AI-focused HPC Data Centers Can Provide More Power Grid Flexibility and at Lower Cost

Yihong Zhou[1], Ángel Paredes[2], Chaimaa Essayeh[3], and Thomas Morstyn[4]

[1]School of Engineering, The University of Edinburgh, U.K., yihong.zhou@ed.ac.uk
[2]Department of Electrical Engineering, University of Málaga, Spain, angelparedes@uma.es
[3]Department of Engineering, Nottingham Trent University, U.K., chaimaa.essayeh@ntu.ac.uk
[4]Department of Engineering Science, University of Oxford, U.K., thomas.morstyn@eng.ox.ac.uk

***Abstract*—The recent growth of Artificial Intelligence (AI), particularly large language models, requires energy-demanding high-performance computing (HPC) data centers, which poses a significant burden on power system capacity. Scheduling data center computing jobs to manage power demand can alleviate network stress with minimal infrastructure investment and contribute to fast time-scale power system balancing. This study, for the first time, comprehensively analyzes the capability and cost of grid flexibility provision by GPU-heavy AI-focused HPC data centers, along with a comparison with CPU-heavy general-purpose HPC data centers traditionally used for scientific computing. A data center flexibility cost model is proposed that accounts for the value of computing. Using real-world computing traces from 7 AI-focused HPC data centers and 7 general-purpose HPC data centers, along with computing prices from 3 cloud platforms, we find that AI-focused HPC data centers can offer greater flexibility at 50% lower cost compared to general-purpose HPC data centers for a range of power system services. By comparing the cost to flexibility market prices, we illustrate the financial profitability of flexibility provision for AI-focused HPC data centers. Finally, our flexibility and cost estimates can be scaled using parameters of other data centers through algebraic operations, avoiding the need for re-optimization.**

I. Introduction

The rapid development of Artificial Intelligence (AI) has attracted significant interest from researchers, industry, policy makers, and the general public. The most representative example is Large Language Models (LLMs) with versatile language understanding, such as ChatGPT. AI is also widely applied in other domains, such as computer vision and health care [1], [2]. To support rapid development and widespread use, there are now high-performance computing (HPC) data centers dedicated to AI [3]. To fully exploit the parallelizability of AI jobs, these AI-focused HPC data centers are accelerator-intensive [3], [4], and the most typical accelerator is the Graphics Processing Unit (GPU). Our paper uses the term "GPU", but it should be noted that other types of accelerator exist, such as the Tensor Processing Unit (TPU) in Google.

Prior to the emergence of AI-focused HPC data centers, general-purpose HPC data centers were used for scientific computing applications, such as simulating complex physical systems and solving large-scale mathematical programming problems. Large-scale general-purpose HPC data centers include those at Oak Ridge National Laboratory (ORNL) [5] and Argonne Leadership Computing Facility (ALCF) [6]. While GPUs are present in these general-purpose HPC data centers, they primarily use Central Processing Unit (CPUs) and many of their jobs are not amenable to GPU acceleration [4], [7].

AI-focused HPC data centers can be more energy-demanding than general-purpose HPC data centers due to their heavy use of energy-demanding GPUs. For example, a latest-generation "192-thread AMD EPYC 9654 CPU" has a rated power of only 360 W, while the latest-generation "NVIDIA B200 GPU" can draw 1000 W. The other contributor is the rapidly expanding use of LLMs, which requires intensive computing [8]. Venture Capital firms invested $290 billion in AI over the last 5 years [9], suggesting substantial growth in the field. Estimated by Electric Power Research Institute (EPRI), data centers now consume 4% of the total power generation annually in the US, and this proportion can increase to as much as 9.1% by 2030, with AI being one of the main drivers [10]. The increased energy demand for AI poses a major challenge for power grid infrastructure [11], especially when considered alongside the ongoing efforts to electrify heating and transportation to achieve net-zero carbon emissions [12]. Being aware of this challenge, AI companies and hardware manufacturers are making their systems more energy efficient [13], [14]. However, Jevons Paradox suggests that increasing energy efficiency may lead to greater demand for computing and higher energy usage [15].

Another avenue is to make data center power demand flexible, that is, adjusting a data center's computing workload and consequently its power demand. This flexibility can be used for power system services that are critical to maintain normal and stable power system operation [16]. For example, Refs. [17] and [18] studied the data center flexibility in maintaining power system frequency. In [19], the data center flexibility was used for mitigating voltage and imbalance issues in distribution networks. Ref. [20] explored the data center flexibility in regulation services, where data centers track power system signals every few seconds. Ref. [21] further studied the use of data center flexibility to participate in energy balancing market. These studies investigated data center flexibility in a wide range of power system services. However, there is a lack of assessment of these diverse power system services within one study, making it difficult to understand the capability of the same data center in providing different power system services. Moreover, these studies did not distinguish

between AI-focused and general-purpose HPC data centers. As mentioned, AI-focused HPC data centers can be more power-demanding due to the heavy use of GPUs, which may lead to different capabilities in providing power system services. In addition, recent work [22] observed that there exist distinct job patterns in the two types of data centers, which can also affect the capability in power system services. Finally, although some work imposed a limit on the disruption of data center computing workload [18], [20], [23], this disruption was not quantified as a cost. Data center operators may still remain disincentivized due to the unclear financial profitability of providing power system services by (potentially) disrupting valuable computing jobs.

To fill these gaps, this paper evaluates the maximum power flexibility of AI-focused HPC data centers compared to general-purpose HPC data centers for a comprehensive set of power system services. In addition, we propose a method for estimating the cost incurred by data centers providing power system services. This cost estimation accounts for the value of computing by using data from three cloud computing platforms (Google Cloud, Amazon AWS, and Oracle). Our analysis uses real-world datasets of 7 AI-focused HPC data centers and 7 general-purpose HPC data centers. These datasets vary between 40 to 80 days, with a temporal resolution of one second or even finer, and have between 13,397 and 962,602 computing jobs. Table I (at the end of the paper) provides a detailed description of these datasets. We find that AI-focused HPC data centers can provide greater power flexibility and at 50% lower cost than general-purpose HPC data centers for a range of power system services, which enables AI-focused data centers to be more competitive for the same market conditions. By comparing flexibility cost and real-world power system service prices, we illustrate the financial profitability of flexibility provision for AI-focused HPC data centers. This may break the stereotype that computing jobs are always more valuable than providing power system services, and brings financial motivations for data center operators to provide power system services. Additionally, a correlation analysis illustrates that data center utilization patterns also contribute to the greater flexibility and lower cost of AI-focused HPC data centers. Finally, we investigate the opportunity for dynamic quotas for parallelizable jobs to increase data center flexibility and reduce flexibility provision cost. The superiority of AI-focused HPC data centers still persists in this opportunity.

An advantage of our novel methodology is that our flexibility and cost estimates can be scaled using parameters of other data centers through algebraic operations, avoiding the need for re-optimization. These scaling formulas can be found in the Methods section. We have developed an interactive Google Colab page [24] which implements this functionality, making our analysis approach accessible to anyone to have quick assessment with their own data center information.

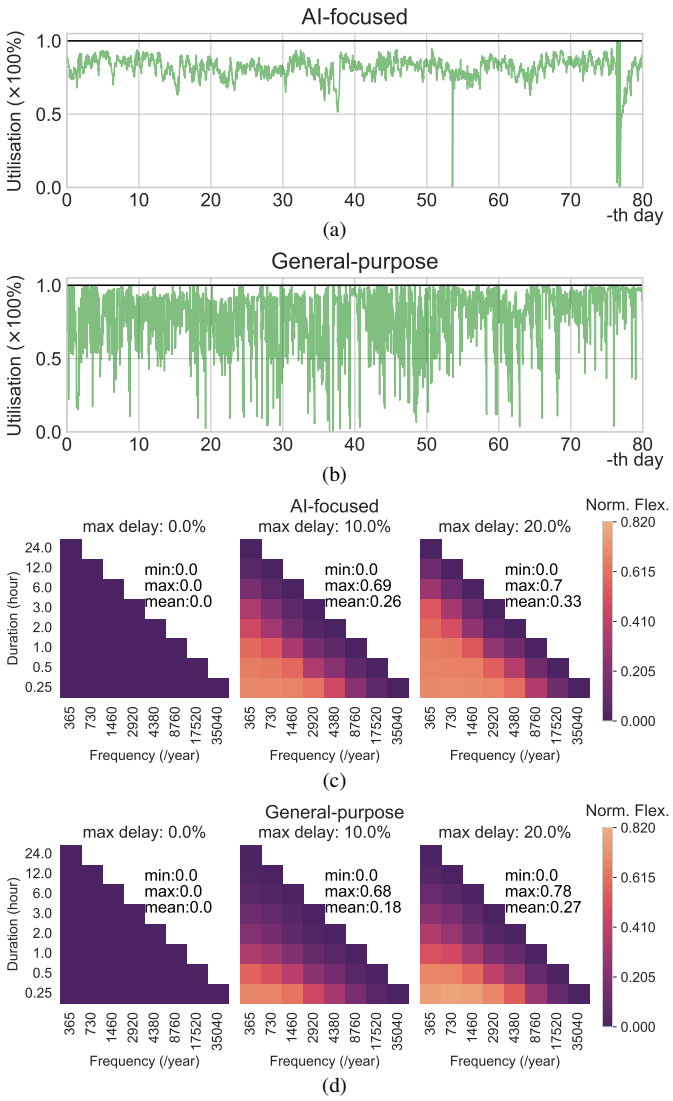

Figure 1. Baseline utilization time series and maximum amount of flexibility for two selected data centers. (a) Utilization of the AI-focused HPC data center (Saturn in Table I). (b) Utilization of the general-purpose HPC data center (ORNL in Table I). The black lines represent the utilization upper bound, while the green lines show the utilization time series.
(c) and (d) are for the maximum amounts of flexibility for power system services with different duration and frequency requirements. (c) Results for the AI-focused HPC data center (Saturn). (d) Results for the general-purpose HPC data center (ORNL). The maximum flexibility amounts are normalized ('Norm. Flex') as ratios of the maximum power of the data center. The heat maps in each column are subject to a specific maximum delay limit (max delay), which is the maximum delay time proportional to the job computing time. "min", "max", and "mean" refer to the minimum, maximum, and mean values across all blocks in each heat map.

