PowerChord: An Optimization Framework for Orchestrating Power-Grid-Aware Data Centers

Fatih Acun, Can Hankendi, Ioannis Ch. Paschalidis, Ayse K. Coskun
Boston University, Boston, USA, {acun, hankendi, yannisp, acoskun}@bu.edu

***Abstract*—The rapid growth of AI workloads has substantially increased data center energy demand, placing significant strain on power grids. While Independent System Operators (ISOs) offer demand response (DR) programs to improve grid stability, data center adoption remains low due to concerns that power restrictions will degrade workload performance. This paper introduces an optimization framework, `PowerChord`, that enables multiple data centers in the same ISO area to participate in DR programs. By co-optimizing DR participation, our approach increases the DR performance of data centers while reducing the risk of workload quality-of-service (QoS) violations. Our optimization framework enables higher power consumption flexibility through an enhanced inter-data-center power management method compared to individual DR participation. By increasing power flexibility by up to 22%, `PowerChord` reduces electricity costs by up to 12.1% relative to individual DR participation, while also mitigating QoS violations.**

I. Introduction

To meet growing computational demand, data centers are being built at an unprecedented pace, driving sharp increases in electricity consumption. As a result, they are becoming a major source of stress on power grids. For example, in Virginia, data centers are projected to account for over 20% of the state's total electricity use by 2030 [1]. This growth raises concerns about grid reliability, motivating the need for grid-aware data centers that can operate as *flexible energy consumers* [2], [3]. In this paper, we define flexibility as a data center's ability to modulate its power consumption in response to grid needs.

Data centers have significant potential to support independent system operators (ISOs) through demand response (DR) programs by adjusting power consumption through workload scheduling [4] and hardware power caps [5]. Prior work has studied these mechanisms and shown that data centers can provide grid services in multiple DR programs [6], [7]. In particular, participation in regulation service reserves (RSR) has been explored via fast power adjustments [5], [8]. Even though data centers could play a significant role in supporting power grids and receive financial incentives, real-world adoption of DR by data centers remains limited to a few examples [9], [10]. One key reason is that data centers are highly cautious about any power management actions that could cause Quality-of-Service (QoS) violations. Another constraint on DR participation is that data centers must comply with the power tracking requirements from ISOs and accurately adjust their power consumption [11]. However, ensuring accurate power tracking while preserving QoS places a dual burden on data centers. A further practical barrier is that ISOs must coordinate with a growing number of data centers, often operated by different providers, without a structured communication mechanism [1].

To address the challenges of adhering to power and QoS constraints for data centers, prior work explores coordinated scheduling across multiple data centers through geographical workload migration and load balancing, aiming to reduce the overall carbon footprint [12], [13]. Similarly, workload migration is used to reduce data center electricity costs by utilizing DR incentives [14]. Workload migration can help mitigate the negative effects of power constraints across data centers, but is not always practical due to high time and energy overheads for certain workloads [15]. Alternatively, a dynamic runtime framework is proposed for DR participation to allocate power across multiple data centers in the same ISO region according to their time-varying performance needs and enable coordination without workload migration [16]. However, this approach provides only heuristic runtime coordination and does not optimize DR bids across multiple data centers.

In this paper, we introduce `PowerChord`, an optimization framework that jointly utilizes flexibility across multiple data centers in the same ISO region to optimize power consumption, flexibility, and job performance for DR participation. By coordinating data centers without requiring workload migration, `PowerChord` broadens DR participation to independent facilities. This also addresses a practical challenge for ISOs, which otherwise must coordinate with many heterogeneous data centers individually. **Key contributions** of this paper are:

- An optimization framework for fleet-level data center participation in DR programs, providing a coordinated bidding approach without requiring workload migration.
- An automated augmented Lagrangian-based parameter selection method [17] for multi-data-center optimization that efficiently handles a larger parameter space by incorporating adaptive penalty terms for QoS and power tracking constraints.
- An extensive evaluation of our framework across diverse data center workloads, including AI inference, training, and traditional High-Performance Computing (HPC) jobs.

Our results show `PowerChord` meets all the QoS and power tracking constraints and achieves up to 12.1% monetary savings on data center energy bills by providing up to 22% higher flexibility, compared to the flexibility provided by each data center's individual DR participation.

II. A Multi-Participant Demand Response Optimization Framework for Data Centers

DR participation by multiple data centers through a data center-aware and grid-facing aggregator can increase flexibility, lower electricity costs, and improve adherence to QoS requirements. Such coordination also benefits the grid: when managed collectively, multiple data centers act as a single, more flexible resource, enabling greater and more reliable flexibility than independent participation. In this section, we first discuss our motivation, review the prior work that shaped its design, and then describe our optimization methodology.

A. Motivation

During DR events, data centers must meet power regulation requirements while maintaining QoS, which is difficult when handled independently. Optimization across multiple data centers allows this burden to be shared. For example, a data center facing higher workload demand can temporarily consume more power while another with looser QoS constraints absorbs a larger share of regulation. This improves responsiveness to both power tracking for the ISO and QoS for data centers.

Figure 1 illustrates `PowerChord` framework for optimizing multiple data centers. `PowerChord` optimizes three parameters for each data center i: average power consumption ($\bar{P}_i$), reserve capacity (R_i), and job-resource weights (W_i). Here, R is the capacity committed to the ISO for regulating power above and below $\bar{P}$, and W determines the ratio of resources allocated for different job types in each data center. `PowerChord` iteratively runs DR simulations with real-time QoS sharing and power balancing from previous work [16], and applies gradient descent to optimize $\bar{P}_i$, R_i, and W_i. In each iteration, `PowerChord` submits the total $\bar{P}$ and R to the ISO, receives the power target for the data center fleet, and runs the DR simulation, then updates parameters based on the costs of energy, QoS, and power tracking.

