# Evaluating and Comparing the Potentials in Primary Response for GPU and CPU Data Centers

Yihong Zhou[1], Ángel Paredes[2], Chaimaa Essayeh[1], and Thomas Morstyn[3]

[1]University of Edinburgh, U.K., Email: {yihong.zhou, cessayeh}@ed.ac.uk
[2]University of Málaga, Spain, Email: angelparedes@uma.es
[3]University of Oxford, U.K., Email: thomas.morstyn@eng.ox.ac.uk

***Abstract*—The rapid growth of Large Language Models (LLMs) and Artificial Intelligence (AI) has transformed traditional CPU-centric Data Centers (DaCe) into more power-demanding GPU DaCes. Previous work has explored methods to reduce energy costs and carbon emissions in GPU DaCes. However, there remains a gap in understanding the potential of GPU DaCes for providing primary response, a crucial ancillary service for stabilizing the power system. Drawing on real-world job traces from a GPU-intensive DaCe operated by SenseTime and a CPU-intensive DaCe at Oak Ridge National Laboratory, we developed a mixed-integer linear programming model to assess the DaCe flexibility potentials considering individual jobs' characteristics. We show that the GPU DaCe possesses a larger flexibility for delivering primary responses compared to the CPU DaCe. Furthermore, the GPU DaCe exhibits lower variability in flexibility across different times of the day and over a 7-month evaluation horizon, making them more dependable and stable sources for offering primary response.**

***Index Terms*—Large Language Model (LLM), GPU Data Center (DaCe), Flexibility, Primary Response**

## I. Introduction

THE surging development of Large Language Models (LLMs), such as ChatGPT by OpenAI and Bard by Google, has gained significant interest from researchers, industries, and the general public. However, the remarkable success of LLMs comes at the cost of substantial energy consumption. As studied in [1], the training of a GPT-3 model can consume roughly 1,300 MWh of electricity. It is expected that the latest iteration, GPT-4, will have an even greater number of parameters than GPT-3, resulting in higher energy demand.

The high energy consumption not only results in substantial energy bills for Data Center (DaCe) owners but also contributes to high power demand, which poses challenges for maintaining the demand-supply balance in power systems. DaCes have flexibility when scheduling the computing tasks, allowing for adjustments in the power demand. This scheduling flexibility has already found application in industrial production to maximize resource utilization rates, such as Borg at Google DaCes [2]. Ref. [3] proposed an online resource management system to enhance the throughput of a DaCe while adhering to a specified power budget. From the perspective of the power system, [4] demonstrated that DaCe's spatial flexibility can help mitigate space-time price volatility. Additionally, [5] illustrated how DaCe's temporal flexibility can be leveraged to reduce carbon emissions and energy costs while maintaining service quality. Moreover, [6] showed that DaCes can provide ancillary services to the power grid, thus helping stabilize the power system, especially in the context of the growing trend of fluctuant renewables in the future.

However, the aforementioned studies focused on traditional CPU DaCe for scientific computation or web service processing. With the rapid development of AI and particularly LLMs, GPU DaCes are now an increasingly large share of the world of DaCes. A GPU DaCe differs from a CPU DaCe in its higher power consumption of individual computing units. For instance, a state-of-the-art CPU like the AMD EPYC 9654 (96 cores/192 threads) has a rated power of only 360 W, whereas a single NVIDIA H100 SXM GPU can draw 700 W of electric power. Moreover, the current large AI model training often requires a multitude of GPUs [1]. These distinct characteristics of GPU DaCes introduce new challenges and opportunities to the power system. Ref. [7] applied power capping to schedule the GPU DaCe to reduce energy consumption. Ref. [8] applied reinforcement learning to reduce energy costs, enhance GPU utilization, and reduce carbon emissions of a GPU DaCe. However, there is a lack of work examining the coupling between the GPU DaCe and the power system.

This study aims to bridge the gap by providing insights from a power system's perspective. Specifically, our focus is on evaluating and comparing the potential of GPU and CPU DaCes in providing primary responses (including primary reserve and frequency regulation) [9], which are important ancillary services with short duration that fit DaCes constrained by quality of service requirements. The comparison aims to shed light on what new implications the GPU DaCes can have for the power systems. To the best of our knowledge, there is currently no work on GPU DaCes from this power system perspective. Our contributions encompass the following:

1) We provide a first "stab" at assessing the potential of the GPU DaCe in providing primary response.
2) A comparative analysis with CPU DaCes was conducted, revealing that GPU DaCes present distinct implications for the power grid.
3) We performed case studies based on real-world job log datasets of both GPU and CPU DaCes.

