A Data Center Demand Response Policy for Real-World Workload Scenarios in HPC

Yijia Zhang, Daniel C. Wilson, Ioannis Ch. Paschalidis and Ayse K. Coskun
Boston University, Boston, MA, USA; E-mail: {zhangyj,danielcw,yannisp,acoskun}@bu.edu

***Abstract*—Demand response programs offer an opportunity for large power consumers to save on electricity costs by modulating their power consumption in response to demand changes in the electricity grid. Multiple types of such programs exist; for example, regulation service programs enable a consumer to bid for a sustainable amount of power draw over a time period, along with a reserve amount they are able to provide at request of the electricity service provider. Data centers offer unique capabilities to participate in these programs since they have significant capacity to modify their power consumption through workload scheduling and CPU power limiting. This paper proposes a novel power management policy and a bidding policy that enable data centers to participate in regulation service programs under real-world constraints. The power management policy schedules computing jobs and applies server power-capping under both the constraints of power programs and the constraints of job Quality-of-Service (QoS). Simulations with workload traces from a real data center show that the proposed policies enable data centers to meet both the requirement of regulation service programs and the QoS requirement of jobs. We demonstrate that, by applying our policies, data centers can save their electricity costs by 10% while abiding by all the QoS constraints in a real-world scenario.**

***Index Terms*—HPC, demand response, Quality of Service**

I. Introduction

Data centers[1] are significant power consumers. Since a server typically consumes hundreds of watts, a 2k-node data center can consume nearly 1 megawatt, which corresponds to thousands of dollars spent on power consumption every day. Data center consumption also occupies a considerable portion of nationwide energy usage. In 2014, all data centers in the US consumed 70 billion kWh, close to 2% of US electricity usage [1]. Therefore, reducing the power consumption and electricity costs of data centers is not only beneficial to data center owners or users, but also critical for a sustainable society.

With the increasing adoption of renewable energy production into the electricity grid, balancing the demand and supply side of the grid becomes challenging due to the volatility of renewable energy sources. Demand response programs offer a solution to this challenge by monetarily motivating consumers to regulate their power consumption following market requirements [2]. Regulation service is one of the demand response programs that are suitable for data centers [3], [4]. Regulation service programs require participants to regulate their power consumption following a regulation signal that varies every few seconds. Data centers are promising candidates for such programs because data centers are capable of regulating their power consumption in a wide range at the timescale of seconds.

Recent work has designed policies that enable data centers to participate in regulation service programs [5]–[7]. However, HPC systems typically run multi-node jobs with long execution times (from several hours to days), but those policies in previous works do not explicitly target jobs with long duration because they mostly rely on job scheduling to regulate power. Job scheduling, however, does not always provide sufficient control points when handling uninterruptible jobs with long duration. Also, long-duration jobs may go through multiple contract bidding cycles and previous bidding policies are not suitable for this scenario either. In addition, most prior approaches rely on synthetic workload traces for evaluation, and they seldom experiment with real workload traces which contain large variations in job arrival times.

We propose the QoS-aware-Capping policy to address these needs. Our power management policy relies on the power capping capability of servers to regulate data center power for demand response participation. The proposed policy applies power limits while considering the Quality-of-Service (QoS) of jobs. We also propose an Adaptive Bidding policy that selects appropriate contract parameters for data centers to participate in regulation service markets when applying the QoS-aware-Capping policy. We evaluate our policies by simulation using parameters and workload traces taken from a real data center [8]. Our results show that our proposed policies enable data centers to participate in regulation service and save 10% on electricity costs while abiding by all QoS constraints.

The contributions of this work are summarized as follows:

- We propose a power management policy, QoSCap, which enables data centers to participate in regulation service programs offering QoS awareness to computing jobs with long duration. Working with long-duration jobs is a necessary feature as real data center traces demonstrate.
- We propose an Adaptive Bidding policy for data center participation in regulation service to select appropriate contract parameters. When working in tandem with the QoS-aware-Capping policy, this bidding policy reduces data centers' electricity costs.
- We evaluate our proposed policy by simulation using system parameters and workload traces from a real data center, and we show the proposed policies outperform prior policies (e.g., Tracking-only [5] and EnergyQARE [6]).