II. Results

A. Maximum amount of data center flexibility

We first evaluate the maximum amount of flexibility that data centers can provide for a wide range of power system services. Each of these services can be characterized by duration, frequency, and response time [25], [26]. Frequency indicates how often the power system operator activates the service; duration is the length of time that each activation lasts; and response time refers to how quickly a resource delivers the requested flexibility. For example, primary response and tertiary response are two types of power system services that are often used by grid operators to maintain system frequency within an acceptable range following a demand-supply mismatch. Primary response only requires power delivery to be sustained for a few minutes after activation, but may be

activated 15,000 times a year [26]. On the other hand, tertiary response aims to restore the power system to its normal state, and is rarely activated (approximately 20-50 times a year), but when activated may require continuous power delivery for several hours [26]. It should be noted that although electricity markets will generally have similar power system services, their names, delivery requirements, and detailed procurement mechanisms will vary. This paper focuses only on duration and frequency. Ref. [20] showed that data centers can track power adjustment every few seconds, which is generally sufficient for most power system services.

During the activation periods of each power system service, the data center flexibility is the difference between the baseline power and the power after adjusting the computing workloads. This workload adjustment can be achieved in practice through job preemption [27], [28] and hardware controls [29]–[31] as discussed in the Methods section. We only consider upward flexibility (i.e., demand reduction), because data centers tend to be highly utilized so downward flexibility potential is very limited. Job completion times could be delayed after adjustment, but we have imposed a maximum delay limit as was done in [18], [20], [23]. In this setting, the maximum flexibility for each power system service is calculated by solving optimization problems for job rescheduling (see Methods). Due to the computational complexity of solving the optimization problems, our analysis starts by comparing the results for one general-purpose HPC data center (ORNL in Table I) and one AI-focused HPC data center (Saturn in Table I). These two data centers have average utilization rates both at 81%, and Figs. 1a and 1b shows their baseline utilization time series. Due to the lack of data for data center electric power, we assume a linear relationship between the data center utilization and electric power. This paper considers CPUs as the resource constraint for general-purpose HPC data centers, while GPUs as that for AI-focused HPC data centers due to their importance in AI jobs and higher power consumption [3]. General-purpose HPC data centers also have some GPU jobs, and their CPU utilization has been considered.

Fig. 1c shows the maximum amounts of flexibility calculated for power system services with different frequency and duration requirements for the AI-focused HPC data center (Saturn), and 1d shows the results for the general-purpose HPC data center (ORNL). The time resolution in our optimization problems is set to 15 minutes for computational tractability. Therefore, Figs. 1c and 1d have considered all possible duration and frequency requirements in this setting. These requirements also cover all the 12 typical power system services listed in Ref. [26]. Comparing Figs. 1c and 1d, it is apparent that the AI-focused HPC data center provides greater flexibility than the general-purpose HPC data center as the service duration increases above 1 hour. This is because the general-purpose HPC data center has more variable baseline utilization (Figs. 1a and 1b), so it is more difficult to sustain demand reduction flexibility for a long time. This difference in utilization patterns is also observed in our analysis for all 14 data centers, as will be illustrated in Fig. 6. Note that, Figs. 1c and 1d show that the general-purpose HPC data center has slightly greater flexibility for short-duration services. In fact, this is because our job partition strategy (see Methods) underestimates the flexibility of the AI-focused HPC data center with more long-computing-time jobs. As will be illustrated in Fig. 8, these two data centers have similar amounts of flexibility for short-duration services with a sufficiently long optimization horizon.

B. *Cost of data center flexibility*

We assume that data centers charge a price for each computing job, as is the case with cloud computing platforms. We further assume that the price will be lower if there is greater delay in job completion, due to reduced timeliness. Thus, the total cost for the data center providing a particular power system service is the sum of the price reduction incurred by the delays across all jobs given optimal job scheduling. We propose a linear cost model such that the price reduction for each job is equal to the product of the original price, the job delay proportion, and a price reduction coefficient. This model is formally defined as Eq. (11) in the Methods section. Note that the proposed cost model is not limited to cloud computing data centers, because other data centers may also implement pricing schemes for internal usage; delays in computing-related projects (e.g., developing systems such as ChatGPT) could similarly be monetized with appropriate financial data and analysis—however, these data are typically only available for cloud computing data centers. Nevertheless, we anticipate that any deviations from our current cost estimates would be minor for the same computing hardware, as such hardware (e.g., GPUs or CPUs) would otherwise not be rentable through cloud computing data centers.

We focus on *average flexibility cost*, which is the total flexibility cost divided by the total shifted energy during the service activation periods in kWh (the product of the flexibility amount, duration, and frequency). If the average cost is less than the price of the power system service, then providing that service is profitable. In our proposed cost model, the average flexibility cost is calculated using optimal job scheduling. Therefore, the cost is calculated by solving an optimization problem which is similar to how we calculated the maximum amount of flexibility. The objective is to minimize the flexibility cost, with constraints specifying the amount of flexibility provision. This optimization problem is solved for a nominal set of data center parameters, including the power of the computing devices, the computing price, and the price reduction coefficient. Thanks to the use of the linear cost model, we can quickly calculate the average flexibility cost under other parameter settings by multiplying the corresponding cost scaling factors (see Methods). The cost scaling factor represents the ratio of the average flexibility cost under a specific data center parameter setting to that under the nominal setting. Furthermore, the cost scaling factor that reflects real-world data centers can be estimated by comparing the price of one machine with another slower and cheaper machine using real-world data, as detailed in Methods. Based on the above, we collect the price, power, and computing speed information of all HPC CPU and GPU computing rental options from three large-scale cloud computing platforms:

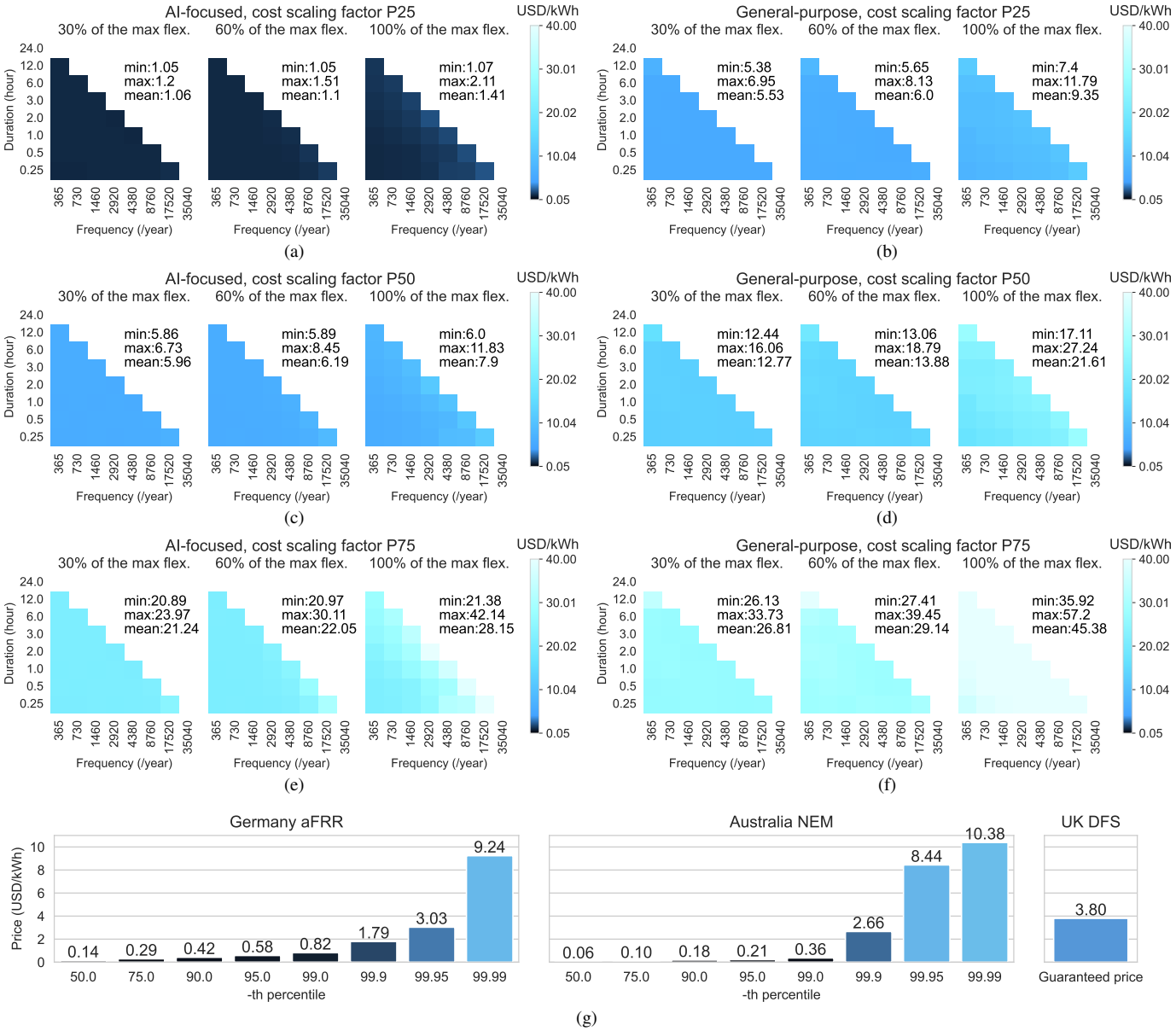

Figure 2. Average flexibility cost and power system service prices. In (a)-(f), each heatmap displays the average cost of providing different percentages of the maximum amounts of flexibility evaluated in Figs. 1c and 1d. The maximum delay limit is set to 20%. "min", "max", and "mean" refer to the minimum, maximum, and mean values across all blocks in each heatmap. Plots (a), (c), and (e) correspond to the AI-focused HPC data center (Saturn) under 25-th (P25), 50-th (P50), and 75-th (P75) percentiles of the cost scaling factor, estimated using data from Google Cloud, AWS, and Oracle. Plots (b), (d), and (f) correspond to the general-purpose HPC data center (ORNL). Plot (g) shows percentiles of real-world power system service prices from Germany's aFRR [32], Australia's NEM [33], and UK's DFS [34].