Section II provides a high-level description of the problem, which is mathematically modeled in Section III. Section IV presents the case studies, and Section V concludes this paper.

The work was supported by the Engineering Studentship from the University of Edinburgh and was also supported by the UK Engineering and Physical Sciences Research Council (EPSRC) (EP/S031901/1, EP/T028564/1).

## II. Problem Overview

### A. Primary Response

Primary response is a typical ancillary service in power systems [9]. This term encompasses primary reserve and regulating reserve, which help stabilize frequency in response to demand-generation imbalances. After reaching an agreement between the system operator and the service provider, the system operator will call (activate) the previously agreed resources for primary response in need. Each activation of the primary response lasts for a relatively short duration, typically less than 15 minutes [10], [11]. This short duration aligns well with the nature of DaCes, as their primary task is computation and they are less likely to disrupt their jobs for an extended period to provide power system services.

### B. Data Center Operation for Primary Response

As discussed, there have been scheduling systems in real-world GPU and CPU DaCes that utilise job flexibility to enhance computing efficiency [2], [7], [12]. The scheduling process may involve job preemption and later restoration [2], [12], and/or more granular power adjustment at a hardware level [7]. When a system operator requests primary reserve activation or sends frequency regulation signals, a GPU or CPU DaCe scheduling system can dynamically preempt or restore a certain number of jobs to adjust the DaCe power demand so that it tracks the regulation signals.

This paper aims to provide an initial assessment of the maximum size of flexibility that a GPU DaCe can contribute to upward primary response (reducing demand) allowing for minor job completion delays, and to assess what distinguishes GPU DaCes from traditionally studied CPU DaCes.

## III. Data Centre Model

The maximum flexibility that a DaCe can provide per one or several primary response calls can be obtained by solving the following Mixed-Integer Linear Programming (MILP):

$$\max_{x_{j,t},z_{j,t},y_j,n_j^P,p_t,f_t,s_i} \quad \sum_{i\in\mathcal{N}^C} s_i \tag{1a}$$

s.t.

$$x_{j,t}, z_{j,t} \in \{0,1\}, y_j \in \mathbb{Z}, \qquad \forall j \in \mathcal{J},\ t \in \mathcal{T}, \tag{1b}$$

$$x_{j,t} = 0, z_{j,t} = 0, \qquad \forall j \in \mathcal{J},\ t \notin \mathcal{T}_j, \tag{1c}$$

$$z_{j,t} \geq x_{j,t} - x_{j,t+1}, \qquad \forall j \in \mathcal{J},\ t \in \mathcal{T}, \tag{1d}$$

$$n_j^P = \sum_{t\in\mathcal{T}}(z_{j,t}) - 1, \qquad \forall j \in \mathcal{J}, \tag{1e}$$

$$y_j + 1 \geq \frac{m^P}{\Delta t} n_j^P + 0.5 \geq y_j, \quad \forall j \in \mathcal{J}, \tag{1f}$$

$$\sum_{t\in\mathcal{T}} x_{j,t} = d_j + y_j, \qquad \forall j \in \mathcal{J}, \tag{1g}$$

$$\sum_{j\in\mathcal{J}} n_j^R \cdot x_{j,t} \leq N^R, \qquad \forall t \in \mathcal{T}, \tag{1h}$$

$$p_t = a \sum_{j\in\mathcal{J}} n_j^R \cdot x_{j,t} + b, \qquad \forall t \in \mathcal{T}, \tag{1i}$$

$$f_t = p_t^{\text{base}} - p_t, \qquad \forall t \in \mathcal{T}, \tag{1j}$$

$$f_t \geq s_i, \qquad \forall t \in \mathcal{T}_i^C, \forall i \in \mathcal{N}^C, \tag{1k}$$

$$s_i \geq 0, \qquad \forall i \in \mathcal{N}^C \tag{1l}$$

The explanation of (1) is given below, where the objective (1a) is explained at the end.

