-ahead and real-time pricing during a heat wave. http://avalonenergy.us/blog/?p=691.*

[2] *Amazon EC2 Spot Instances. http://aws.amazon.com/ec2/purchasing-options/spot-instances/.*

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

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

[5] U.S. Environmental Protection Agency. Report to congress on server and data center energy efficiency: Public law 109-431. Technical report, EPA Energy Star Program, 2007.

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

[7] B. Aksanli, E. Pettis, and T. Rosing. Architecting efficient peak power shaving using batteries in data centers. In *MASCOTS*, pages 242–253. IEEE, 2013.

[8] L. Benini, A. Bogliolo, and G. De Micheli. A survey of design techniques for system-level dynamic power management. *Transactions on Very Large Scale Integration (VLSI) Systems*, 8(3):299–316, 2000.

[9] M. Brocanelli, S Li, X. Wang, and W. Zhang. Joint management of data centers and electric vehicles for maximized regulation profits. In *IGCC*, pages 1–10. IEEE, 2013.

[10] T. D. Burd and R. W. Brodersen. Energy efficient CMOS microprocessor design. In *System Sciences, Proceedings of the 28th Hawaii International Conference on*, volume 1, pages 288–297. IEEE, 1995.

[11] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle. Managing energy and server resources in hosting centers. *ACM SIGOPS Operating Systems Review*, 35(5):103–116, 2001.

[12] H. Chen, M.C. Caramanis, and A.K. Coskun. The data center as a grid load stabilizer. In *Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific*, pages 105–112, 2014.

[13] 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.

[14] 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 (ICCAD)*, 2013.

[15] D. Chiu, C. Stewart, and B. McManus. Electric grid balancing through lowcost workload migration. *SIGMETRICS Performance Evaluation Review*, 40(3):48–52, 2012.

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

[17] 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 (MICRO)*, pages 175–185. ACM, 2011.

[18] Coincident Peak Pricing (CPP). www.fcgov.com/utilities/business/rates /electric/coincident-peak.

[19] G. Dhiman, G. Marchetti, and T. Rosing. vGreen: a system for energy efficient computing in virtualized environments. In *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design (ISLPED)*, pages 243–248. ACM, 2009.

[20] X. Fan, W.-D. Weber, and L. A. Barroso. Power provisioning for a warehouse-sized computer. In *ACM SIGARCH Computer Architecture News*, volume 35, pages 13–23. ACM, 2007.

[21] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy. Optimal power allocation in server farms. In *Proceedings of the ACM SIGMETRICS*, pages 157–168. ACM, 2009.

[22] A. Gandhi, M. Harchol-Balter, and M. A. Kozuch. Are sleep states effective in data centers? In *IGCC*, pages 1–10. IEEE, 2012.

[23] M. Ghamkhari and H. Mohsenian-Rad. Data centers to offer ancillary services. In *Smart Grid Communications (SmartGridComm), 3rd International Conference on*, pages 436–441. IEEE, 2012.

[24] G. Ghatikar, V. Ganti, N. Matson, and M. A. Piette. Demand response opportunities and enabling technologies for data centers: Findings from field studies. *LBNL-5763E. pdf*, 2012.

[25] Zhenhuan Gong, Xiaohui Gu, and John Wilkes. Press: Predictive elastic resource scaling for cloud systems. In *International Conference on Network and Service Management*, pages 9–16. IEEE, 2010.

[26] S. Govindan, A. Sivasubramaniam, and B. Urgaonkar. Benefits and limitations of tapping into stored energy for datacenters. In *Computer Architecture (ISCA), 38th Annual International Symposium on*, pages 341–351. IEEE, 2011.

[27] C. Hankendi, S. Reda, and A. K. Coskun. vCap: Adaptive power capping for virtualized servers. In *ISLPED*, pages 415–420. IEEE, 2013.

[28] I. Hwang, T. Kam, and M. Pedram. A study of the effectiveness of CPU consolidation in a virtualized multi-core server system. In *ISLPED*, pages 339–344. ACM, 2012.

[29] C. Isci, S. McIntosh, J. Kephart, et al. Agile, efficient virtualization power management with low-latency server power states. In *ISCA*, pages 96–107. ACM, 2013.

[30] J. G. Koomey. Worldwide electricity used in data centers. *Environmental Research Letters*, 3(3):034008, 2008.

[31] J. Li and J. F. Martinez. Dynamic power-performance adaptation of parallel computation on chip multiprocessors. In *High-Performance Computer Architecture (HPCA). The 12th International Symposium on*, pages 77–87. IEEE, 2006.

[32] M. Lin, Z. Liu, A. Wierman, and L. L. Andrew. Online algorithms for geographical load balancing. In *IGCC*, pages 1–10. IEEE, 2012.

[33] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew. Greening geographical load balancing. In *Proceedings of the ACM SIGMETRICS*, pages 233–244. ACM, 2011.

[34] Z. Liu, I. Liu, S. Low, and A. Wierman. Pricing data center demand response. In *ACM SIGMETRICS*, 2014.

[35] D. Meisner, B. T. Gold, and T. F. Wenisch. PowerNap: eliminating server idle power. *ACM Sigplan Notices*, 44(3):205–216, 2009.

[36] D. Meisner, C. M. Sadler, L. A. Barroso, W.-D. Weber, and T. F. Wenisch. Power management of online data-intensive services. In *ISCA*, pages 319–330. IEEE, 2011.

[37] M. P. Mills. The cloud begins with coal. Technical report, Digital Power Group, 2013.

[38] A. W. Mu'alem and D. G. Feitelson. Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling. *Parallel and Distributed Systems, IEEE Transactions on*, 12(6):529–543, 2001.

[39] 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.

[40] 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.

[41] U. Y Ogras, R. Marculescu, P. Choudhary, and D. Marculescu. Voltage-frequency island partitioning for gals-based networks-on-chip. In *Design Automation Conference (DAC)*, pages 110–115. IEEE, 2007.

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

[43] 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.

[44] 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.

[45] 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.

[46] H. Wang, J. Huang, X. Lin, and H. Mohsenian-Rad. Exploring smart grid and data center interactions for electric power load balancing. *ACM SIGMETRICS Performance Evaluation Review*, 41(3):89–94, 2014.

[47] 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.

[48] Y. Wang, X. Lin, and M. Pedram. A sequential game perspective and optimization of the smart grid with distributed data centers. In *Innovative Smart Grid Technologies (ISGT)*, pages 1–6. IEEE, 2013.

[49] X. Zhan and S. Reda. Techniques for energy-efficient power budgeting in data centers. In *Proceedings of DAC*, page 176. ACM, 2013.

[50] C. Zhao, U. Topcu, and S. H. Low. Frequency-based load control in power systems. In *American Control Conference (ACC), 2012*, pages 4423–4430. IEEE, 2012.