Research was partially supported by the NSF under grants IIS-1914792, DMS-1664644, and CNS-1645681, by the NIH under grant R01 GM135930, by the ONR under grant N00014-19-1-2571, and by the Boston University College of Engineering under the Dean's Catalyst Award.

[1]In this work, we define data centers broadly, including both enterprise and high-performance computing (HPC) data centers.

978-3-9819263-5-4/DATE21/©2021 EDAA

II. Background on Demand Response and Regulation Service

There has been much progress in recent years in developing strategies for data centers to participate in power markets such as peak shaving [9], dynamic energy pricing [10], emergency load reduction [11], [12], etc. Among all power markets, regulation service programs are specially suitable for data centers because these programs require participants to regulate their power consumption at the time scale of seconds, and they offer significant electricity cost reduction in return. Data centers have large capacity to modulate their power consumption through various power management settings on servers.

When consumers participate in a regulation service program [3], they first bid for two key parameters, $\bar{P}$ and R, before starting each contract. $\bar{P}$ constrains the average power usage of the consumer. R represents the range above and below $\bar{P}$ in which the consumer is willing to modulate power usage throughout the contract. These two parameters can be changed every hour by updating the contract.

When $\bar{P}$ and R are determined, the consumer receives a signal $y(t)$ broadcast by an Independent System Operator (ISO) every 4 seconds. Then, the target power for this consumer is determined by

$$P_{tgt}(t) = \bar{P} + y(t)R. \tag{1}$$

The value of $y(t)$ is not known in advance, but it is limited in range $[-1, 1]$ and having a long-term average of 0.

Consumers are required to regulate their actual power consumption to match the target power closely. The consumer needs to follow a tracking error constraint enforcing that the relative tracking error defined as $\epsilon(t) = |P(t) - P_{tgt}(t)|/R$ cannot be larger than 0.3 for more than 10% of time. Here, $P(t)$ represents the actual power usage at time t. In other words, the tracking error constraint is Prob$[\epsilon(t) > 0.3] < 10\%$.

When a 1-hour contract ends, the electricity cost for the consumer is calculated based the bidding parameters $\bar{P}$, R and the average tracking error $\bar{\epsilon} = \mathrm{E}[\epsilon(t)]$. This electricity cost can be estimated by

$$\text{Cost} = \left(\Pi^P \bar{P} - \Pi^R R + \Pi^\epsilon R\bar{\epsilon}\right) \times 1\text{h}, \tag{2}$$

where Π^P, Π^R, and Π^ϵ are fixed coefficients determined by the market. In this paper, we assume $\Pi^P = \Pi^R = \Pi^\epsilon = \$0.1/\text{kWh}$.

III. Policies for Data Center Participation in Demand Response

In this section, we first introduce our models for data centers, servers, and workloads. Then, we discuss three power management policies including the new QoS-aware-Capping policy, as well as three bidding policies including the new Adaptive Bidding policy.

A. Data center and workload model

A typical data center consists of servers, a cooling system, a storage system, and other affiliated components. This work focuses on the power consumption of servers because server power can be easily and quickly regulated through power management techniques or tools such as dynamic voltage frequency scaling (DVFS) [13], Model-Specific Register (MSR) [14], and Running Average Power Limit (RAPL) [15].

We assume computing jobs submitted to a data center can be classified into different types. We assume there are separate queues for each job type so that it is possible for the job scheduler to prioritize a specific job type when needed. For a job type j, we assume we know the number of nodes m_j required to run the job,[2] the minimum execution time T_j^{min}, and the maximum power consumption per node p_j^{max}. We also assume we know the minimum power consumption per node p_j^{min} for running the job with maximal power-capping, and the maximum execution time T_j^{max} corresponding to execution under the minimum power consumption.

While there are reported manufacturing variations in processors [16], [17], we assume such variations are negligible in this work as variability information is not available in the data center traces we use, so running a job on any server has the same min/max power consumption and max/min execution time. Servers running jobs are called active servers, and servers not running jobs are called idle servers. We assume the power of all idle servers is identical, denoted as p_{idle}. Our policy would be able to take variation information into account through a preliminary set of runs that would characterize power/performance relationship on target servers.