B. Formulation of Data Center Participation in RSR

In this paper, we focus on RSR programs, which offer real-time regulation critical to grid stability but impose strict demands for rapid power tracking [11]. In the RSR program, participants provide their bids for average power consumption, $\bar{P}$, and reserve, R, to ISOs before the DR execution. During execution, they are required to follow the power target, $P_{\text{target}}(t)$, based on provided $\bar{P}$ and R bids as follows:

$$P_{\text{target}}(t) = \bar{P} + y(t)R, \tag{1}$$

where t denotes time, and $y(t) \in [-1, 1]$ is the DR signal broadcast by ISO every 4 seconds [11]. Tracking error, $\epsilon(t)$, is defined as the absolute difference between a demand-side consumer's actual power consumption, $P_{\text{actual}}(t)$, and target power, $P_{\text{target}}(t)$, normalized by R:

$$\epsilon(t) = |P_{\text{actual}}(t) - P_{\text{target}}(t)|/R. \tag{2}$$

Then, the electricity cost for a participant is:

$$M = \left(\Pi^P \bar{P} - \Pi^R R + \Pi^\epsilon R\bar{\epsilon}\right) \times \mathrm{T}, \tag{3}$$

where Π^P, Π^R, and Π^ϵ are price coefficients, T is the duration, and $\bar{\epsilon}$ denotes average tracking error. This formulation incentivizes offering R and penalizes for tracking error.

Accurately estimating average $\bar{P}$ and R bids is critical for data center participation in RSR programs. Underestimating $\bar{P}$ can cause QoS violations by leaving too little power for computation, while overestimating $\bar{P}$ or R can increase tracking error if the data center cannot sustain high power or respond quickly to steep power target changes.

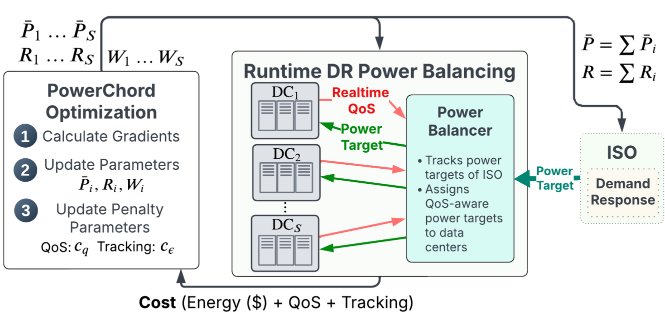

Fig. 1. Overview of the `PowerChord` framework for multi-data-center DR optimization using a runtime DR power-balancing methodology to follow ISO power targets.

C. Data Center Model and QoS Requirements

To track power targets by adjusting power consumption during DR execution for each data center, we use a DR runtime policy, the Adaptive QoS Assurance (AQA) framework [8]. We implement its data center model, which includes multiple queues for different job types, each assigned a weight, w_j, to determine the proportion of resources allocated for job type j. The runtime policy adjusts the number of active/idle servers and applies power caps to precisely match $P_{\text{target}}(t)$. To perform data center DR simulations, we build on the open-source simulator `FlexDC-Sim` [18] and extend it to support multi-data-center simulation. `FlexDC-Sim` models job performance and power consumption using profiles collected from real hardware, enabling realistic simulation of job execution under varying power caps during DR events.

To assess job performance in a data center, we use a relative-latency metric to quantify QoS degradation:

$$Q^{j,k} = (T^{j,k}_{\text{so}} - T^j_{\text{min}})/T^j_{\text{min}}, \tag{4}$$

where j denotes the job type and k indexes an individual job instance of type j. T^j_{min} is the minimum possible execution time for type j without queueing or power caps, and $T^{j,k}_{\text{so}}$ is the sojourn time from submission to completion. Each job type has a QoS threshold Q^j_{th}, and job instance k satisfies the QoS requirement if $Q^{j,k} \le Q^j_{\text{th}}$.

D. Multi-Data-Center Optimization with `PowerChord`

Our goal is to optimize multiple data centers for DR participation while ensuring that each meets its QoS and power tracking constraints. As the number of data centers increases, the number of optimization parameters grows linearly. However, this linear scaling leads to a combinatorial challenge when empirically tuning fixed penalty parameters, which control the relative weighting of different cost components for QoS degradation and tracking errors. To address this issue, we adopt the augmented Lagrangian method for inequality-constrained optimization [17]. This approach handles constraint violations related to tracking error and QoS by updating penalty parameters during the optimization process as violations are observed. The initial constraints for the optimization are defined as:

$$h_i(x_i) = \epsilon^r - \tau \leq 0, \tag{5}$$

$$g_i^j(x_i) = \Pr[Q^j > Q^j_{\text{th}}] - \delta \leq 0,\ \forall j \in [1, J] \tag{6}$$

where i denotes the data center index, $h_i(x_i)$ denotes the inequality constraint for tracking error, ϵ^r is the tracking error at 90^{th} percentile, and $\tau = 0.3$ is the upper bound threshold for tracking error at 90^{th} percentile. This constraint ensures the tracking error is less than or equal to 0.3, 90% of the time [8]. $g_i^j(x_i)$ denotes the set of inequality constraints for QoS assurance for each job type j, and J is the total number of job types. $\Pr[Q^j > Q^j_{\text{th}}]$ denotes the probability of jobs exceeding their QoS thresholds. $\delta = 0.1$ enforces that at least $1 - \delta = 90\%$ of jobs of type j meet the QoS threshold. To enable differentiable optimization and penalize constraint violations with a steeply increasing function, we reformulate the inequality constraints using the softplus function, $\ln(1 + e^z)$, with the following smooth constraint mappings:

$$H_i(x_i) = \ln(1 + e^{\rho h_i(x_i)}) - \Gamma \leq 0, \tag{7}$$

$$G_i^j(x_i) = \ln(1 + e^{\rho g_i^j(x_i)}) - \Gamma \leq 0, \tag{8}$$

where $\Gamma = \text{softplus}(0) = \ln(2)$. This transformation preserves the original feasible set, such that $H_i(x_i) \leq 0$ and $G_i^j(x_i) \leq 0$ if and only if $h_i(x_i) \leq 0$ and $g_i^j(x_i) \leq 0$, for $\rho > 0$, while providing smooth gradients. We formulate the optimization with the objective function $f_i(x_i)$ for data center i, as follows:

$$x_i = [\bar{P}_i, R_i, W_i], \tag{9}$$

$$\min\ f_i(x_i) = \left(\Pi^P \bar{P}_i - \Pi^R R_i + \Pi^\epsilon R_i \bar{\epsilon}_i\right) \times \mathrm{T}, \tag{10}$$

$$\text{s.t.} \quad H_i(x_i) \leq 0,\ G_i^j(x_i) \leq 0, \tag{11}$$

where x_i denotes the optimized parameters. Finally, we define the Lagrangian function L_i for each data center as follows:

$$\begin{aligned} L_i(x_i, c_{i,\epsilon}, c_{i,q}, \mu_{i,\epsilon}, \mu_{i,q}) &= f_i(x_i) \\ &+ \frac{1}{2c_{i,\epsilon}} \left(\max\left(0, \mu_{i,\epsilon} + c_{i,\epsilon} H_i(x_i)\right)^2 - \mu_{i,\epsilon}^2\right) \\ &+ \frac{1}{2c_{i,q}} \sum_j \left(\max\left(0, \mu_{i,q}^j + c_{i,q} G_i^j(x_i)\right)^2 - (\mu_{i,q}^j)^2\right), \end{aligned} \tag{12}$$

where $c_{i,\epsilon}$ and $c_{i,q}$ are non-decreasing penalty parameters, $\mu_{i,\epsilon}$ and $\mu_{i,q}^j$ are Lagrange multipliers. The resulting problem is solved as follows:

$$\textbf{Primal step: } x_i^{K+1} \approx \arg\min_{x_i} L_i \quad \text{(via GD)}, \tag{13}$$

$$\textbf{Dual step: } \mu_{i,q}^{j,K+1} = \max\left(0, \mu_{i,q}^{j,K} + c_{i,q}^K G_i^j(x_i^K)\right), \tag{14}$$

$$\mu_{i,\epsilon}^{K+1} = \max\left(0, \mu_{i,\epsilon}^K + c_{i,\epsilon}^K H_i(x_i^K)\right), \tag{15}$$

where K denotes the optimization step index. We update the penalty parameters only when the corresponding constraint is violated. Specifically, if $H_i(x_i^K) > 0$, we set $c_{i,\epsilon}^{K+1} = \gamma\, c_{i,\epsilon}^K$; otherwise $c_{i,\epsilon}^{K+1} = c_{i,\epsilon}^K$. Similarly, if $G_i^j(x_i^K) > 0$, we set $c_{i,q}^{K+1} = \gamma\, c_{i,q}^K$; otherwise $c_{i,q}^{K+1} = c_{i,q}^K$, where $\gamma > 1$ is a fixed multiplicative factor. We use gradient-descent-based optimization based on the analytical estimations of AQA for derivative calculations of $H_i(x_i)$ and $G_i^j(x_i)$ with respect to $\bar{P}$, R, and W [8].

Our optimization approach is a standard application of the augmented Lagrangian method, and without any convexity assumptions, convergence is expected to a local optimum. We verify this through extensive experimental evaluation in Section III by showing improvements over prior work.

E. Runtime Power Balancing During Demand Response

During the optimization with `PowerChord`, we enable power balancing across data centers to reflect the benefits of centralized DR participation in the optimized parameters. This enables the `PowerChord` optimizer to explore optimal power bids, with the enhanced elasticity of power allocations over time depending on data center workload performance.

We use a dynamic power allocation mechanism at runtime across data centers, introduced in prior work [16]. This framework allocates QoS-aware power targets to data centers through a *power balancer* that acts as a trusted 3[rd] party between the data centers and the ISO. The power balancer reads real-time QoS, $Q^R(t)$, from each data center every second, calculated by each data center as follows:

$$Q^R(t) = \frac{1}{J} \sum_j \frac{q_{0.9}\left(\left(T_{\text{wait}}(t) + T_{\text{exec}}(t)\right)/T^j_{\text{min}}\right)}{Q^j_{\text{th}}}, \tag{16}$$

where $T_{\text{wait}}(t)$ stands for the queue wait time and $T_{\text{exec}}(t)$ is the total execution time of jobs at current time t. $q_{0.9}$ denotes the 90^{th} percentile and finally the values are averaged across J job types. Using $Q^R(t)$ metric and individual bids, $\bar{P}_i$, from S many data centers, the power balancer calculates the power target assignments for each data center as follows:

$$P_{i,\text{target}}(t) = P_{\text{target}}(t) \left(\alpha \frac{Q_i^R(t)}{\sum_{k=1}^S Q_k^R(t)} + \phi \frac{\bar{P}_i}{\sum_{k=1}^S \bar{P}_k}\right), \tag{17}$$

where $P_{\text{target}}(t)$ is the total power target for all data centers calculated as $P_{\text{target}}(t) = \sum_{i=1}^S \bar{P}_i + y(t) \sum_{i=1}^S R_i$, α and ϕ are balancing coefficients summing up to 1.