(1b): The binary variable $x_{j,t}$ represents the status (1 for running) of job $j$ at time-step $t$. We define $\mathcal{J}$ as the index set for all the jobs ($\{1, \cdots, J\}$), and $\mathcal{T}$ as the index set for the optimisation horizon of problem (1) ($\{1, \cdots, T\}$). Job preemption, which involves saving and reloading job statuses, requires additional time especially when the computing job is of a large scale. To model this additional time, we introduce a binary variable $z_{j,t}$ to represent if preemption occurs during a given time-step, and a variable $y_j$ to model the total additional time due to job preemption, which will be described in (1f).

(1c): This equation enforces that job statuses $x_{j,t}$ and preemption statuses $z_{j,t}$ can only be zero outside the available period $\mathcal{T}_j$ of a job. The available period $\mathcal{T}_j = \{t^s, \cdots, t^s_j + \lceil(1+\epsilon)(t^c_j - t^s_j)\rfloor\}$ is determined by the time-step of job submission $t^s_j$ and the time-step of job completion $t^c_j$, with a minor proportion $\epsilon$ of delay to provide flexibility. The symbol $\lceil\cdot\rfloor$ stands for the round function. We consider a minor delay acceptable and will not affect DaCe's service quality.

(1d): This ensures that $z_{j,t}$ is equal to 1 when a job running at the current time-step is not running at the next time-step, indicating preemption. This modelling is akin to the shut-down modelling of electrical generators in unit commitment problems. The boundary condition assumes that $x_{j,T+1} = 0$.

(1e): The decision variable $n_j^P$ is introduced to count the total number of times a job is preempted. We add a "minus one" because $z_{j,t}$ equals 1 for the time-step when the job finishes, which is not considered preemption.

(1f): Each job preemption results in additional time $m^P$ spent on that job due to the saving and reloading process. The total additional time can be represented as $m^P n_j^P$. Since the optimization framework discretizes the timeline into time-steps, this continuous time value needs to be rounded to an integer variable $y_j$ through (1f), representing the number of time-steps. Here, $\Delta t$ is the length of a single time-step.

(1g): This constraint ensures that the total time-steps of job running sum up to the total time-steps $d_j$ required to complete the job, including the additional time $y_j$ due to job preemption.

(1h): Each computing job requires a certain number of resources, denoted as $n_j^R$. (1h) enforces that the total resource requirement does not exceed the total number of available resources in the DaCe ($N^R$). In this paper, only GPU (resp. CPU) resources are considered for a GPU (resp. CPU) DaCe.

(1i): This equation establishes a linear relationship between the scheduled DaCe power $p_t$ and the total number of running resources $\sum_j^J n_j^R \cdot x_{j,t}$, i.e., GPU (resp., CPU) for GPU DaCe (resp., CPU DaCe). It assumes that the device consumes 100% of its power when a job is running and consumes no power when the job is not running. The coefficient $a$ captures the variable power when the job is running, including GPU, CPU, memory, etc [5]; $b$ captures rather fixed power like lighting.

(1j): Here, the decision variable $f_t$ represents the flexibility in reducing demand, while $p_t^{\text{base}}$ is the baseline power of the

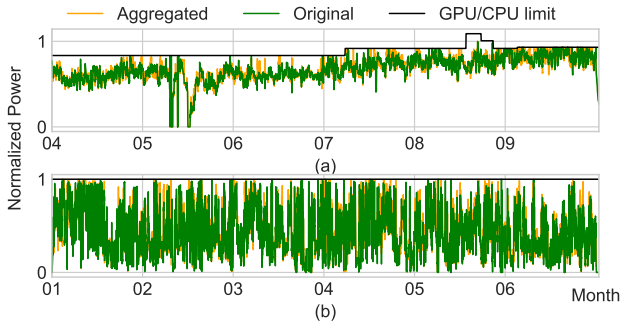


Figure 1. Baseline power before and after job aggregation for both the (a) GPU DaCe and (b) CPU DaCe using real-world datasets. The black lines are the maximum power due to the GPU/CPU limits.

DaCe without providing primary response, which is inferred from data (see Section IV-B) with the power model (1i).

(1k): This constrains that the flexibility must be maintained at a certain level (size) $s_i$ for each of the service calls of primary response with index set $\mathcal{N}^C$, where each service call $i$ spans several time-steps denoted as $\mathcal{T}_i^C$. The size must be nonnegative as specified in (1l).

Finally, the objective (1a) is to find the maximum size of flexibility that a DaCe can provide for one or several times of primary response services with indices in $\mathcal{N}^C$, and thus the summation operation is used.