To quantify the QoS of a job, we use a metric called *QoS degradation* defined as the extra time of processing a job divided by its minimum execution time. For a job of type j, its QoS degradation can be expressed as $Q_j = (T_j^{so} - T_j^{min})/T_j^{min}$. Here, T_j^{so} is the sojourn time of the job in the system, including the waiting time T_j^{wait} and the actual processing time T_j^{proc}, i.e., $T_j^{so} = T_j^{wait} + T_j^{proc}$. We assume there are constraints on the average QoS degradation of each type of jobs: Avg$[Q_j] < Q_j^{thres}$. Here, Q_j^{thres} is the QoS threshold for job type j. We assume we cannot stop a running job before it finishes.

B. Power management policies

Power management policies match the actual data center power with the data center's target power through job scheduling and server power capping. A common strategy for these power management policies is to start running more jobs and to

[2]In this paper, we interchangeably use the word *node* and *server*. A multi-node job means a job with $m_j > 1$.

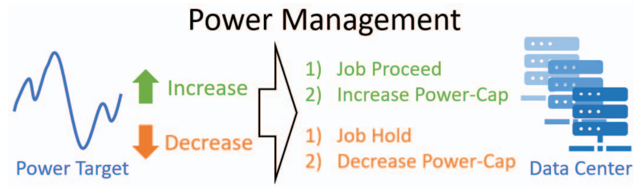


Fig. 1: Power management policies regulate data center power through job scheduling and power-capping to match the actual power consumption with the target power.

increase CPU power caps when the actual power is lower than the target power. These policies typically hold waiting jobs and decrease the CPU power caps when the actual power is higher than the target power, as shown in Fig. 1. In the following paragraphs, we first discuss the Tracking-only policy and the EnergyQARE policy proposed in previous works [5], [6]. Then, we introduce our new policy called QoS-aware-Capping.

The Tracking-only policy [5] regulates data center power following the basic strategy in Fig. 1 to track the target power without including job QoS in its control decisions. As a consequence, the policy views the target power as a hard upper bound and will never start a job if that leads to power consumption that exceeds the target power.

The EnergyQARE policy [6] also aims to track the ISO's target power signal in the same way as the Tracking-only policy, but it also considers QoS degradation of computing jobs. This policy balances the regulation service tracking error and the QoS degradation of finished jobs. As either metric degrades, more weight is put toward repairing the degradation. To be specific, excessive tracking error is addressed by following the target power as an upper bound. On the other hand, if the average QoS degradation is too high, then the policy activates more servers to run more jobs regardless of the target power.

The QoS-aware-Capping (QoSCap) policy not only applies the strategy in Fig. 1 but also intelligently adjusts power caps on servers considering the estimated QoS of jobs at run time. Every second, the policy calculates an *estimated QoS degradation* of each job by

$$Q_{est} = (T_{wait} + T_{elapse} + T_{remain})/T_{min}. \quad (3)$$

Here, T_{wait} is the waiting-in-queue time of the job. If the job is currently running, then T_{elapse} is non-zero and represents the time from job beginning to the current time. T_{remain} represents the remaining time to finish the job, estimated as the remaining percentage of the work to be done multiplied by the minimum execution time. In real world workloads, the percentage of work to be done for a job can be estimated based on the number of finished phases or loops of the job. Calculating the estimated QoS degradation metric enables the system to know which job's QoS degradation will exceed the threshold in advance.

Based on the estimated QoS degradation, the QoSCap policy always starts jobs whose QoS is close to violation. For the other jobs, the policy prioritizes job types with larger average QoS degradation. When target power is low, the policy decreases power caps only for job types meeting their QoS constraints. As a consequence, jobs waiting too long in the queue or running with too low power caps will have a higher estimated QoS degradation later, and they will be prioritized by the policy.

C. Bidding policies

Bidding policies select the bidding parameters, $\bar{P}$ and R, at the beginning of every hour. $\bar{P}$ and R determine the average and the variation of the target power, as shown in Fig. 2.

According to the electricity cost in Eq. (2), a data center pursues a smaller $\bar{P}$ and a larger R to reduce its cost. However, a $\bar{P}$ too small or R too large degrades QoS. Therefore, an appropriate bidding policy is needed. In the following, we discuss the Fixed Heuristics Bidding policy and the Fixed Exhaustive Search Bidding policy used in previous works [5], [6]. Then, we introduce our new Adaptive Bidding policy.