Google Cloud, Amazon AWS, and Oracle. These data are provided in our supplementary material. We collected 38 samples of the cost scaling factor for general-purpose HPC data centers, and 55 samples for AI-focused HPC data centers. These samples provide an indication of the real-world cost characteristics of data centers.

Figs. 2a–2f show the average flexibility cost for various power system services when data centers provide different percentages of their maximum flexibility (as evaluated in Figs. 1c and 1d). These figures also display results under different percentiles of the cost scaling factor derived from the samples. It can be seen that the AI-focused HPC data center has lower flexibility cost than the general-purpose HPC data center for all power system services and all percentiles of the cost scaling factor. Specifically, when the cost scaling factor is at the median (P50), the AI-focused HPC data center shows a lower flexibility cost of at least 50%. One reason for the lower flexibility cost is the smaller cost scaling factor of AI-focused HPC data centers. Fig. 3 shows the distribution of the cost scaling factor and its components. One interesting observation is that, although GPUs (representing AI-focused HPC data centers) are more expensive than CPUs (representing general-purpose HPC data centers), the GPU power consumption is also large, which leads to a lower price-to-power ratio than CPUs, which in turn brings a lower cost scaling factor to AI-focused HPC data centers.

To evaluate the financial profitability of providing power system services, we collect prices of power system services in

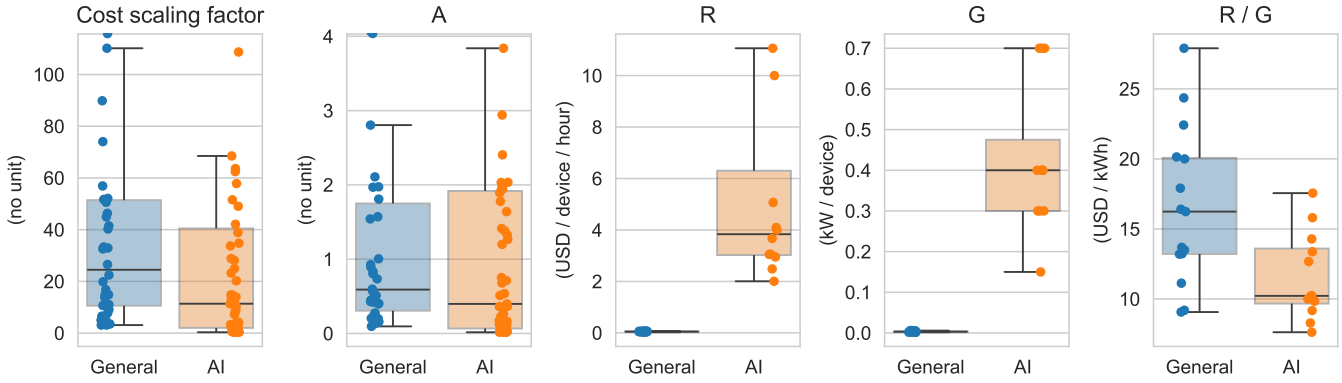

Figure 3. Samples of the cost scaling factor and the elements (A, R, and G) involved in its computation (see Eq. (26)). These samples are derived based on data of computing rental options on Google Cloud, AWS, and Oracle. The 25th, 50th, and 75th percentiles of the cost scaling factor for general-purpose HPC data centers (General) are 10.60, 24,49, and 51.42 respectively. The 25th, 50th, and 75th percentiles of the cost scaling factor for AI-focused HPC data centers (AI) are 2.03, 11.37, and 40.48 respectively.
A is the price reduction coefficient that represents the proportionate price reduction in response to a certain proportion of job delay. R is the hourly price of a single virtual CPU (vCPU, for general-purpose HPCs) or a single GPU (for AI-focused HPCs). G is the power of a single vCPU or a single GPU. These parameters are interpreted in the Methods section. On most cloud platforms, CPUs are rented on the basis of vCPUs, which typically represents one thread of a physical CPU. We follow their convention here. The power of a vCPU (G) is calculated by dividing the rated power of the physical CPU by the number of vCPUs. Note that, in the plots of R, G, and R/G, when a set of computing rental options are essentially renting different portions of the same type of machine, we only keep one option to avoid over-counting the same machine for its power and price parameter.

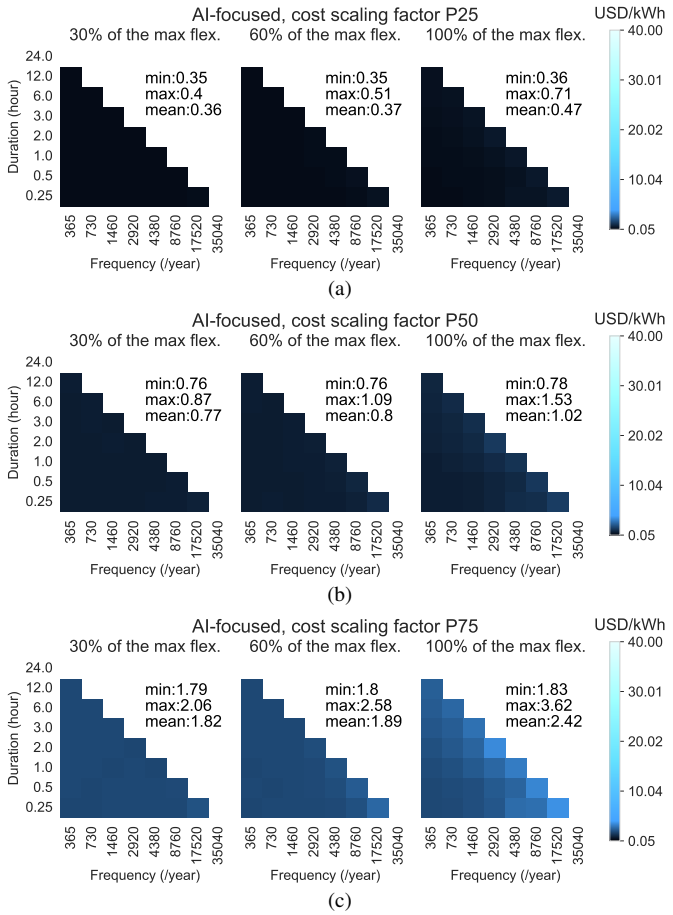

Figure 4. Average flexibility cost under the cost scaling factor derived by Lambda GPU Cloud data [35]. These cost results are for the AI-focused HPC data center (Saturn) when providing different percentages of the maximum amounts of flexibility evaluated in Fig. 1c. The maximum delay limit is set to 20%. "min", "max", and "mean" refer to the minimum, maximum, and mean values across all blocks in each heatmap. (a), (b), and (c) are for 25-th, 50-th, and 75-th percentiles of the cost scaling factor respectively.

the UK, Australia, and Germany. In the winters of 2022/23 and 2023/24, the UK's Demand Flexibility Service (DFS) offered a guaranteed acceptance price at 3 GBP/kWh (around 3.8 USD/kWh) for flexibility during stressful system periods [34]. 34 test events and 4 live events were performed. For Australia, we collect price data from the National Energy Market (NEM) for 2022-2024 [33]. For Germany, we collect 2022-2024 prices of automatic frequency restoration reserve (aFRR), a secondary reserve power system service common among Europe [32]. Fig. 2g shows the percentiles of these prices. Comparing the price and cost results, it can be seen that the AI-focused HPC data center can achieve profitability under the following conditions: 1) when the cost scaling factor is at the 25th percentile and prices are at or above the 99.9th percentiles for aFRR, NEM, and UK DFS; or 2) when the cost scaling factor is at the 50th percentile and prices are at the 99.99th percentile for aFRR and at or above the 99.95th percentile for NEM. Although rare, these price events underscore the high value of certain power system services, which could be profitable for data centers to provide.

When gathering data from various cloud providers, we found that Lambda GPU Cloud [35], which specializes in GPU rentals, offers GPU options approximately 70% cheaper than those from Google Cloud, AWS, and Oracle. The flexibility cost associated with the AI-focused HPC data center is consequently reduced when using a cost scaling factor based on Lambda GPU Cloud data, as shown in Fig. 4. By comparing Fig. 4 and Fig. 2g, it can be seen that the profitability of the AI-focused HPC data center can be achieved even under the 75th percentile of the cost scaling factor. This finding indicates that the profitability varies between data centers with different perceived value in computing. Data centers with lower perceived value in computing might achieve higher profits in providing power system services.