III. Experimental Evaluation

In this section, we describe our evaluation methodology for `PowerChord` based on: (1) improvements in power bids for RSR participation, (2) meeting constraints for QoS and power tracking, (3) electricity costs.

A. Experimental Setup

Our experimental setup consists of three main components: (1) scenarios that include multiple participants, (2) workload traces that consist of job submissions, and (3) CPU and GPU job types used to generate the workload traces.

Scenario. To support a diverse set of experiments, we design scenarios involving multiple data centers, each assigned a distinct workload trace as shown in Table I. We cover a wide range of scenarios, including data centers with various characteristics such as tight/slack QoS requirements, low/high power consumption, CPU/GPU jobs, and low/high utilization.

Workload Trace and Job Properties. Table II shows the workload traces that we generate using a Poisson process with specified target system utilizations. Our workload traces consist of job types in two categories: (1) CPU jobs from NAS Parallel Benchmarks [19], and (2) GPU jobs consisting of AI training and inference for GPT-2, LLaMA-7B, BLOOM-560M, and ResNet-50. For CPU jobs, we use the power performance profiles collected under varying parallelization and power caps [8]. For GPU workloads, we perform power performance profiling on Chameleon Cloud [20] using nodes with 4 NVIDIA P100 GPUs, using the `--power-limit` feature of the NVIDIA System Management Interface [21]. We use tight QoS thresholds for inference applications ($Q_{\text{th}} \leq 5$), as suggested in prior work [22]. For other jobs, we vary Q_{th} from 2 to 8 to capture both tight and delay-tolerant workloads.

Baseline Methods. We compare `PowerChord` against two baseline approaches: AQA [8] and Conductor [16]. AQA proposes a data center DR framework designed for independent participation by a single data center, providing an optimization method to forecast $\bar{P}$, R, and job weights W, along with a runtime policy for power management during DR events. Its optimization method uses empirically decided fixed penalty parameters for different cost terms and does not provide a structured way to set penalty parameters. Poorly tuned penalties can therefore lead to an unfair comparison. To ensure fairness, we enhance AQA's optimization method by incorporating the same augmented Lagrangian approach used in `PowerChord`. The second baseline, Conductor, offers a runtime heuristic-based framework that enables multiple data centers to jointly execute DR, as described in Section II-E.

TABLE I
SCENARIOS WITH MULTIPLE PARTICIPANTS USING DIFFERENT WORKLOAD TRACES.

Scenario	# of Participants	Workload Traces
S1	2	tight-qos (tq), slack-qos (sq)
S2	2	high-util, low-util
S3	2	low-power(lp), high-power(hp)
S4	4	tq, sq, lp, hp
S5	2	inf-tight-qos, train
S6	2	inf-low-util, inf-high-util
S7	2	train-inf, inf-tight-qos
S8	3	inf-tight-qos, train, high-util

TABLE II
DESCRIPTION OF WORKLOAD TRACES, INCLUDING VARIOUS UTILIZATION LEVELS USING CPU AND GPU APPLICATIONS.

Workload Trace	Utilization	Description
tight-qos	80%	job types with tight QoS constraints
slack-qos	70%	job types with slack QoS constraints
high-util	90%	high cluster utilization
low-util	50%	low cluster utilization
low-power	70%	job types consuming $< 350\,\text{W}$
high-power	70%	job types consuming $> 350\,\text{W}$
inf-high-util	80%	Inference jobs with high utilization
inf-low-util	60%	Inference jobs with low utilization
inf-tight-qos	70%	Inference jobs with tight-QoS
train-inf	70%	Inference and training jobs mixed
train	70%	AI training jobs

B. Optimization and Demand Response Bids

Finding optimal bids is crucial for capturing workload trace heterogeneity (e.g., differences in QoS constraints, power performance characteristics, and utilization). These optimal bid configurations can shift under centralized DR participation, since data centers' joint commitments can change the amount of power flexibility that can be reliably offered. Figure 2 presents the optimal $\bar{P}$ and R bids generated by AQA and `PowerChord` optimization. Conductor is not shown in this comparison as it uses the same bids as AQA due to its role as a runtime policy without implementing an optimization framework for bidding. Across all scenarios, the average power consumption $\bar{P}$ is higher for the data center when (i) utilization is higher (Scenarios 2 and 6), (ii) QoS constraints are tighter (Scenario 1), or (iii) the workload trace contains more power-intensive applications (Scenario 3). Overall, AQA and `PowerChord` yield comparable total aggregate power, $\bar{P}$, differences within $\pm 3\%$ for each scenario. In terms of reserve bids R, `PowerChord` delivers substantial gains in six scenarios, improving total flexibility between 4.9% and 22% relative to the baseline (the sum of individually optimized R bids using AQA), as illustrated in Figure 3. In Scenarios 2 and 6, `PowerChord` produces slightly smaller R bids (up to 2.7%). We consider this behavior expected, since AQA-based individual optimizations fail to satisfy the QoS constraints, as we discuss in detail in the next section.