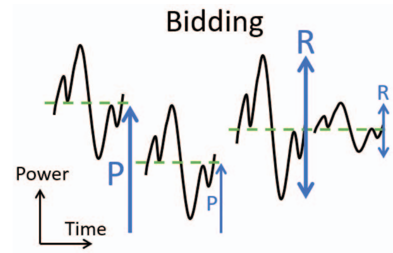


Fig. 2: Bidding policies select the appropriate $\bar{P}$,R parameters that determine the average and the variation of the target power.

The Fixed Heuristics (FH) Bidding policy selects $\bar{P}$ and R based on the long-term average power and power control range estimation. As long as the estimated job arrival rates do not change, that power estimation does not change over time, so the FH policy selects the same $\bar{P}$ and R at all times.

Assuming we know the arrival rate λ_j for each job type j, we can estimate the average number of active servers in the data center as $N_{active} = \sum_j \lambda_j m_j T_j^{min}$. Here, m_j is the size (i.e., required number of nodes) for each job of type j. Then, the total power of all idle servers can be estimated as $P_{idle} = (N_{total} - N_{active}) \times p_{idle}$ on average. The active servers can change their power consumption through power capping, so the total power of all active servers can vary from $P_{active}^{min} = N_{active} \times p^{min}$ to $P_{active}^{max} = N_{active} \times p^{max}$. Here, p^{min} or p^{max} is the min/max power of an active server, which can be estimated as the average min/max power of all job types, i.e., $p^{min} = (\sum_j p_j^{min})/J$, $p^{max} = (\sum_j p_j^{max})/J$. Here, J is the number of job types.

Based on the discussion above, we can derive the average minimum and maximum power of the data center as $P_{all}^{min} = P_{active}^{min} + P_{idle}$ and $P_{all}^{max} = P_{active}^{max} + P_{idle}$. Therefore, the FH policy selects $\bar{P} = (P_{all}^{max} + P_{all}^{min})/2$ and $R = (P_{all}^{max} - P_{all}^{min})/2$ as they are the estimated long-term average power and control range.

The Fixed Exhaustive Search (FES) Bidding policy selects the optimal parameters that minimize the electricity cost. It finds the optimal point through exhaustive search by running simulation over a wide range of $\bar{P}$, R with synthetic workloads (generated according to the arrival rates λ_j), and selects the parameters that minimize the electricity cost while meeting tracking-error and QoS constraints. This policy does exhaustive search once and applies the fixed values of $\bar{P}$, R to all hours.

The Adaptive Bidding policy determines the bidding parameters based on the current active and waiting jobs rather than based on the long-term average utilization of a data center. The motivation is, when long-duration and large multi-node jobs are common in a data center, the average power and the power control range in a certain hour could significantly deviate from their long-term average.

At the bidding time, the Adaptive Bidding policy calculates the possible max/min total power, P_{all}^{max} and P_{all}^{min}, by summing the max/min power of each active or waiting job. For an active or waiting job that suffers from a QoS violation ($Q_j > Q_j^{thres}$), its power is set as the maximum, p_j^{max} per node. For an active or waiting job not violating QoS constraints, its achievable min/max power is p_j^{min} and p_j^{max} per node. Idle nodes always consume power p_{idle}. After calculating P_{all}^{max} and P_{all}^{min}, the policy selects $\bar{P} = (P_{all}^{max} + P_{all}^{min})/2$ and $R = (P_{all}^{max} - P_{all}^{min})/2$.

IV. Experimental Methodology

To evaluate our policies in a real-world scenario, we conduct simulations using real system parameters and real workload traces taken from the *emmy* and *meggie* clusters at the Regional Computing Center in Erlangen (RRZE) [8]. The following subsections describe how we obtained data about these clusters and their workloads, and how our simulator utilizes that data.

A. Workload traces

Simulating a cluster requires the descriptions of the cluster's nodes, the workloads being executed, and the job submission times of the workloads in its job queue. We extract all of these properties from real traces provided by Patel et al. along with their analysis of the *emmy* and *meggie* clusters [8].

1) Node Properties: The *emmy* cluster has 560 nodes, and the *meggie* cluster has 728 nodes. The logs and traces we have available do not report the idle power of these nodes, but a workload efficiency model on this cluster by Klawonn et al. suggests that idle power is in the vicinity of 31 watts [18]. So, we assume $p_{idle} = 31$ W.