## IV. Simulation Results

### A. Datasets

The GPU DaCe dataset is sourced from the Venus cluster in Helios [13], a GPU-intensive DaCe operated by SenseTime, a large AI company. This DaCe consists of a total of 6,416 GPUs and is primarily used for AI model training and inference. The dataset encompasses job traces recorded from April to September 2020, including around 0.25 million jobs (for the Venus cluster). It provides information on job submission time $t_j^s$, completion time $t_j^c$, and the required number of GPUs $n_j^R$ for each job. Additionally, the dataset includes details about the total number of available GPUs. The First-In, First-Out (FIFO) strategy is currently employed in Helios, and this strategy is used to infer the baseline power consumption $p_t^{\text{base}}$ in (1j).

The CPU DaCe datasets are the 2 million job traces in Titan supercomputer at the Oak Ridge National Laboratory (ORNL) in 2018 [14]. Titan comprises 18,688 CPU nodes and is primarily used for CPU-intensive tasks like molecular-scale physics and climate modeling. Although some GPU jobs are present, 74% of the jobs in the dataset are CPU-only. Similar to the GPU dataset, it contains information on job submission and completion times, as well as the required resources.

### B. Data Processing and Job Aggregation

For the GPU dataset, only jobs requiring GPUs are extracted, and for the CPU dataset, only CPU jobs are selected. The FIFO principle is used to determine the baseline job running status and therein the baseline power profile $p_t^{\text{base}}$ through the power model (1i). The baseline power profiles are normalized by the maximum power and are depicted as green lines in Fig. 1. Fig. 1 shows that the GPU DaCe has a high resource utilization while the CPU DaCe has an overall lower and more fluctuant utilization. The GPU limit in the GPU DaCe is not flat as the DaCe owner adjusted the available resources to fit their computing demand. Note that, for the CPU DaCe dataset we only pick data from January to July, a period with an overall higher utilization to have a more like-for-like comparison with the GPU DaCe.

Given the large number of job logs in the dataset and the focus of this paper on evaluating potential rather than detailed implementation, a job aggregation strategy is employed to simplify the dataset, which then reduces the problem complexity of (1). The submitted jobs for each day are divided into 100 clusters using K-means clustering based on the job submission time $t_j^s$ and completion time $t_j^c$. For each cluster, the jobs are replaced by a single aggregated job trace, with the submit time set to the earliest submit time for jobs in the associated cluster, and the end time set to the latest submit time for jobs in that cluster. The required resources (number of GPUs/CPUs) for each aggregated job trace are calculated based on the summation of the product of the computing time and resource requirement for each job $j$ of the cluster, divided by the computing time $(t^c - t^s)$ of the aggregated job trace. This ensures that both the aggregated job and the individual jobs in that cluster have the same total computing energy. The baseline power profiles of these aggregated job traces are plotted as orange lines in Fig. 1, where we can see the aggregated baseline profiles can still well capture the original profiles. We also tested some numeric metrics to measure the difference between the aggregated and the original profiles: for GPU DaCe, the R2 score is 0.85 while the Mean Absolute Error (MAE) is 0.039. For CPU DaCe, the R2 score is 0.72 and the MAE is 0.087, indicating an acceptable accuracy.

### C. Optimisation Problem Settings

We set the scheduling horizon of (1) to be 10 days with a resolution $\Delta t$ of 15 minutes, i.e., $T = 960$. The time information in the dataset is rounded to the nearest time-step. Note that this resolution may be too low to accurately capture the dynamic frequency tracking in primary response, but it is sufficient to provide a conservative lower bound on primary response. Each primary response call $\mathcal{T}_i^C$ will last one time-step, which is the upper bound of the durations suggested in [10], [11]. For jobs that are partially covered by the scheduling horizon $\mathcal{T}$, we cut them so their required computing time for completion is equal to the number of time-steps within $\mathcal{T}$ under the baseline FIFO profiles. The choice of a long horizon (10 days) is to reduce the impact by cutting jobs. The additional computing time $m^P$ in (1f) by each time of preemption is set to 1.5 minutes for GPU DaCe, an upper bound based on a study in preempting typical deep learning models [15]. We consider CPU jobs to be easier to restore due to the smaller model scale and we set $m^P$=0.5 minutes for CPU DaCe. The maximum delay proportion $\epsilon$ is set to 5% (see the explanation for (1c)). The coefficient $a$ is set to 1 as we will finally compare the normalized results. The fixed