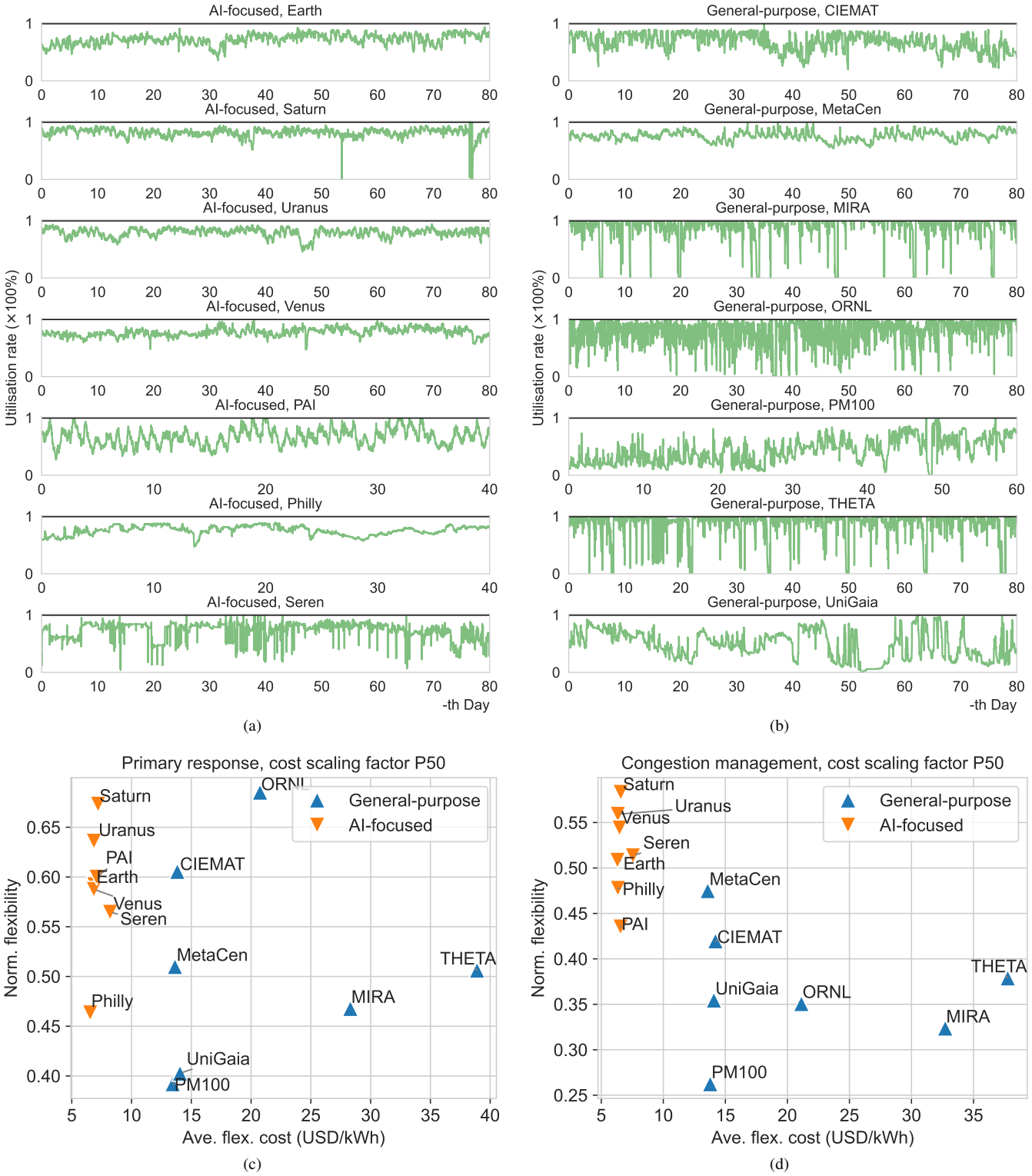

Figure 5. Analysis of all the 14 HPC data centers.
(a) Baseline utilization time series (green lines) of the 7 AI-focused HPC data centers. (b) Baseline utilization time series (green lines) of the 7 general-purpose HPC data centers. The black lines refer to the utilization upper bound.
(c) and (d): The normalized maximum amount of flexibility (Norm. flexibility) versus the average flexibility cost (ave. flex. cost) when providing 100% of the maximum flexibility for two power system services. We use the 50-th percentile (P50) of the cost scaling factor (estimated using data from Google Cloud, AWS, and Oracle). (c) Results for providing primary response (duration of 0.25 hours and frequency of 2920 times/year); (d) Results for providing congestion management (duration of 2 hours and frequency of 365 times/year). Dots closer to the top-left corner indicate better flexibility providers in terms of greater flexibility and lower cost.

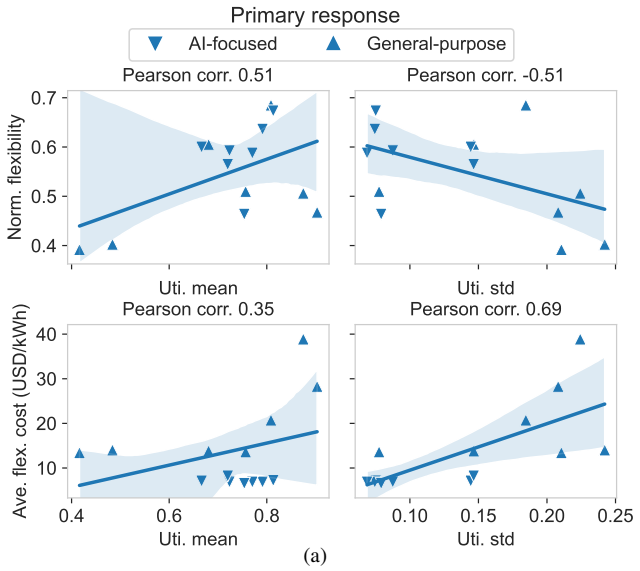

Figure 6. Correlation analysis of the flexibility results and data center utilization patterns. The cost scaling factor is set to the 50th percentile (estimated using data from Google Cloud, AWS, and Oracle). (a) Correlation between the normalized maximum amount of flexibility (Norm. flexibility) for primary response, the average flexibility cost (Ave. flex. cost) for primary response, the mean utilization rate (Uti. mean), and the standard deviation of utilization (Uti. std). Regression lines with confidence intervals (shaded areas) are plotted for highlighting the trends. (b) Correlation results where the flexibility and cost are for congestion management. (c) The mean utilization rate and the standard deviation of utilization for all the 14 data centers.

C. Generalizability of the finding

To generalize the previous findings, this section analyzes the 7 AI-focused HPC data centers and 7 general-purpose HPC data centers in Table I for two typical power system services: One is primary response characterized by short duration and high frequency (0.25 hours and 2920 times a year), while the other is congestion management characterized by long duration and low frequency (2 hours and 365 times a year). It should be noted that these two power system services may have different duration and frequency requirements within a certain range. The specific ranges for these requirements are detailed in [26].

Figs. 5a and 5b display the utilization time series of the 14 HPC data centers. We select 80 consecutive days of job records from the most recent year to ensure up-to-dateness, except a few with only 40 or 60 days after data cleaning. The details of the data processing are given in Methods. Fig. 5c shows the maximum amount of flexibility versus the average flexibility cost when providing primary response, while Fig. 5d shows that for providing congestion management. When providing primary response, it can be seen that AI-focused HPC data centers have lower flexibility cost than all general-purpose HPC data centers. However, it should also be noted that there are two general-purpose HPC data center (ORNL and CIEMAT) that have maximum flexibility approximately equal to or greater than some AI-focused HPC data centers. When providing congestion management, AI-focused HPC data centers are more advantageous: All AI-focused HPC data centers have at least 50% lower flexibility cost than all general-purpose HPC data centers, and only one AI-focused HPC data center (PAI) shows approximately 7% less flexibility than a specific general-purpose HPC data center (MetaCen).

It should be noted that Figs. 5c and 5d are based on the 50th percentile of the cost scaling factor. Our supplementary file provides plots under the 25th and 75th percentiles, and the conclusion of this section remains unchanged.

D. Impact of job patterns on the maximum flexibility and flexibility cost

In addition to the different cost scaling factors illustrated in Fig. 3, the different utilization patterns also contribute to the greater flexibility and lower cost of AI-focused data centers compared to general-purpose data centers. Fig. 6a illustrates the correlation between the normalized maximum amount of flexibility for the primary response service, the average flexibility cost for the primary response service, the mean baseline utilization rate and the standard deviation of baseline utilization; Fig. 6b shows that for the congestion management service. The first column of Figs. 6a and 6b shows a positive correlation between the mean baseline utilization and the normalized maximum amount of flexibility, and between the mean baseline utilization and the average flexibility cost. This is intuitive as a higher mean utilization rate implies a higher baseline power, leading to greater flexibility. At the same time, a highly utilized data center faces more difficulty in restoring jobs disrupted by flexibility provision, resulting in higher cost.

The second column of Figs. 6a and 6b shows a negative correlation between the standard deviation of utilization and the normalized maximum amount of flexibility, which is strong (-0.87) for the congestion management service. This is because data centers with variable utilization time series (indicating variable power demand) are less capable of sustaining flexibility for a long time. We also find a strong positive correlation between the standard deviation of utilization and the average flexibility cost. This is because variant utilization time series can have more frequent utilization spikes, which impede timely job restoration and thus increase flexibility cost.

Finally, Fig. 6c shows the mean utilization and the standard deviation of utilization for all data centers. AI-focused HPC data centers tend to have relatively high utilization rates but low variance, enabling them to provide greater flexibility at lower cost especially for long-duration services such as congestion management, as has been illustrated in Fig. 5d.

Figs. 6a and 6b use the 50th percentile of the cost scaling factor. Our supplementary file provides figures under the 25th and 75th percentiles. The same conclusion can still be derived.

E. Dynamic quota for greater flexibility and lower cost

Certain computing jobs, such as AI training with batch computation, are parallelizable. Therefore, as long as data center computing resources are not always 100% utilized, they can leverage unused resources to speed up computations, reducing job delays and thus flexibility cost. This is called "dynamic quota" as in [36] and is used by data center schedulers [28]. Figs. 7a and 7b show the maximum amounts of flexibility for the two data centers in Figs. 1c and 1d, where 100% extra computing resources can only increase 50% of computing speed. It can be seen that the maximum flexibility increases noticeably compared to Figs. 1c and 1d, and there is now significant flexibility even when no computing delay is allowed (this is in contrast to the results without dynamic quota in Figs. 1c and 1d where there is no flexibility when no delay is allowed).

Figs. 7c and 7d show the flexibility cost with the dynamic quota opportunity. Using extra resources for computational speed-up imposes additional flexibility cost due to higher energy usage, which are calculated based on a wholesale energy market price of 0.05 USD/kWh. Figs. 7c and 7d show that the average flexibility cost can be as low as 0.05 USD/kWh, which is close to grid energy storage [26]. This is because data centers with dynamic quota can provide power system services without incurring job delays.