C. Meeting QoS Requirements

Maintaining a sufficient QoS is critical for data centers participating in DR, as it ensures that power restrictions do not impact customers. `PowerChord` uses the flexibility of multiple data centers to distribute the available power in a QoS-aware manner and optimizes data center parameters accordingly. To evaluate QoS violations, we first measure the percentage of job types whose QoS requirements are violated for each scenario and method. We compute this metric by calculating each job type's QoS degradation at the 90^{th} percentile, and mark j as violating if this value exceeds the corresponding threshold Q^j_{th}. Table III reports the percentage of violating job types among all J job types, and compares against the baselines. In all scenarios, `PowerChord` mitigates the QoS violations observed in individual DR participation by data centers, even if the Conductor framework is failing to mitigate with its runtime DR participation approach. Figure 4 shows the QoS degradation distributions of the job types for the data center executing inf-tight-qos workload in Scenario 5. For the individual DR participation with AQA, QoS violations are observed for job types GPT2 and Bloom, as their QoS degradations at the 90^{th} percentile are approximately 5.1 and 6.1, being above their thresholds at 4 and 5. Those violations are mitigated with `PowerChord` such that the resulting 90^{th} percentile degradations are below the thresholds.

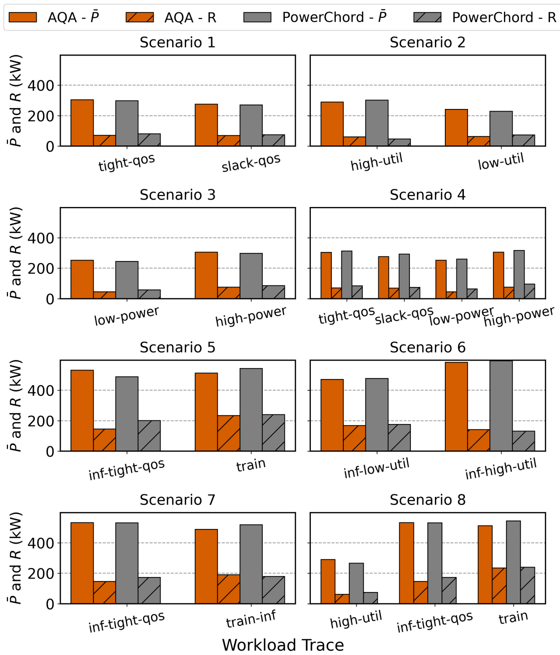

Fig. 2. $\bar{P}$ and R bids with AQA and PowerChord.

To provide an in-depth evaluation for QoS violations in a data center, we define a metric, $\bar{Q}$, which calculates the percentage of job submissions exceeding QoS threshold, per job type, and takes the average across job types as follows:

$$\bar{Q} = \frac{1}{J} \sum_{j} \left(\frac{\sum_{k \in \mathcal{N}_j} \left(Q^{j,k} > Q^{j}_{\text{th}} \right)}{|\mathcal{N}_j|} \times 100 \right), \tag{18}$$

where J is the total number of job types, $\mathcal{N}_j$ is the set of unique job submissions for job type j, k identifies a unique job submission. Figure 5 shows $\bar{Q}$ values for all scenarios. We observe that all data centers improve $\bar{Q}$ for all scenarios with PowerChord compared to AQA. Although Conductor provides a lower $\bar{Q}$ for certain scenarios for some data centers compared to PowerChord, such as the tight-qos data center in Scenario 1, it fails to provide improvement for the other data center(s) in the scenario. For Scenarios 5 and 8, Conductor achieves better QoS among other methods; however, we note

TABLE III

PERCENTAGE OF JOB TYPES VIOLATING THEIR QOS THRESHOLDS USING AQA [8], CONDUCTOR [16], AND POWERCHORD. WORKLOAD TRACES AND SCENARIOS WITH 0% QOS VIOLATIONS ACROSS ALL METHODS ARE OMITTED FOR SIMPLICITY.

Scenario	**Workload**	**AQA** [8]	**Conductor** [16]	PowerChord
S1	tight-qos	**25%**	0%	0%
S2	high-util	0%	**37.5%**	0%
	low-util	**25%**	0%	0%
S3	high-power	**12.5%**	**12.5%**	0%
S4	tight-qos	**25%**	0%	0%
	high-util	**12.5%**	**12.5%**	0%
	high-power	0%	**12.5%**	0%
S5	inf-tight-qos	**50.0%**	0%	0%
S6	inf-low-util	**25.0%**	**25.0%**	0%
	inf-high-util	**55.0%**	**75.0%**	0%
S7	train-inf	**25.0%**	**25.0%**	0%
	inf-tight-qos	**50.0%**	**50.0%**	0%
S8	inf-tight-qos	**50%**	0%	0%

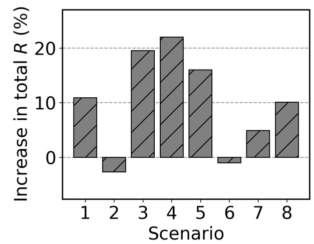

Fig. 3. Total R increase (%) using PowerChord relative to AQA for all scenarios.

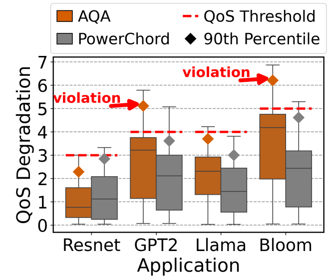

Fig. 4. QoS degradations for the data center executing inf-tight-qos trace in Scenario 5.

that PowerChord provided 16% and 10% more reserve flexibility while still meeting QoS constraints.