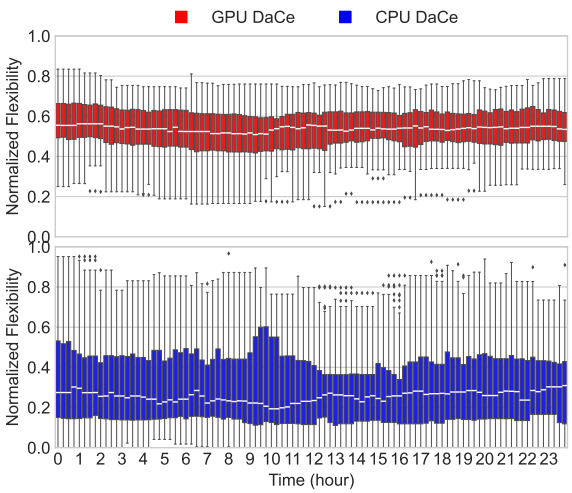


Figure 2. Normalised DaCe flexibility for primary response at each time-step of a day. Each box collects the flexibility sizes obtained by iteratively running the simulation (1) across the 7-month period of the dataset.

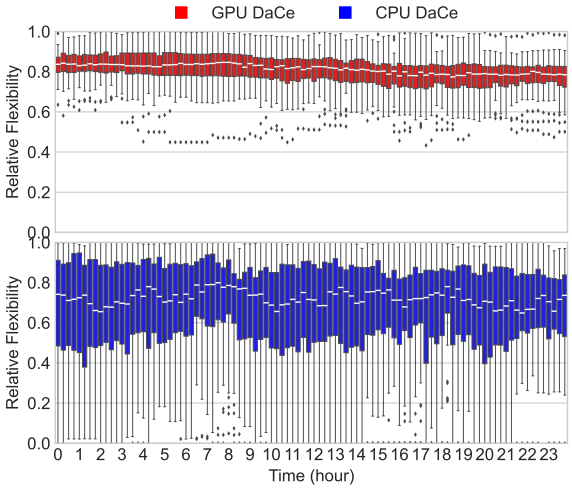


Figure 3. Relative DaCe flexibility for primary response at each time-step of a day. Each box collects the flexibility sizes obtained by iteratively running the simulation (1) over the 7-month period of the dataset.

power $b$ is set to zero as we assume the majority of power consumption varies with the computing demand.

### D. Potential of Primary Response at Each Time-step

Our initial evaluation focuses on the maximum size of flexibility for primary response at each time-step of a day called only once. The results are presented in Fig. 2. To obtain these results, we iteratively ran our simulation (1) across the entire 7-month period of the datasets and created boxplots to collect the outcomes for these simulations at each time-step. The size of flexibility is normalized by the maximum baseline power of the DaCes. Notably, the GPU DaCe has a high flexibility for primary response, whose median is around 60% of its peak power for all time-steps across a day. Additionally, GPU DaCe exhibits a consistently higher level of flexibility for all time-steps than the CPU DaCe. The width of the boxes (indicating variance) of the GPU DaCe is also smaller and more stable than the CPU DaCe, indicating the consistently more stable flexibility of GPU DaCe across the day.

The fluctuant baseline profile of CPU DaCe shown in Fig. 1 may contribute to the smaller size and the higher variance of its flexibility in primary response. To address this concern, we present the results for relative flexibility in Fig. 3. Relative flexibility is defined as the size of flexibility divided by the baseline power at the time-step when the flexibility is called (activated). Similar to the normalized flexibility results in Fig. 2, the relative flexibility results reflect the same conclusion. By observing the median in Fig. 3, we can also conclude that a GPU DaCe can roughly cut 80% of its baseline power for primary response whenever it is called.

### E. Potential of Multiple Calls

In the previous subsection, we assessed the potential for primary response that is called (activated) once a scheduling horizon for different time-steps. However, DaCes are inherently time-coupled by (1g), and the activation of primary response at the current time-step may affect the results for the next call of primary response. The primary response tends to be called several times a day, i.e., size($\mathcal{N}^C$)>1, and the time-spot of each activation is uncertain. To consider this time-coupling effect and the real-world production setting, we assume the service calls are randomly and uniformly distributed: for each scheduling horizon we solve (1) given randomly picked $\mathcal{T}_i^C$ from a uniform distribution. We then iteratively run the problem for the 7 months of the dataset with different activation times $\mathcal{T}_i^C$ and finally evaluate the statistics of the results. Table S2 in [9] suggests that primary response has an activation frequency (determining the size of $\mathcal{N}^C$) between 250 and 15,000 times per year. Based on this, our later analysis sets the size of $\mathcal{N}^C$ to 10, 20, 40, 80, and 160, corresponding to annual frequencies of 365, 730, 1460, 2920, 5840, and 11680 under our 10-day optimization horizon.