2) Workload Extraction: The *emmy* and *meggie* clusters do not use the same job scheduler and resource manager, and their trace outputs follow a different format with different types of events. We use a common subset of the events and properties that are available from both clusters. The selected job description fields are:

- Average DRAM and CPU power from start to end
- First enqueued time (when a job was enqueued and requeued multiple times, we only take the first time)
- Job start time and end time
- Anonymized job name
- Count of nodes assigned to the job

As the power reported in the logs only includes CPU and DRAM power, our simulation and analysis only considers the CPU and DRAM power, and cooling power is ignored. The data also reports job deletion events. We ignore any jobs that were deleted before they could start executing. An example of the power and execution time of jobs extracted from the log and used in our simulation are displayed in Fig. 3. Jobs with the same name and node count are shown as marks with the same color, size, and shape.

B. Simulator setup

We design a simulator based on the parameters of the clusters and workloads. The simulator is fed idle power and the host count for each cluster, as well as a list of jobs and their properties. The workload properties used by the simulator are nodes per job, maximum allowed performance degradation, minimum and maximum observed power, and the

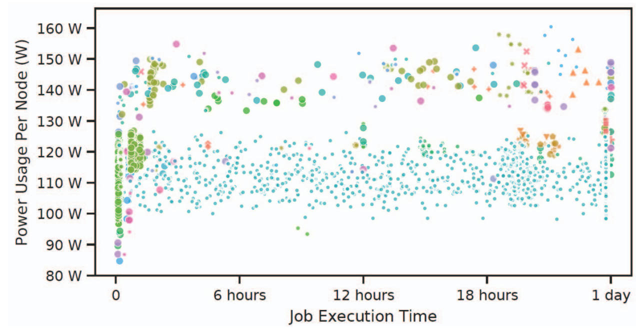


Fig. 3: The logged power and execution time of jobs running on the *meggie* cluster in February 1-8, 2019.

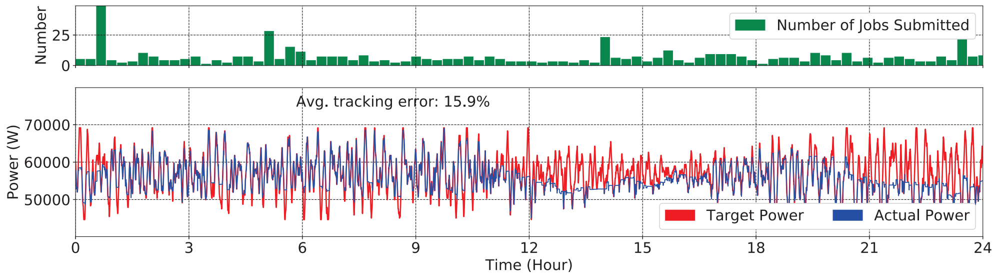
elapsed run time of the job when running under those power levels. When we apply the FES bidding policy, to generate a synthetic queue for bidding parameter searches, our simulator also takes the mean job incoming rates as inputs. For our final evaluations of the selected bidding parameters, we replay the queue submission times from the cluster logs instead of using the synthetic queues.

Given the cluster and workload parameters mentioned above, our simulator creates a cluster in simulation, updates the state of each server (idle, or running a job under a certain power-cap) once per second, and tracks the progress of job execution on each server. To simulate the effect of a continuous range of power capping on the execution time of a job, the simulator assumes there is a linear relation between the power and execution time for each job type. To be specific, when we apply a power cap on every node of a job type j, if we reduce the power cap from p_j^{max} to p_j^{min}, we assume the execution time linearly increases from T_j^{min} to T_j^{max}. For a multi-node job, we always apply the same power-cap to all the nodes running this job. It deserves mentioning that the working mechanism of our proposed policy does not depend on a specific relation between power-cap and job execution time, so our policy can be applied for non-linear power-performance relationships, as long as those relationships can be characterized in preliminary application runs. In this work, we adopt the linear assumption as this is a common trend for many real-world jobs and we do not have access to exact power-performance relationship of jobs in the datasets we use.