D. Tracking Power Targets

Power target tracking is a critical requirement of the RSR program, as participants are expected to strictly follow the power targets. We define tracking error as the absolute difference between the actual power consumption and the power target, normalized by R, as shown in Eq. (2), and this error must remain within a threshold of 0.3 at the 90$^{\text{th}}$ percentile. In individual DR participation, each data center must independently satisfy this constraint. In contrast, under participation with PowerChord, the requirement applies to the aggregate behavior of the data center fleet since the ISO evaluates data centers as a single entity. Figure 6 shows the aggregate power tracking (top panel), power target assignments to each participant (middle panel), and real-time QoS values (bottom panel) at runtime in Scenario 7. The power balancer uses the $\bar{P}_i$ bids and real-time QoS (Q^R_i) values to assign power targets to each participant (see Eq. (17)) by distributing P_{target}. As illustrated in Figure 2, the $\bar{P}$ bids of participants with workload traces, train-inf, and inf-tight-qos for optimization in Scenario 7, using PowerChord, are 518,582 W and 531,001 W, respectively. However, the power

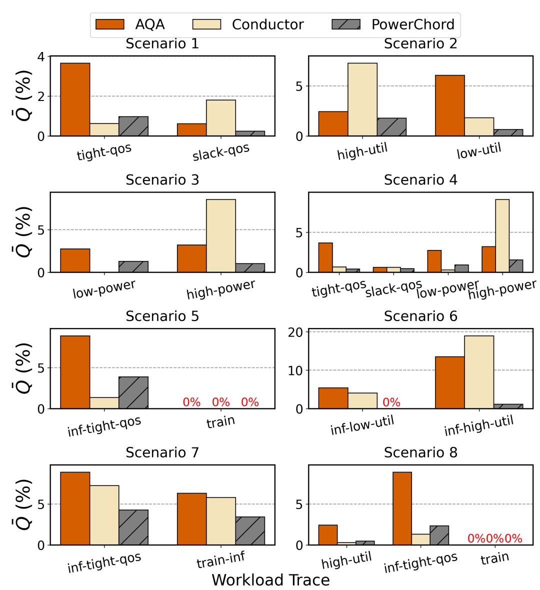

Fig. 5. PowerChord improves aggregated QoS violation percentage ($\bar{Q}$) for each data center across all scenarios.

budget of the data center with the train-inf workload can exceed that of the other one, depending on the real-time QoS values. As observed around 1000-1200 seconds, train-inf data center power exceeds the inf-tight-qos data center even though it provided a lower $\bar{P}$ bid. This example reveals the underlying flexibility for `PowerChord` to mitigate the QoS violations, while still tracking the fleet-level total power target.

Figure 7 illustrates the tracking error at the 90$^{\text{th}}$ percentile across all scenarios, comparing tracking errors for individual participation by each participant using AQA shown with circle-hatched bars, and aggregated tracking error for the data center fleet with Conductor and `PowerChord` with yellow and gray bars. Overall, `PowerChord` satisfies the tracking error constraint in every scenario, despite producing higher R bids, which makes power tracking inherently more challenging.

E. Analysis of Electricity Cost

The increased R bids lead to lower electricity cost since offering more R is incentivized in Eq. (3). Figure 8 shows the normalized electricity cost with respect to the sum of electricity costs for individual data centers using AQA. The total electricity cost of data centers with `PowerChord` is lower in 7 out of 8 scenarios compared to AQA, achieving up to a 12.1% reduction. In Scenario 6, `PowerChord` converges to more conservative bids, reducing total R by 1% and increasing P by 1.5%, which raises electricity cost by 1.3%. Unlike AQA and Conductor, which incur up to 75% QoS violations, `PowerChord` eliminates these violations, making the higher cost a reasonable tradeoff for QoS-critical inference workloads.

We allocate the total cost by quantifying each data center's contribution to it, using a simple proportional mechanism. The total electricity cost, M_{total}, is calculated based on the total bids $\bar{P} = \sum_{i=1}^{S} \bar{P}_i$, $R = \sum_{i=1}^{S} R_i$, and the aggregate tracking error, $\bar{\epsilon}_{\text{agg}}$, as described in Eq. 3. Then, the cost allocated to data center i is given by $M_i = \alpha_i \cdot M_{\text{total}}$, where α_i is the cost allocation ratio. To properly distribute the electricity cost to multiple data centers, we calculate α_i as a weighted combination of: (1) energy consumption E_i, (2) R_i, and (3) the average tracking error ($\bar{\epsilon}_i$) of each data center:

$$\alpha_i = e \cdot \frac{E_i}{\sum_k E_k} + r \cdot \left(\frac{1/R_i}{\sum_k 1/R_k} \right) + t \cdot \frac{\bar{\epsilon}_i}{\sum_k \bar{\epsilon}_k}, \tag{19}$$

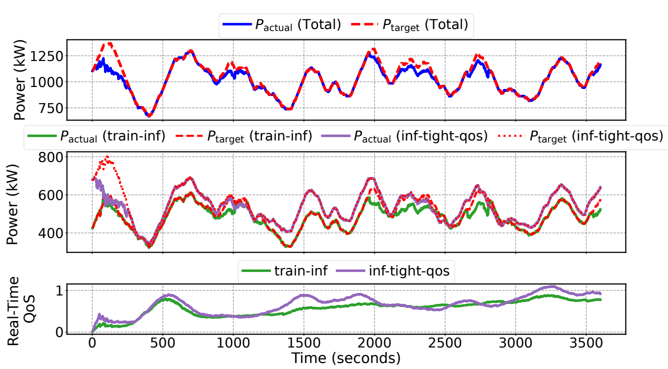

Fig. 6. Runtime aggregated power tracking of data centers with train-inf and inf-tight-qos workload traces in Scenario 7.