Fig. 4 provides the results of flexibility normalized by the maximum baseline power in the left column, and presents the results for relative flexibility in the right column, respectively. The mean values of the flexibility sizes across the 7-month of the dataset by iterative running are given in Fig. 4(a) and (b), along with the associated standard deviations in (c) and (d). As the frequency of primary response increases, the DaCes do have a smaller flexibility on average for primary response. However, the flexibility size is still non-trivial for a large annual frequency: around 13% (resp. 6%) of the maximum power and 20% (resp. 10%) of the baseline power when the primary response is activated under an annual activation frequency of 5,840 (resp. 11,680). Also, the GPU DaCe consistently exhibit a larger size for primary response than the CPU DaCe. Smaller standard deviations for almost all cases suggest that the GPU DaCe is a more stable source for primary response, which is a similar conclusion to Section IV-D.

To gain a better understanding of how the optimization model calculates flexibility, we plot a specific day where the primary response is activated six times in Fig. 5, which shows that GPU DaCe's flexibility is more stable and, on average, larger than that of the CPU DaCe. Interestingly, the CPU DaCe has almost no flexibility for the third primary response call.

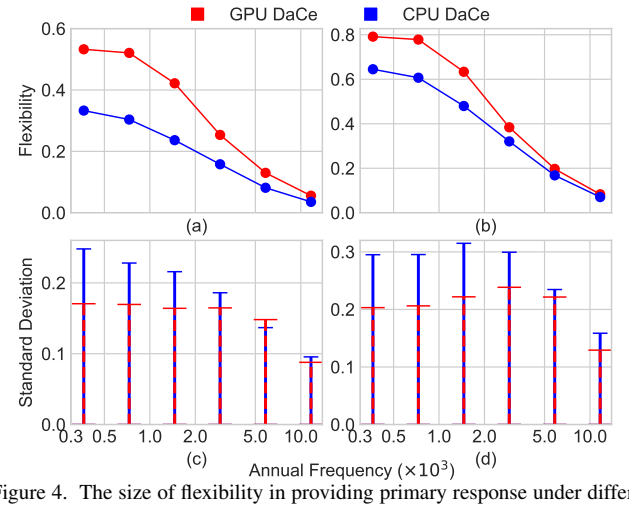


Figure 4. The size of flexibility in providing primary response under different annual frequencies by iteratively solving (1) with randomly selected activation times across the 7-month dataset period. (a): the mean value of flexibility normalised by the maximum baseline power of the DaCe; (b) the mean relative flexibility; (c): the standard deviation (std) for (a); (d): the std for (b).

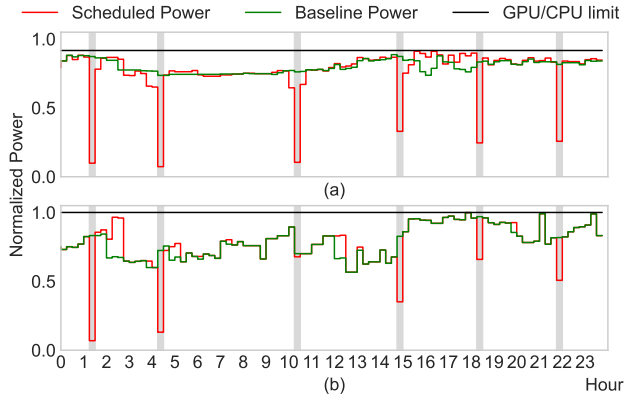


Figure 5. One-day illustration of the scheduled power for primary response v.s. the baseline power for (a) GPU DaCe and (b) CPU DaCe. The black lines are the maximum power due to the GPU/CPU limits. The six times of the activation (the same for GPU/CPU DaCes) are indicated by the grey areas.