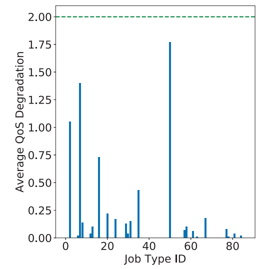
V. Results

We simulate the *meggie* or *emmy* cluster with job arrivals taken from multiple periods of the workload trace. Figure 4 presents a typical 24-hour result which simulates the *meggie* cluster with workload trace on Feb. 5^{th}, 2019. To get this result, we actually simulate the workload trace from Feb. $1^{st} - 5^{th}$ instead of starting directly from Feb. 5^{th}, so we can avoid the unrealistic underutilization of the cluster at the first few days. These simulations assume the threshold for the average QoS degradation of all job types (Q_j^{thres}) is 2.

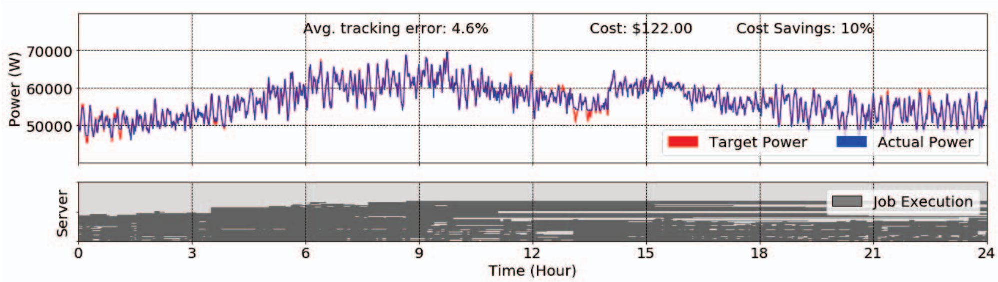
Figure 4(a) shows the results of applying the EnergyQARE with FES Bidding policy, and the average QoS degradation of each job type when applying these policies is shown in



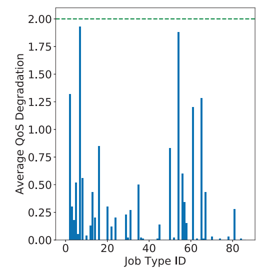
(a) EnergyQARE + FES Bidding policy results.



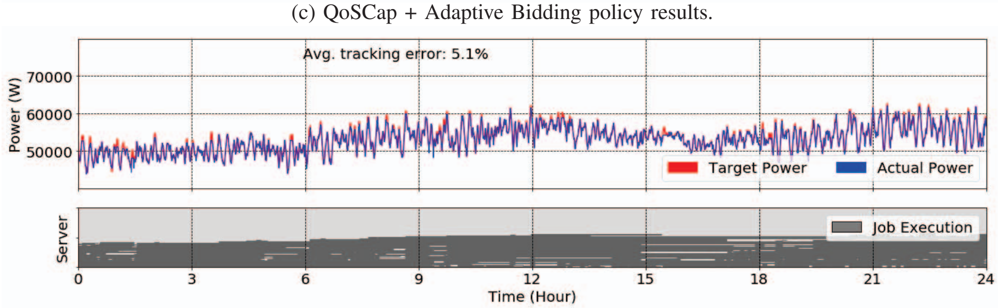
(b) QoS Degradation.



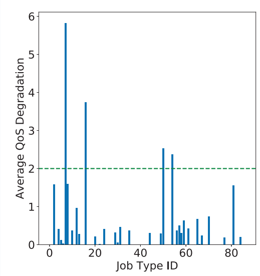
(c) QoSCap + Adaptive Bidding policy results.



(d) QoS Degradation.



(e) EnergyQARE + Adaptive Bidding policy results.



(f) QoS Degradation.

Fig. 4: Simulating a data center participating in regulation reserve programs when applying our policies. The simulations use real server parameters and run workload traces taken from a real 728-node cluster called *meggie*. In (c) and (d), the QoS-aware-Capping with Adaptive Bidding policy enables the data center to match its actual power (blue) with the target power (red) and meet the QoS constraints of jobs. In (a), green bars show the number of jobs submitted to the cluster in each time interval.

Fig. 4(b). Figures 4(c)(d) shows the results of the QoSCap and Adaptive Bidding policy. Figures 4(e)(f) are for the EnergyQARE with Adaptive Bidding policy. In Figs. 4(a)(c)(e), the red curve shows the target power and the blue curve shows the simulated actual power of the cluster. The green bars in Fig. 4(a) show the number of jobs submitted to the cluster in different time intervals, and these job arrivals are the same for Figs. 4(c)(e) because they simulate the same workload trace. In Figs. 4(c)(e), we also draw the time span of executing each job as the gray lines, where the vertical placement of the gray lines represents the server index number. We also calculate the electricity cost according to Eq. (2), and it shows that the proposed QoSCap with Adaptive Bidding policy enables the data center to get an 10% reduction of its electricity cost compared to the cost without regulation service participation.