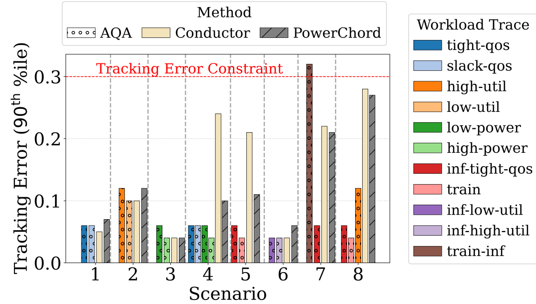

Fig. 7. Tracking errors at 90$^{\text{th}}$ percentile across all scenarios using AQA, Conductor and `PowerChord`.

where coefficients e, r, and t assign varying weights on different terms. We set the coefficients for each term such that data centers are proportionally responsible for the cost by E and $\bar{\epsilon}_i$, but inversely proportional to R since it helps reduce the total cost. To determine α_i, we set e, r, and t for each scenario by analyzing the impact of different components in the total cost for each scenario as follows:

$$D = \Pi^P \bar{P} + \Pi^R R + \Pi^\epsilon R \,\bar{\epsilon}_{\text{agg}}, \tag{20}$$

$$e = \frac{\Pi^P \bar{P}}{D}, r = \frac{\Pi^R R}{D}, t = \frac{\Pi^\epsilon \,\bar{\epsilon}_{\text{agg}}}{D}, \tag{21}$$

where D denotes the total magnitude of the cost components of the collaboration, and e, r, and t coefficients are calculated based on their proportional ratio to D.

Figure 9 shows the electricity costs of each participant with `PowerChord` compared to individual participations with AQA. In scenarios 1, 3, and 5, every participant reduces its costs compared to individual participation. The percentage of monetary savings depends on the cost allocations defined above and may result in varying savings due to contributions to the overall cost, such as offering more R, consuming less energy, and having less tracking error. We note that the electricity cost for each data center is not necessarily required to be lower compared to baselines, since they are not adhering to constraints such as QoS and tracking error. For example, in Scenario 7, the participant with the train-inf trace has a higher cost with `PowerChord` compared to AQA. However, AQA violates the QoS and tracking error constraints as illustrated in Figure 7 and Table III.

IV. Conclusion

The data center industry is growing rapidly, making it increasingly important to redesign data center operations to better align with power grid needs. In this work, we take a holistic view and show that optimizing multiple data centers

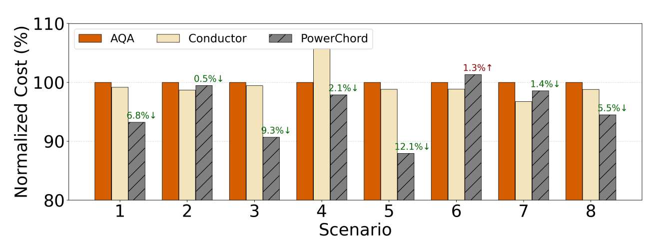

Fig. 8. Electricity costs normalized with respect to the total sum of individual costs of data centers using AQA.

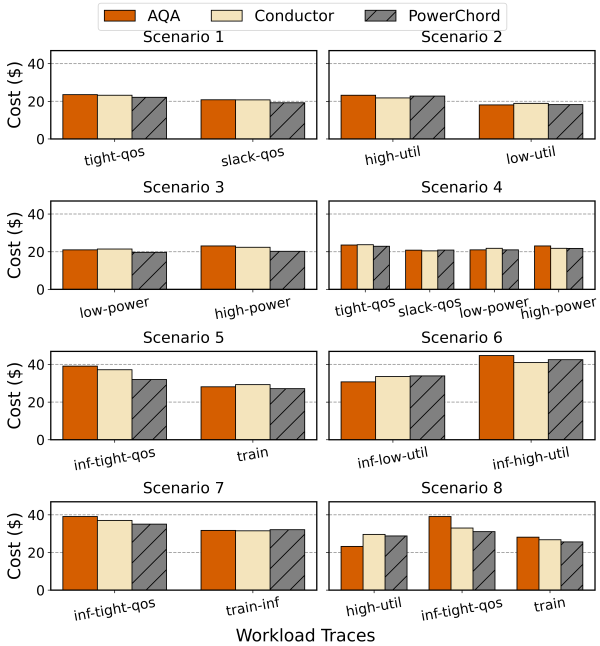

Fig. 9. Electricity cost allocations of participants.

can improve DR participation outcomes for both sides: the power grid benefits, while data centers can also improve performance and reduce electricity costs. As ISOs introduce new programs to manage large loads, the corresponding market mechanisms continue to evolve. In practice, an aggregator or Curtailment Service Provider could coordinate data centers as a single DR resource, enabling fleet-level bidding and tracking, while reducing the operational burden on the ISO.

Our multi-data-center framework requires each data center to share only limited QoS and power-related information with a trusted third party. Each consists of a single scalar value that can be communicated quickly with minimal overhead. Potential privacy concerns related to information sharing can be addressed by extending `PowerChord` with differential privacy or secure multi-party computation [23]. Our evaluation shows that `PowerChord` satisfies the QoS and power-tracking constraints during RSR participation across all experimental scenarios with AI and HPC workloads, while improving the flexibility by up to 22%.