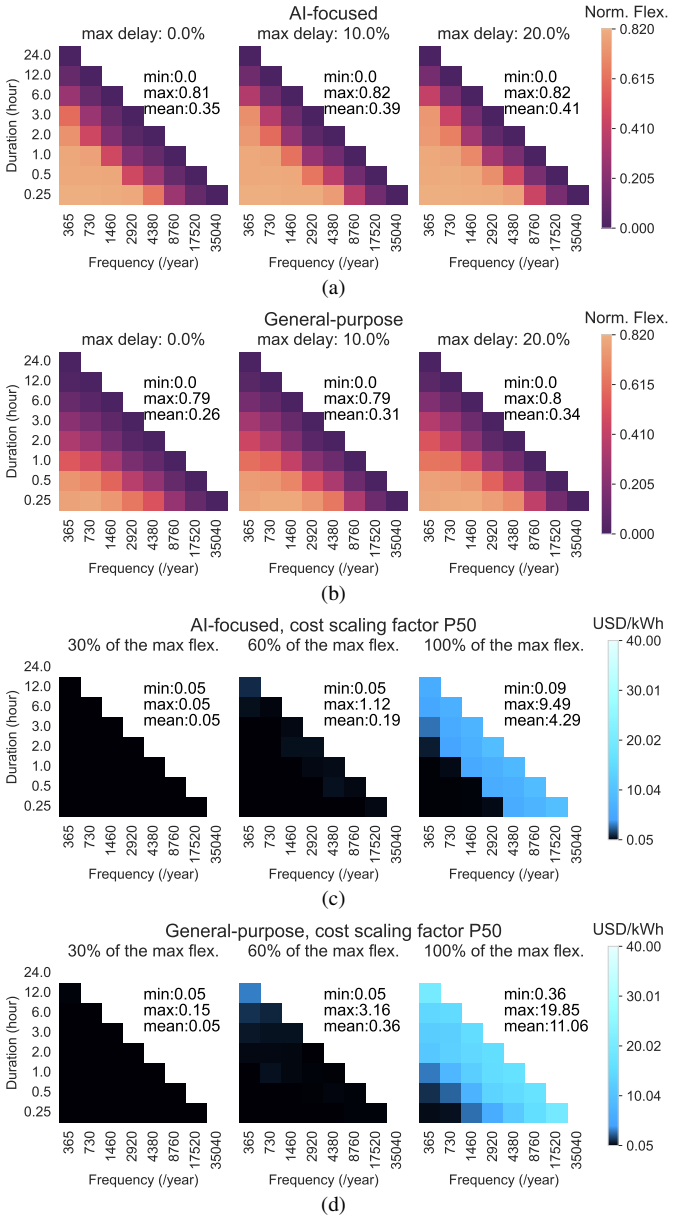

Figure 7. The maximum amount of flexibility and the flexibility cost with the dynamic quota opportunity.
(a) The maximum amounts of flexibility for the AI-focused HPC data center (Saturn). (b) The maximum amounts of flexibility for the general-purpose HPC data center (ORNL).
(c) and (d): The average cost of providing different percentages of the maximum amounts of flexibility evaluated in (a) and (b). (c) The average flexibility cost for the AI-focused HPC data center (Saturn). (d) The average flexibility cost for the general-purpose HPC data center (ORNL). We use the 50th percentile of the cost scaling factor (estimated using data from Google Cloud, AWS, and Oracle).
"min", "max", and "mean" refer to the minimum, maximum, and mean values across all blocks in each subplot. Using 100% extra computing resources is assumed to have 50% computing speed-up.

III. Discussion

This paper evaluated the maximum flexibility that data centers can provide through job scheduling, and proposed a method to estimate the associated flexibility cost. Based on real-world datasets of 14 data centers, we found that AI-focused HPC data centers can provide greater flexibility at lower cost, especially for power system services with longer duration requirements such as congestion management. By comparing the flexibility cost and the real-world power system service prices, we illustrated the financial profitability of data centers providing flexibility. Finally, where the dynamic quota feature is applicable, it can further increase the flexibility of data centers and reduce the associated flexibility cost.

Our findings have implications for several stakeholders. For power system operators who want to manage increasing electricity demand while deferring expensive grid infrastructure upgrades, they can design more targeted collaboration

strategies with data centers based on their differing capabilities for providing specific power system services. For example, AI-focused HPC data centers can be prioritized when securing power system services that require long duration. Our findings also have implications for data center operators, who may believe that they should not disrupt "high-value" computing jobs to provide "low-value" power system services. Our findings show that providing power system services could bring extra net profit to data centers. It is worth noting that the value of flexibility can persist even when data centers are constructed in areas with sufficient connection capacity, because data center flexibility can support the grid by delivering essential system-wide services like reserves and frequency regulation (e.g., aFRR). Finally, for regulators and policymakers, our data center flexibility and cost models can help assess the benefits of integrating data centers into power system flexibility markets.

This paper is a high-level comparative analysis of data center flexibility and cost. An important area for future work is the design of algorithms for real-time job scheduling to coordinate the real-time provision of data center flexibility services. In addition, this paper focuses on the temporal flexibility of data centers, that is, the flexibility of shifting computing workloads in time. Organizations with access to multiple data center sites also have spatial flexibility, which means the ability to shift workloads between locations. Evaluating spatial flexibility would require power flow analysis for the grids where the data centers are located.

IV. Acknowledgments

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) (Project reference EP/S031901/1). Yihong Zhou's work was also supported by the Engineering Studentship from the University of Edinburgh; Ángel Paredes' work was supported by the FPU grant (FPU19/03791) founded by the Spanish Ministry of Education.

V. Methods

This section is divided into several sub-sections for different parts of our results:

1) Section Solving the maximum amount of flexibility describes the method for calculating the maximum amounts of flexibility. Related figures include Figs. 1c, 1d, 5, 6, and 8.
2) Section Data center cost of flexibility details the method for calculating the data center flexibility cost. Related figures include Figs. 2, 3, 4, 5, and 6.
3) Section Dynamic quota describes the method for calculating the maximum flexibility and the cost with the dynamic quota opportunity (Fig. 7).
4) Section Data processing describes the processing of the 14 HPC datasets in Table I.
5) Section Problem settings describes our case study settings.

The following is a nomenclature for this section.

Nomenclature

Functions

$\lvert\cdot\rvert$	The cardinality function that returns the number of elements of a set.
$\lceil\cdot\rfloor$	The round function that returns the nearest integer.

Number sets

$\mathcal{J}$	The index set of computing jobs.
$\mathcal{T}$	The index set of time steps $\{1,\cdots,T\}$ in the optimization horizon.
$\mathcal{T}_j$	The index set collecting the available time steps of job j.
$\mathcal{T}_i^{\mathrm{A}}$	The time steps to sustain flexibility over the i^{th} activation of power system service.
$\mathcal{N}^{\mathrm{A}}$	The set that contains indices i of the activation of power system service.

Parameters

$\delta^{\max}$	The maximum allowable time delay (%) of the job completion, relative to the computing time before providing flexibility.
t_j^{C}	The completion time of job j under the baseline scheduling strategy before providing flexibility.
t_j^{S}	The submission time of job j.
M^{P}	The time overhead triggered by each job preemption.
p_t^{base}	The baseline data center electric power.
N_j^{R}	The number of GPUs (respectively, CPUs) used by job j specified by users in an AI-focused (respectively, general-focus) data center.
$\hat{N}^{\mathrm{R}}$	The total number of GPUs (respectively, CPUs) in an AI-focused (respectively, general-purpose) data center.
D_j	The computing workload for job completion, which is the smallest number of time steps required to complete the job (computing time) under the user-specified number N_j^{R} of computing resources.
G	The increase of power (kW) of an AI-focused (respectively, general-purpose) data center when the utilization of a single GPU (respectively, CPU) increases from 0% to 100%, which can include the contribution from memory, cooling, etc.

G_0 The parameter that captures the fixed power (kW) independent of computing, such as the lighting power or the server idle power, of the whole data center.
A_j/A The price reduction coefficient.
R The hourly computing price of a single GPU/CPU.
π The energy price per kWh of electricity (USD/kWh).
K^{DQ} The speed-up coefficient in the dynamic quota opportunity.
S The amount (kW) of flexibility a data center aims to provide.

Decision variables

$x_{j,t}$ The completed proportion of workload of job j at time step t.
$z_{j,t}$ The preemption counter variable that quantifies the amount of preemption for job j at time step t.
n_j^P The total amount of job preemption of job j.
p_t The scheduled data center electric power.
f_t The flexibility at each time step t in reducing demand.
s_i The amount of the provided flexibility over the i^{th} activation of flexibility.
$x'_{j,t}$ The binary variable that models the job running status at time step t.
δ_j The delay proportion for job j.
e_j The scheduled completion time after flexibility provision.
t^{Ex} The time delay of job j after providing flexibility.
C^{Flex} The total flexibility cost.

A. Solving the maximum amount of flexibility

1) Flexibility maximization problem: The following optimization problem is solved to calculate the maximum amount of flexibility that a data center can deliver for a specific power system service.

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

s.t.

$$x_{j,t}\in[0,1], z_{j,t}\in[0,1], \quad \forall j\in\mathcal{J},\ t\in\mathcal{T}, \tag{1b}$$
$$x_{j,t}=0, z_{j,t}=0, \quad \forall j\in\mathcal{J},\ t\notin\mathcal{T}_j, \tag{1c}$$
$$z_{j,t}\geq x_{j,t}-x_{j,t+1}, \quad \forall j\in\mathcal{J},\ t\in\mathcal{T}, \tag{1d}$$
$$n_j^P=\sum_t^T(z_{j,t})-1, \quad \forall j\in\mathcal{J}, \tag{1e}$$
$$\frac{M^{\mathrm{P}}}{\Delta t}n_j^P\leq\epsilon\cdot D_j, \quad \forall j\in\mathcal{J}, \tag{1f}$$
$$\sum_t^T x_{j,t}=D_j, \quad \forall j\in\mathcal{J}, \tag{1g}$$
$$\sum_j^J N_j^{\mathrm{R}}\cdot x_{j,t}\leq\hat{N}^{\mathrm{R}}, \quad \forall t\in\mathcal{T}, \tag{1h}$$
$$p_t=\sum_j^J G\cdot N_j^{\mathrm{R}}\cdot x_{j,t}+G_0, \quad \forall t\in\mathcal{T}, \tag{1i}$$
$$f_t=p_t^{\mathrm{base}}-p_t, \quad \forall t\in\mathcal{T}, \tag{1j}$$
$$f_t\geq s_i, \quad \forall t\in\mathcal{T}_i^{\mathrm{A}}, \forall i\in\mathcal{N}^{\mathrm{A}}, \tag{1k}$$
$$s_i\geq 0, \quad \forall i\in\mathcal{N}^{\mathrm{A}} \tag{1l}$$

The objective (1a) is to maximize the amount of the power system service that the data center can provide on average over the $|\mathcal{N}^{\mathrm{A}}|$ activation events. Constraints are listed below:

(1b): This constraint defines the value range for two decision variables. The continuous variable $x_{j,t}$ represents the completed workload of job j at time step t. $x_{j,t}=1$ represents the maximum workload achievable using the N_j^{R} user-specified resources for a single time step. We assume that the computing workload per time step can be continuously adjusted. This is feasible under hardware control techniques such as dynamic power capping or DVFS applicable to both GPUs [29], [30], [37] and CPUs [31], [38]. In addition, real-world data center schedulers (such as Run:AI) and the NVIDIA MIG functionality support fractional GPU allocation [39], [40]; CPUs are commonly allocated on a fractional basis in current data centers. Therefore, continuity may be better achieved by combining hardware controls and preemption of jobs with fractional resources. The continuous preemption counter variable $z_{j,t}$ is described in the explanation of (1d).