From Fig. 4, we see the first two policy combinations meet the QoS constraints of jobs, QoSCap with Adaptive Bidding also meets tracking constraints as we see the actual power curve follows the target power closely. On the other hand, EnergyQARE with FES Bidding cannot meet tracking constraints as the actual power sometimes cannot follow the lower part of the target (during hours 4 to 10) and sometimes cannot follow the higher part of the target (during hours 11 to 16). The $\bar{P}$ and R for this case are already optimally selected using the FES Bidding policy, so other $\bar{P}$, R selections either violate the tracking constraints or violate the QoS constraints.

The reasons for EnergyQARE with FES Bidding not performing well include the abundance of long-duration multi-node jobs and the large variation of job arrivals. The work that proposed the EnergyQARE and FES Bidding policies [6] targets minute-long and single-node jobs which provide higher granularity to regulate power through job scheduling. In the real workload trace we simulate here, jobs mostly have an execution time of multiple hours (up to 24 hours) and many multi-node jobs require multiple nodes to run (up to 64 nodes).

Since jobs use many nodes at a time and span long durations, variations in job arrivals cannot be handled well by a fixed bidding parameter selection. On the other hand, Fig. 4(c) shows

Workload	Power Management	Bidding Policy	Tracking Error	QoS Degrad.	Energy Cost
Meggie Feb. 5 2019	EnergyQARE	FH	7.7%	4.0	-
		FES	18.1%	1.8	-
		Adaptive	2.0%	5.8	-
	QoSCap	FH	9.3%	1.5	$124
		FES	20.3%	1.4	-
		Adaptive	1.0%	1.9	$122
Meggie Jan. 16 2019	EnergyQARE	FH	2.2%	3.0	-
		FES	0.2%	4.8	-
		Adaptive	1.9%	2.6	-
	QoSCap	FH	18.0%	2.0	-
		FES	4.1%	1.9	$151
		Adaptive	0.2%	2.0	$166
Emmy Nov. 15 2018	EnergyQARE	FH	73.9%	2.2	-
		FES	0.8%	1.0	$115
		Adaptive	0.1%	2.1	-
	QoSCap	FH	70.4%	1.9	-
		FES	2.8%	2.1	-
		Adaptive	0.7%	2.0	$119

TABLE I: Simulation results of applying different policies for data center participation in regulation service. The proposed policy combination QoSCap+Adaptive is in bold. The "Tracking Error" column shows the percentage of large tracking error, and a value larger than 10% violates the tracking constraint. "QoS Degrad." column shows the largest average QoS degradation among all job types, and a value larger than 2.0 violates the QoS constraints. Values violating constraints are shown in red, otherwise in green. The electricity cost for policies that meet all constraints are displayed.

that the Adaptive Bidding policy can improve the tracking by selecting a higher $\bar{P}$ during hours 4 to 10 and a lower $\bar{P}$ later. However, EnergyQARE with Adaptive Bidding does not perform well as shown in Fig. 4(e)(f) because QoS constraints are not met. This is because the EnergyQARE policy shifts its priority between tracking and QoS based on the average QoS of all jobs. So, QoS violations of a small count of job types is not handled by the policy, which leads to the large QoS violation of some job types shown in Fig. 4(f).

Results for other combinations of power management and bidding policies are presented in Table I. The table shows whether a policy combination meets tracking or QoS constraints. It also includes the simulation of the *meggie* cluster with workload trace on Jan. 16^{th}, 2019, as well as the simulation of the *emmy* cluster with workload trace on Nov. 15^{th}, 2018. Among all policy combinations, only the QoSCap with Adaptive Bidding policy meets all constraints in all the three workload traces. The Tracking-only policy is not listed since it is ignorant of QoS and never meets all constraints. We have also simulated several other periods of the workload traces (not shown) and we observe the same result that QoSCap with Adaptive Bidding meets both constraints.