## V. Discussion and Conclusion

In summary, this paper conducted an evaluation of the potential for primary response in the GPU DaCe and CPU DaCe, considering a 5% maximum delay in job completion time. We developed a MILP model to calculate the flexibility potential that captures the individual job characteristics. Real-world datasets from SenseTime and ORNL were employed for our case studies. Our findings suggest that GPU DaCes exhibit a larger size of primary response potential compared to CPU DaCes, with a smaller and stable variance across time-steps of a day, unlike wind and solar with intermittent availability. This distinction may originate from the different job characteristics between GPU and CPU jobs: as observed in [13], GPU jobs (for AI purposes) have a high resource utilization rate and have a longer computing time than CPU jobs, so the flexibility is greater under the same proportion of delay in job completion. This larger size and greater stability are beneficial for power system operators as they ensure a reliable and sufficient source of flexibility for future primary response needs. GPU DaCe owners may also earn additional profits by offering primary response services, at a minor cost incurred by slightly delaying their computing jobs. Given the surging demand in GPU DaCe to develop AI applications like LLMs, there can be more collaboration and interaction between power systems and GPU DaCes, leading to win-win results.

This work helps inform power system operators and DaCe operators about the potential for DaCe primary response provision. To realize the primary response engagement of GPU DaCes, effort is still needed in several areas, including: 1) market reforms to ensure financial incentives are in place for DaCe operators to provide flexibility when it is valuable for the system. For example, the UK has introduced a £3/kWh guarantee for the Demand Flexibility Service which is called during peak demand periods [16]; 2) installing high-resolution power monitoring devices in DaCes to track power adjustments; 3) establishing communications between DaCes and power system operators; and 4) upgrading DaCe scheduling systems to enable timely reactions to primary response signals.

## References


[1] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, "Carbon emissions and large neural network training," 2021.

[2] A. Verma *et al.*, "Large-scale cluster management at google with borg," in *Proceedings of the 10th european conference on computer systems*, 2015, pp. 1–17.

[3] O. Sarood, A. Langer, A. Gupta, and L. Kale, "Maximizing throughput of overprovisioned hpc data centers under a strict power budget," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2014, pp. 807–818.

[4] W. Zhang and V. M. Zavala, "Remunerating space–time, load-shifting flexibility from data centers in electricity markets," *Applied Energy*, vol. 326, p. 119930, 2022.

[5] W. Liu *et al.*, "Online job scheduling scheme for low-carbon data center operation: An information and energy nexus perspective," *Applied Energy*, vol. 338, p. 120918, 2023.

[6] K. Kaur *et al.*, "An adaptive grid frequency support mechanism for energy management in cloud data centers," *IEEE Systems Journal*, vol. 14, no. 1, pp. 1195–1205, 2019.

[7] J. McDonald *et al.*, "Great power, great responsibility: Recommendations for reducing energy for training language models," *arXiv preprint arXiv:2205.09646*, 2022.

[8] S. Zhang, M. Xu, W. Y. B. Lim, and D. Niyato, "Sustainable aigc workload scheduling of geo-distributed data centers: A multi-agent reinforcement learning approach," *arXiv:2304.07948*, 2023.

[9] O. Schmidt, S. Melchior, A. Hawkes, and I. Staffell, "Projecting the future levelized cost of electricity storage technologies," *Joule*, vol. 3, no. 1, pp. 81–100, 2019.

[10] B. Zakeri and S. Syri, "Electrical energy storage systems: A comparative life cycle cost analysis," *Renewable and sustainable energy reviews*, vol. 42, pp. 569–596, 2015.

[11] A. A. Akhil *et al.*, "DOE/EPRI electricity storage handbook in collaboration with nreca." Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2015.

[12] "Run:ai," accessed: 2023-10-22. [Online]. Available: https://pages.run.ai/hubfs/PDFs/RunAI-Platform-vs-Kubernetes.pdf

[13] Q. Hu, P. Sun, S. Yan, Y. Wen, and T. Zhang, "Characterization and prediction of deep learning workloads in large-scale gpu datacenters," in *SC:21*, 2021, pp. 1–15.

[14] F. Wang, S. Oral, S. Sen, and N. Imam, "Learning from five-year resource-utilization data of titan system," in *2019 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2019, pp. 1–6.

[15] J. Gu *et al.*, "Tiresias: A GPU cluster manager for distributed deep learning," in *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, 2019, pp. 485–500.

[16] "The ESO's Demand Flexibility Service," https://www.nationalgrideso.com/industry-information/balancing-services/demand-flexibility-service/esos-demand-flexibility-service, accessed: 14-Jan-2024.