(1c): This constraint ensures that both $x_{j,t}$ and the preemption counter variable $z_{j,t}$ are zero outside the available period $\mathcal{T}_j$ of job j. The available period is defined as $\mathcal{T}_j \coloneqq \{t_j^S,\cdots,t_j^{\mathrm{C}}+\lceil(1+\delta^{\max})D_j\rfloor\}$, determined by the job submission time t_j^S and the completion time t_j^{C}, incorporating the maximum proportion $\delta^{\max}$ of job delay.

(1d): This constraint models the the preemption counter variable $z_{j,t}$. As explained for (1b), when the completed workload $x_{j,t+1}$ decreases from the previous timestep $x_{j,t}$, it may be due to: 1) hardware control, and/or 2) preemption of jobs that use fractional resources. In the latter case, extra computing workload is required to save and reload checkpoints. $z_{j,t}$ considers the worst-case scenario and counts each workload decrease as preemption. The boundary condition $x_{j,T+1}=0$.

(1e): This constraint introduces n_j^P to count the total amount of preemption for each job. The preemption counter $z_{j,t}$ counts the decrease of $x_{j,t}$ for job completion, which is not considered preemption and is thus subtracted from n_j^P.

(1f): This constraint ensures that the total extra workloads due to preemption should be minor compared to the computing workload D_j for job completion. Here, ϵ is a small positive number. M^{P} (minutes) is the extra computing workload per preemption. Δt is the length (minutes) of a single time step.

(1g): This constraint ensures job completion. Completing job j requires a computing workload D_j, which is the smallest number of time steps (computing time) required to complete the job using the user-specified number (N_j^{R}) of computing resources. As mentioned in Section Maximum amount of data center flexibility (in Results), the computing resources refer to CPUs for general-purpose HPC data centers and GPUs for AI-focused HPC data centers.

(1h): This constraint ensures that the total used resources at each time step do not exceed the total number ($\hat{N}^{\mathrm{R}}$) of resources in the data center.

(1i): This constraint establishes a linear relationship between the data center electric power p_t and the computing workload $x_{j,t}$. The coefficient G in kW represents the increase in power when the utilization of a single unit of computing resource increases from 0% to 100%, which can include the

contribution of memory, cooling, etc. G_0 in kW captures fixed power demand which is independent of computing (e.g. from lighting or idle power).

(1j): This constraint models the demand-reduction flexibility f_t. The baseline power p_t^{base} is a parameter and is inferred from the data center datasets using the same power model (1i).

(1k): This constrains that the flexibility must be sustained at a specific amount s_i for each service activation indexed by $i \in \mathcal{N}^{\text{A}}$, where the i^{th} activation spans several consecutive time steps collected in the set $\mathcal{T}_i^{\text{A}}$. The amount of flexibility s_i must be non-negative as specified in (1l). Different types of power system service have different frequency $|\mathcal{N}^{\text{A}}|$ and duration $|\mathcal{T}_i^{\text{A}}|$ requirements, where $|\cdot|$ calculates the number of elements in the set. For a specific type of power system service, the duration of each activation is the same: $|\mathcal{T}_i^{\text{A}}| = |\mathcal{T}_k^{\text{A}}|, \forall i, k \in \mathcal{N}^{\text{A}}$.

2) Scaling the maximum amounts of flexibility: We solve (1) to get the maximum flexibility under a set of nominal data center parameters. Due to the use of linear power model in (1i), for data centers with other parameter settings, we can directly scale the maximum flexibility under the nominal parameters without time-consuming re-optimization. We use overhead lines $\overline{\blacksquare}$ to specify our nominal parameters. It is worth noting that we must have a specific nominal setting $\overline{G_0} = 0$ for the following derivation. The derived scaling formula can scale the result for arbitrary settings of G_0.

Assuming all decision variables take the values of the optimal solution, the maximum amount S of flexibility by solving (1) can be expressed as:

$$S = ave_{i \in \mathcal{N}^{\text{A}}} \min_{t \in \mathcal{T}_i^A} f_t \tag{2}$$

The operator $\min_{t \in \mathcal{T}_i^A}(\cdot)$ finds the sustained amount of flexibility during the i^{th} activation. Based on Eqs. (1j) and (1i), f_t can be expressed as:

$$\begin{aligned} f_t &= p_t^{\text{base}} - p_t \\ &= \sum_j^J G \cdot N_j^{\text{R}} \cdot x_{j,t}^{\text{base}} + G_0 - \sum_j^J G \cdot N_j^{\text{R}} \cdot x_{j,t} - G_0 \\ &= \sum_j^J G \cdot N_j^{\text{R}} \cdot (x_{j,t}^{\text{base}} - x_{j,t}) \end{aligned} \tag{3}$$

where the superscript $\blacksquare^{\text{base}}$ denotes values corresponding to the baseline utilization of the data center (before flexibility provision). Then, the maximum amount of flexibility can be re-written as:

$$S = ave_{i \in \mathcal{N}^{\text{A}}} \min_{t \in \mathcal{T}_i^A} \left\{ \sum_j^J G \cdot N_j^{\text{R}} \cdot (x_{j,t}^{\text{base}} - x_{j,t}) \right\} \tag{4}$$

Figs. 1c, 1d, 5, 6, 7a, and 7b show the normalized maximum amounts of flexibility, which is the flexibility in kW divided by the data center maximum power. Here, the maximum power $p^{\max}$ is defined as the power when all computing devices are used 100%. In other words:

$$p^{\max} = G \cdot \hat{N}^{\text{R}} + G_0 \tag{5}$$

The normalized flexibility S^{norm} is therefore expressed as:

$$\begin{aligned} S^{\text{norm}} &= \frac{S}{p^{\max}} \\ &= ave_{i \in \mathcal{N}^{\text{A}}} \min_{t \in \mathcal{T}_i^A} \left\{ \frac{\sum_j^J G \cdot N_j^{\text{R}} \cdot (x_{j,t}^{\text{base}} - x_{j,t})}{G \cdot \hat{N}^{\text{R}} + G_0} \right\} \end{aligned} \tag{6}$$

Then, the normalized flexibility under our nominal parameters (recall that $\overline{G_0} = 0$) can be expressed as:

$$\overline{S^{\text{norm}}} = ave_{i \in \mathcal{N}^{\text{A}}} \min_{t \in \mathcal{T}_i^A} \left\{ \frac{\sum_j^J \cdot \overline{N_j^{\text{R}}} \cdot (x_{j,t}^{\text{base}} - x_{j,t})}{\overline{\hat{N}^{\text{R}}}} \right\} \tag{7}$$

Here we assume other data centers have similar job characteristics and the only differences are parameters N_j^{R}, $\hat{N}^{\text{R}}$, G and G_0. In other words, we assume that jobs in other data centers utilize the same proportion of the total data center resources:

$$\frac{\overline{N_j^{\text{R}}}}{N_j^{\text{R}}} = \frac{\overline{\hat{N}^{\text{R}}}}{\hat{N}^{\text{R}}}, \forall j \in \mathcal{J} \tag{8}$$

Combining Eqs. (6), (7), and (8), the normalized amount of flexibility S^{norm} under arbitrary parameters N_j^{R}, $\hat{N}^{\text{R}}$, G and G_0 can be calculated by a linear operation of the normalized flexibility under our parameter setting $\overline{S^{\text{norm}}}$:

$$S^{\text{norm}} = \overline{S^{\text{norm}}} \cdot \frac{G \cdot \hat{N}^{\text{R}}}{G \cdot \hat{N}^{\text{R}} + G_0} \tag{9}$$

By combining (5), the amount of flexibility S in kW after scaling is:

$$S = S^{\text{norm}} \times p^{\max} = \overline{S^{\text{norm}}} \cdot G \cdot \hat{N}^{\text{R}} \tag{10}$$

Eq. (9) suggests that the normalized flexibility results in our figures (such as 1c and 1d) are independent of data center parameters, as long as there is negligible fixed power that is independent of computing (i.e., $G_0 = 0$).

It is important to note that, while the assumption stated in (8) could produce imprecise scaling results if other data centers exhibit different job characteristics, our scaling formula can still help other data centers to quickly evaluate their flexibility by scaling the results calculated in our nominal setting (or via our interactive Google Colab page [24]).

B. Data center cost of flexibility

1) Cost modeling: As explained in Section Cost of data center flexibility (in Results), The total cost for the data center providing a particular power system service is the sum of the price reduction, which is incurred by the delays of jobs. We propose using the following linear model to express the price reduction c_j for each job j:

$$c_j = A_j \cdot \delta_j \cdot V_j \tag{11}$$

where V_j is the original computing price of the job, A_j is the price reduction coefficient, measuring the proportionate price reduction in response to a certain delay proportion δ_j. The delay proportion δ_j is defined as the ratio of the extra time

t_j^{Ex} beyond the original job completion time relative to the computing time D_j:

$$\delta_j = \frac{t_j^{\text{Ex}}}{D_j} \tag{12}$$

In several cloud computing platforms such as Google Cloud, AWS, Oracle, and Lambda Cloud, the computing price V_j is calculated as:

$$V_j = D_j \cdot R \cdot N_j^{\text{R}} \tag{13}$$

where R is the hourly price of a single unit of computing resource and N_j^{R} is the user-specified number of resources. Recall that the computing resources refer to GPUs for AI-focused HPC data centers and refer to CPUs for general-purpose HPC data centers. Also note that most cloud GPU prices will include necessary CPU, memory, and storage.