VI. Conclusion

To enable data centers to participate in demand response, we propose the QoSCap power management policy and the Adaptive Bidding policy. From simulations using real workload traces, we demonstrate that the proposed policies meet both the tracking constraints and the QoS constraints, while the other policies cannot meet all constraints due to the existence of long-duration and multi-node jobs as well as large variations in job arrival times in real workload traces. We show that our proposed policies enable data centers to reduce their electricity cost by 10% while abiding by all QoS constraints when participating in smart-grid power programs.

References

[1] A. Shehabi, S. Smith, N. Horner *et al.*, "United states data center energy usage report," *Lawrence Berkeley National Laboratory, Berkeley, California. LBNL-1005775 Page*, vol. 4, 2016.

[2] J. Hansen, J. Knudsen, and A. M. Annaswamy, "Demand response in smart grids: Participants, challenges, and a taxonomy," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 4045–4052.

[3] New York Independent System Operator (NYISO), "Ancillary services manual, v6.0," NYISO, Manual, May 2020. [Online]. Available: https://www.nyiso.com/manuals-tech-bulletins-user-guides

[4] A. L. Ott, "Experience with pjm market operation, system design, and implementation," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 528–534, 2003.

[5] H. Chen, M. C. Caramanis, and A. K. Coskun, "The data center as a grid load stabilizer," *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, no. i, pp. 105–112, 2014.

[6] H. Chen, Y. Zhang, M. C. Caramanis, and A. K. Coskun, "Energyqare: Qos-aware data center participation in smart grid regulation service reserve provision," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 4, no. 1, pp. 2:1–2:31, Jan. 2019.

[7] Y. Zhang, I. C. Paschalidis, and A. K. Coskun, "Data center participation in demand response programs with quality-of-service guarantees," in *ACM Intl Conf on Future Energy Systems (eEnergy)*, 2019, p. 285–302.

[8] T. Patel, A. Wagenhäuser, C. Eibel *et al.*, "What does power consumption behavior of hpc jobs reveal? : Demystifying, quantifying, and predicting power consumption characteristics," in *IEEE Intl Parallel and Distributed Processing Symposium (IPDPS)*, 2020, pp. 799–809.

[9] S. Govindan, A. Sivasubramaniam, and B. Urgaonkar, "Benefits and limitations of tapping into stored energy for datacenters," in *38th Intl Symposium on Computer Architecture (ISCA)*, 2011, pp. 341–352.

[10] T. N. Le, Z. Liu *et al.*, "Joint capacity planning and operational management for sustainable data centers and demand response," in *7th Intl Conf on Future Energy Systems*, ser. e-Energy '16, 2016, pp. 16:1–16:12.

[11] L. Zhang, S. Ren, C. Wu *et al.*, "A truthful incentive mechanism for emergency demand response in colocation data centers," in *IEEE Conf on Computer Communications (INFOCOM)*, 2015, pp. 2632–2640.

[12] N. H. Tran, C. Pham, S. Ren, Z. Han, and C. S. Hong, "Coordinated power reduction in multi-tenant colocation datacenter: An emergency demand response study," in *ICC*, May 2016, pp. 1–6.

[13] S. Reda, R. Cochran, and A. K. Coskun, "Adaptive power capping for servers with multithreaded workloads," *IEEE Micro*, vol. 32, no. 5, pp. 64–75, 2012.

[14] B. Rountree *et al.*, "libmsr," https://github.com/LLNL/libmsr, 2020.

[15] H. David, E. Gorbatov, U. R. Hanebutte, R. Khanna, and C. Le, "Rapl: Memory power estimation and capping," *ACM/IEEE Intl Symposium on Low-Power Electronics and Design (ISLPED)*, pp. 189–194, 2010.

[16] A. Marathe, Y. Zhang, G. Blanks *et al.*, "An empirical survey of performance and energy efficiency variation on intel processors," in *5th Intl Workshop on Energy Efficient Supercomputing*, ser. E2SC'17, 2017.

[17] D. Chasapis, M. Moretó, M. Schulz *et al.*, "Power efficient job scheduling by predicting the impact of processor manufacturing variability," in *ACM Intl Conf on Supercomputing*, ser. ICS '19, 2019, p. 296–307.

[18] A. Klawonn, M. Lanser, O. Rheinbach, G. Wellein, and M. Wittmann, "Energy efficiency of nonlinear domain decomposition methods," *The Intl Journal of High Performance Computing Applications*, 2020.