Acknowledgements

This work is funded by the National Science Foundation DESC program under Award No. 2450111.

References

[1] Electric Power Research Institute (EPRI), "Electric Power Usage in Data Centers." https://www.epri.com/research/products/000000003002028905, 2024. Accessed: 2025-07-24.

[2] N. Groom, "Big Tech's data center boom poses new risk to U.S. grid operators." https://www.reuters.com/technology/big-techs-data-center-boom-poses-new-risk-us-grid-operators-2025-03-19/, 2025. Reuters, Accessed: 2025-03-24.

[3] P. Parikh, M. Ali, N. Virani, and S. C. Evans, "Strain to synergy: Powering ai data centers with sustainable and smart power grid infrastructures," in *2025 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–6, 2025.

[4] J. L. Berral, I. n. Goiri, R. Nou, F. Julià, J. Guitart, R. Gavaldà, and J. Torres, "Towards energy-aware scheduling in data centers using machine learning," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, e-Energy '10, (New York, NY, USA), p. 215224, Association for Computing Machinery, 2010.

[5] A. Jahanshahi, N. Yu, and D. Wong, "Powermorph: Qos-aware server power reshaping for data center regulation service," *ACM Trans. Archit. Code Optim.*, vol. 19, aug 2022.

[6] H. Wang, J. Huang, X. Lin, and H. Mohsenian-Rad, "Proactive demand response for data centers: A win-win solution," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1584–1596, 2016.

[7] A. Wierman, Z. Liu, I. Liu, and H. Mohsenian-Rad, "Opportunities and challenges for data center demand response," in *IEEE International Green Computing Conference*, (Dallas, TX, USA), pp. 1–10, IEEE, 2014.

[8] Y. Zhang, D. C. Wilson, I. C. Paschalidis, and A. K. Coskun, "Hpc data center participation in demand response: An adaptive policy with qos assurance," *IEEE Transactions on Sustainable Computing*, vol. 7, no. 1, pp. 157–171, 2022.

[9] V. Mehra and R. Hasegawa, "Using demand response to reduce data center power consumption — google cloud blog," Oct 2023.

[10] J. Kwan, "Climate change threatens supercomputers," *Science (New York, NY)*, vol. 378, no. 6616, pp. 124–124, 2022.

[11] PJM Interconnection, "PJM Manual 12: Balancing Operations, Revision 44." https://www.pjm.com/-/media/DotCom/documents/manuals/archive/m12/m12v44-balancing-operations-03-01-2022.pdf, Mar. 2022. Accessed: 2025-03-24.

[12] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Geographical load balancing with renewables," *ACM SIGMETRICS Performance Evaluation Review*, vol. 39, no. 3, pp. 62–66, 2011.

[13] J. Zheng, A. A. Chien, and S. Suh, "Mitigating curtailment and carbon emissions through load migration between data centers," *Joule*, vol. 4, no. 10, pp. 2208–2222, 2020.

[14] Y. Zhang, A. Tsiligkaridis, I. C. Paschalidis, and A. K. Coskun, "Data center and load aggregator coordination towards electricity demand response," *Sustainable Computing: Informatics and Systems*, vol. 42, p. 100957, 2024.

[15] N. Hogade, S. Pasricha, and H. J. Siegel, "Energy and network aware workload management for geographically distributed data centers.," *IEEE Trans. Sustain. Comput.*, vol. 7, no. 2, pp. 400–413, 2022.

[16] F. Acun, I. C. Paschalidis, and A. K. Coskun, "Conductor: A collaboration framework for multi-data-center demand response," in *2024 IEEE 15th International Green and Sustainable Computing Conference (IGSC)*, (Austin, TX, USA), pp. 22–28, IEEE, 2024.

[17] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 3 ed., 2016.

[18] F. Acun, C. Hankendi, E. Levine, H. Reynolds, J. Bardwick, and A. K. Coskun, "Investigating power consumption flexibility of ai data centers for demand response participation," E-Energy '26, (New York, NY, USA), p. 344348, Association for Computing Machinery, 2026.

[19] NASA Advanced Supercomputing Division, "NASA Parallel Benchmarks." https://www.nas.nasa.gov/software/npb.html. Accessed: 2024-06-29.

[20] K. Keahey, J. Anderson, Z. Zhen, P. Riteau, P. Ruth, D. Stanzione, M. Cevik, J. Colleran, H. S. Gunawi, C. Hammock, J. Mambretti, A. Barnes, F. Halbach, A. Rocha, and J. Stubbs, "Lessons learned from the chameleon testbed," in *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC '20)*, USENIX Association, 2020.

[21] NVIDIA Corporation, "Nvidia system management interface (nvidia-smi)." https://developer.nvidia.com/nvidia-system-management-interface, 2024. Accessed: 2025-07-24.

[22] J. Stojkovic, C. Zhang, . Goiri, J. Torrellas, and E. Choukse, "Dynamollm: Designing llm inference clusters for performance and energy efficiency," in *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, (Las Vegas, NV, USA), pp. 1348–1362, IEEE, 2025.

[23] S. N. Guzelhan, F. Acun, C. Hankendi, A. K. Coskun, and A. Joshi, "Privacy-preserving data center demand response using multi-party computation," in *Euro-Par 2026: Parallel Processing*, Lecture Notes in Computer Science, Springer, 2026.