As explained above, the total flexibility cost C^{Flex} is:

$$C^{\text{Flex}} = \sum_{j \in \mathcal{J}} c_j \tag{14}$$

2) Cost minimization problem: The following problem is solved to find the data center cost of providing S amount of flexibility for a specific power system service.

$$\min_{x_{j,t}, x'_{j,t}, z_{j,t}, n_j^P, p_t, f_t, s_i, e_j, \delta_j, c_j} \quad C^{\text{Flex}} \tag{15a}$$

s.t.

$$(1b) - (1l), \tag{15b}$$

$$x'_{j,t} \geq x_{j,t}, \qquad \forall j \in \mathcal{J},\ t \in \mathcal{T}, \tag{15c}$$

$$e_j \geq t \cdot x'_{j,t} + 1, \qquad \forall t \geq t_j^S + D_j, \forall j \in \mathcal{J}, \tag{15d}$$

$$\delta_j \geq 0, \delta_j \geq \frac{e_j - t_j^S - D_j}{D_j}, \quad \forall j \in \mathcal{J}, \tag{15e}$$

$$c_j \geq A_j \cdot \delta_j \cdot D_j \cdot R \cdot N_j^{\text{R}}, \quad \forall j \in \mathcal{J}, \tag{15f}$$

$$\frac{1}{|\mathcal{N}^{\text{A}}|} \sum_{i \in \mathcal{N}^{\text{A}}} s_i \geq S \tag{15g}$$

Constraint (15c) introduces binary variables $x'_{j,t}$ to indicate the job running status. In (15d), the decision variable e_j models the last time step of job running. Therefore, the term $e_j - t_j^S - D_j$ in (15e) represents the time delay t^{Ex}. Constraint (15f) models the reduction in computing price as defined in (11). Constraint (15g) ensures that the average flexibility over the $|\mathcal{N}^{\text{A}}|$ times of service activation is not less than the specified S.

After solving (15), we calculate the average cost of flexibility ($ACoF$) by dividing C^{Flex} by the total shifted energy over the service activation periods. $ACoF$ is exactly the average flexibility cost in all our cost plots, such as in Fig. 2.

3) Cost minimization problem tightening: The cost minimization problem (15) is a challenging mixed-integer linear programming (MILP). To speed up the computation, we introduce a tightening method by integrating a valid lower bound of the cost into (15). This method assumes that there is no queuing time, meaning submission time equals start time.

Based on Eqs. (11) and (12), the reduction of the computing price for job j can be rewritten as:

$$c_j = A_j \cdot \delta_j \cdot D_j \cdot R \cdot N_j^{\text{R}} = A_j \cdot R \cdot t_j^{\text{Ex}} \cdot N_j^{\text{R}} \tag{16}$$

For subsequent derivation, we need to assume that all jobs have the same price reduction coefficient $A = A_j$. In practice, the value of A can be calculated as a weighted average of the coefficients of A_j for all jobs. Under this assumption, the total flexibility cost C^{Flex} becomes:

$$C^{\text{Flex}} = \sum_{j \in \mathcal{J}} c_j = A\Big(\sum_{j \in \mathcal{J}} t_j^{\text{Ex}} \cdot N_j^{\text{R}}\Big) \cdot R \tag{17}$$

Because t_j^{Ex} is the extra time beyond the original job completion time and N_j^{R} is the number of computing resources, the term $(\sum_{j \in \mathcal{J}} t_j^{\text{Ex}} \cdot N_j^{\text{R}})$ reflects the effort to complete the "total delayed computing workload of all jobs W^{delay}". Furthermore, $(\sum_{j \in \mathcal{J}} t_j^{\text{Ex}} \cdot N_j^{\text{R}})$ is an upper bound of W^{delay}, because jobs over their delayed times t_j^{Ex} may not 100% utilize their computing resources due to the data center capacity limit. In other words:

$$W^{\text{delay}} \leq \Big(\sum_{j \in \mathcal{J}} t_j^{\text{Ex}} \cdot N_j^{\text{R}}\Big) \tag{18}$$

When there is no dynamic quota, providing flexibility will definitely trigger workload delays. Based on our linear power model (1i), the delayed computing workload is proportional to the shifted energy over flexibility provision through the power coefficient G. Now, consider the provision of S units of flexibility for $|\mathcal{N}^{\text{A}}|$ times with each of the flexibility activation sustaining $|\mathcal{T}_i^{\text{A}}|$ time steps, the total shifted energy is $|\mathcal{T}_i^{\text{A}}| \cdot |\mathcal{N}^{\text{A}}| \cdot S$, so the "total delayed computing workload of all jobs W^{delay}" can be expressed as:

$$W^{\text{delay}} = |\mathcal{T}_i^{\text{A}}| \cdot |\mathcal{N}^{\text{A}}| \cdot \frac{S}{G} \tag{19}$$

which, combined with (16) and (18), leads to a lower bound $\underline{C}$ of the total flexibility cost C^{Flex}, namely the total reduction in computing prices:

$$\underline{C} = A \cdot |\mathcal{T}_i^{\text{A}}| \cdot |\mathcal{N}^{\text{A}}| \cdot R \cdot \frac{S}{G} \leq C^{\text{Flex}} = \sum_{j \in \mathcal{J}} c_j \tag{20}$$

In our simulation, setting (20) as an additional constraint in (15) reduces the solution time significantly.

4) Scaling the cost of flexibility provision: We solve (15) to get the minimum flexibility cost C^{Flex} and then the average flexibility cost $ACoF$ under a set of nominal data center parameters. Due to the use of linear cost model, for other data center parameters, we can directly scale the $ACoF$ calculated under the nominal parameters without re-optimization. We use overhead lines $\overline{\blacksquare}$ to specify our nominal parameters.

Again, we assume that all decision variables take the values of the optimal solution. Note that we have assumed $A_j = A$ in Section Cost minimization problem tightening. Then based on Eq. (11), the (minimum) total cost of providing a certain amount S of flexibility can be expressed as:

$$C^{\text{Flex}} = \sum_{j \in \mathcal{J}} c_j = A \cdot \sum_{j \in \mathcal{J}} \delta_j \cdot D_j \cdot R \cdot N_j^{\text{R}} \tag{21}$$

The total shifted energy E^f in kWh over the service activation periods can be expressed as:

$$E^f \coloneqq \Delta t \cdot |\mathcal{N}^{\text{A}}| \cdot |\mathcal{T}_i^{\text{A}}| \cdot S \tag{22}$$

Based on Eq. (10), we further have

$$E^f = \Delta t \cdot |\mathcal{N}^{\text{A}}| \cdot |\mathcal{T}_i^{\text{A}}| \cdot \overline{S^{\text{norm}}} \cdot \hat{N}^{\text{R}} \cdot G \tag{23}$$

Note that, due to the use of (10), we need to set the nominal parameter $\overline{G_0} = 0$ to maintain the initial assumption. Given (21) and (23), we have:

$$ACoF = \frac{C^{\text{Flex}}}{E^f} = \frac{A \cdot \sum_{j \in \mathcal{J}} \delta_j \cdot D_j \cdot R \cdot N_j^{\text{R}}}{\Delta t \cdot |\mathcal{N}^{\text{A}}| \cdot |\mathcal{T}_i^{\text{A}}| \cdot \overline{S^{\text{norm}}} \cdot \hat{N}^{\text{R}} \cdot G} \tag{24}$$

Again, we assume that if (8) holds, the average flexibility cost under our nominal parameter setting can be expressed as:

$$\overline{ACoF} = \frac{\overline{A} \cdot \sum_{j \in \mathcal{J}} \delta_j \cdot D_j \cdot \overline{R} \cdot N_j^{\text{R}}}{\Delta t \cdot |\mathcal{N}^{\text{A}}| \cdot |\mathcal{T}_i^{\text{A}}| \cdot \overline{S^{\text{norm}}} \cdot \hat{N}^{\text{R}} \cdot \overline{G}} \tag{25}$$

Combining (24) and (25), given $\overline{ACoF}$ calculated in our nominal setting, the formula to scale the $ACoF$ according to other data center parameters A, R, and G is:

$$ACoF = \overline{ACoF} \cdot (A \cdot R \cdot \overline{G})/(G \cdot \overline{A} \cdot \overline{R}) \tag{26}$$

The term $(A \cdot R \cdot \overline{G})/(G \cdot \overline{A} \cdot \overline{R})$ is the *cost scaling factor*.

5) Estimating the cost scaling factor: The sections above provide the method for calculating the flexibility cost in a nominal setting (can be set freely with $\overline{G_0} = 0$), and describe how to scale the cost to other parameter settings with the cost scaling factor. However, a question remains of how to estimate a cost scaling factor that allows us to adjust the nominal cost result to reflect the flexibility costs of real-world data centers. Answering this question is the foundation of all our cost plots, such as Figs. 2, 3, 4, 5, and 6. This section describes the estimation method based on real-world cloud platform data.

Recall that our flexibility cost is based on the assumption that slower computing will lead to a lower computing price. Cloud platforms provide several CPU and GPU rental options with different computing speeds and prices, based on which we can estimate the cost scaling factor.

The method is illustrated by the following example. Suppose a cloud platform provides two computing rental options. Option I has N_1^R computing resources, a computing speed of O_1, hourly price per computing resource of R_1, and the power per computing resource is G_1 (in kW), while the parameters for Option II are N_2^R, O_2, R_2, and G_2. Suppose a computing job has computing workload of W. Then, if Option I is rented, the computing time is $D_1 = W/O_1$. Based on (13), the computing price is $V_1 = W/O_1 \cdot R_1 \cdot N_1^R$. Similarly, if Option II is rented, the computing time is $D_2 = W/O_2$, and the price is $V_2 = W/O_2 \cdot R_2 \cdot N_2^R$. Without loss of generality, we assume that Option I is the faster and more expensive option, namely $D_1 < D_2$ and $V_1 > V_2$. Now, if Option I is used to provide power system flexibility service and the computing time is increased to $D_1' = D_2$ (job delay), then the computing price of the delayed Option I should not exceed the price of Option II. In other words, the flexibility cost of Option I is:

$$c_1 = V_1 - V_2 \tag{27}$$

By definition, the delay proportion of Option I for providing flexibility is $\delta_1 = (D_2 - D_1)/D_1$. Then, based on Eq. (11), we can estimate the price reduction coefficient of Option I as:

$$A_1 = c_1/(\delta_1 \cdot V_1) \tag{28}$$

Then, based on (26), we can estimate the corresponding cost scaling factor as $(A \cdot R \cdot \overline{G})/(G \cdot \overline{A} \cdot \overline{R})$.

Through the example above, we see that we can get the cost scaling factor by comparing two computing rental options. Therefore, we collect all GPU and CPU rental options available in Google Cloud, AWS, Oracle, and Lambda Cloud. It should be noted that cloud platforms provide CPUs not just for HPC but also for other applications such as web services. As we focus on HPC data centers, we only select a subset of CPU options on cloud platforms with optimized computing performance. Details of the collection process can be found in our supplementary file. The CPU speed information can be found on the PassMark website [41]. We use the CPU mark as the speed information, which is the weighted harmonic average of the computing speed for several benchmark tests (e.g., integer math, floating point math, physics) that are important for HPC applications [42]. The GPU speed information is available at Lambda Lab [43], which is the weighted average speed for a range of AI tasks. The speed information is provided on both the FP32 and FP16 basis, and we use the harmonic average of the two as the final GPU speed. More than 100 CPU/GPU options with the complete desired information are recorded. We set G as the device rated power. Note that, as the utilization of a device (GPU or CPU) increases from 0 to 100%, the power consumption may be lower than the device rated power due to the idle power, leading to an overestimation of G. However, there are also power increases in memory and cooling that could approximately counterbalance the overestimation of G. Finally, on most cloud platforms, CPUs are rented on the basis of virtual CPUs (vCPUs), which typically represent one thread of a physical CPU. The power of a vCPU is calculated by dividing the rated power of the physical CPU by the number of vCPUs.

With more than 100 CPU/GPU options from these cloud service providers, for either CPU (represents general-purpose data centers) or GPU (represents AI-focused data centers), we compare any two options within the same provider to estimate the corresponding cost scaling factor, as illustrated in the previous example. We avoid comparing options across different platforms because their additional services also affect the pricing. We do not compare the computing options if one would require 100% more computing time than the other. This is to exclude options that are excessively slow and therefore less comparable. In addition, there are comparisons where one option is slower and more expensive than the other, and these comparisons are dropped. These "slower and more expensive" options exist because (potentially) there is a shortage in other computing resources or they have larger memory or storage spaces.

Note that, as mentioned in the main content, the Lambda Cloud GPU prices are significantly lower than the other three cloud platforms. As in Fig. 4, the analysis of Lambda Cloud's data has been separated from the other three cloud platforms.

C. Dynamic quota

1) Solving the maximum flexibility under dynamic quota: To model the dynamic quota functionality, we introduce another set of continuous variables $x_{j,t}^{\text{DQ}} \in [0, 1]$ and a parameter

$K^{\mathrm{DQ}} \in [0,1]$ that represents the speed-up coefficient. In other words, using 100% more computing resources will complete K^{DQ} more computing workload in a single time step. The problems to find the maximum amount of flexibility with dynamic quota have the following set of updated constraints:

- (1g) is updated to

$$\sum_{t}^{T}(x_{j,t} + K^{\mathrm{DQ}} x_{j,t}^{\mathrm{DQ}}) = D_j \tag{29}$$

- (1h) is updated to

$$\sum_{j}^{J} N_j^{\mathrm{R}} \cdot (x_{j,t} + x_{j,t}^{\mathrm{DQ}}) \le \hat{N}^{\mathrm{R}} \tag{30}$$

- (1i) is updated to

$$p_t = \sum_{j}^{J} G \cdot N_j^{\mathrm{R}} \cdot (x_{j,t} + x_{j,t}^{\mathrm{DQ}}) + G_0 \tag{31}$$

- Adding more computing resources may not always result in speed-up due to communication delays across devices. To avoid overestimating the capability of dynamic quota, we limit the number of additional resources to not exceed the originally specified resources through:

$$x_{j,t}^{\mathrm{DQ}} \le x_{j,t} \tag{32}$$

2) Calculating the flexibility cost under dynamic quota: As extra computing resources typically do not achieve a full 100% speed-up, activating the dynamic quota would result in additional energy costs, alongside the flexibility cost triggered by the reduction in computing prices as defined in (14). The extra energy cost can be expressed as:

$$C^{\mathrm{E}} = \pi \Delta t (\sum_{t\in\mathcal{T}} p_t - \sum_{t\in\mathcal{T}} p_t^{\mathrm{base}}) \tag{33}$$

where π is the energy price and $\Delta t(\sum_{t\in\mathcal{T}} p_t - \sum_{t\in\mathcal{T}} p_t^{\mathrm{base}})$ is the additional energy consumption. The total flexibility cost with the dynamic quota opportunity becomes:

$$C^{\mathrm{Flex}} = \sum_{j\in\mathcal{J}} c_j + C^{\mathrm{E}} \tag{34}$$

Eqs. (33) and (34) are included in the minimization problem (15) to calculate the flexibility cost under dynamic quota.

3) Cost minimization problem tightening under dynamic quota: Because job delays could be avoided by using spare resources to accelerate computation, the lower bound in (20) is not valid for dynamic quota, and an updated lower bound is required. Suppose S_a is the maximum flexibility without causing any job delays in the dynamic quota opportunity. In case where delivered flexibility $S_b > S_a$, the new valid lower bound for the total computing price reduction is:

$$\underline{C} = A \cdot |\mathcal{T}_i^{\mathrm{A}}| \cdot |\mathcal{N}^{\mathrm{A}}| \cdot R \cdot \frac{S_b - S_a}{G(1 + K^{\mathrm{DQ}})} \tag{35}$$

To see this, the profile offering the maximum flexibility S_a under zero job delay can be considered a new baseline profile. Then, providing $(S_b - S_a)$ additional flexibility will certainly trigger job delays and thus the reduction in computing prices. Therefore, the lower bound of the total price reduction can be obtained by substituting S with $(S_b - S_a)$ in (20), which is further divided by $1 + K^{\mathrm{DQ}}$ that represents the computing speed-up with dynamic quota.

4) Cost scaling under dynamic quota: With the dynamic quota opportunity, there will be extra energy cost, so the $ACoF$ would additionally include the average extra energy cost of flexibility ($AECoF$). To differentiate, here we refer to the $ACoF$ without dynamic quota (Eq. (26)) as $APCoF$. The $ACoF$ with the dynamic quota opportunity is the sum of $APCoF$ and $AECoF$. The scaling of $APCoF$ still follows (26). As for $AECoF$, first, based on Eq. (1i), the extra energy cost C^{E} in (33) can be re-express as:

$$C^{\mathrm{E}} = \pi \Delta t \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{J}} G \cdot N_j^{\mathrm{R}} \cdot (x_{j,t}^{\mathrm{base}} - x_{j,t}) \tag{36}$$

Combined with Eq. (23), $AECoF$ can be expressed as:

$$AECoF = \frac{C^{\mathrm{E}}}{E^f} = \frac{\pi \Delta t \sum_{t\in\mathcal{T}} \sum_{j}^{J} \cdot N_j^{\mathrm{R}} \cdot (x_{j,t}^{\mathrm{base}} - x_{j,t})}{\Delta t \cdot |\mathcal{N}^{\mathrm{A}}| \cdot |\mathcal{T}_i^{\mathrm{A}}| \cdot \overline{S^{\mathrm{norm}}} \cdot \hat{N}^{\mathrm{R}}} \tag{37}$$

Again we assume that if (8) holds, following a similar process, given the $\overline{AECoF}$ calculated in our nominal parameter $\overline{\pi}$, the formula to scale to other parameter settings is:

$$AECoF = \overline{AECoF} \cdot \frac{\pi}{\overline{\pi}} \tag{38}$$

Therefore, the scaling formula for $ACoF$ with the dynamic quota opportunity is:

$$\begin{aligned} ACoF &= APCoF + AECoF \\ &= \overline{APCoF} \cdot \frac{A \cdot R \cdot \overline{G}}{G \cdot \overline{A} \cdot \overline{R}} + \overline{AECoF} \cdot \frac{\pi}{\overline{\pi}} \end{aligned} \tag{39}$$

D. Data processing

This section presents data processing of all 14 HPC data center datasets used in our case studies. Some background information is provided. There are three important time steps for each job record: the submission time, start time, and completion time. The submission time is when the user commits the job. The start time is when the submitted job begins running, which may be later than the submission time due to resource unavailability. The completion time is when the job is completed. The difference between the completion time and the start time is the job computing time. Because some of our data center datasets lack job submission times, for consistent comparisons, we set all submission times equal to the start times, resulting in zero queue time. This can further lead to an underestimation of the data center flexibility, since there could have been queue-time flexibility.

1) Data selection: Among the 14 HPC data center datasets, some of them contain job data that span several years. To reduce simulation complexity and ensure up-to-dateness, we selected 80 consecutive days of job records from the most recent year. Note that, the data center recorder might miss jobs that were submitted before or completed after the recording period, which leads to under-utilization at the beginning or the end of the recording period. These under-utilized periods are trimmed accordingly. Some datasets contain fewer than 80 days of records post-trimming, thus only 40 or 60 consecutive days are chosen.

2) Short-duration jobs: Job start times and completion times are rounded to our time resolution Δt. For jobs having zero computing time after rounding, we set their computing times to one time step and rescale the required number of resources to ensure the same computing workload: the product of the computing time and the required number of resources.

3) Job partition: For jobs started before and/or completed after the optimization horizon $\mathcal{T}$, we truncate their computing times to equal the number of time steps of job running within $\mathcal{T}$ under the baseline profiles.

For example, suppose our optimization horizon is 00:00 Day 1 - 24:00 Day 1, if there is a job started at 18:00 Day 0 (6 hours before the horizon) and completed at 2:00 Day 2 (2 hours after the horizon) in the baseline profile, then we set the computing time of this job to 24 hours.

As the second example, considering the same optimization horizon, if there is a job started at 18:00 Day 1 and completed at 2:00 Day 2 (2 hours after the horizon), in this case, this job only needs computation for 24:00D1 - 18:00D1 = 6 hours.

This partition strategy can lead to conservativeness in finding the maximum flexibility, as jobs completed after the optimization horizon will have no ability to delay (as in the two examples above). This conservativeness increases for data centers with more long-computing-time jobs, such as the AI-focused HPC data center, as previously observed in Fig. 1c. This conservativeness will reduce with the increase of the optimization horizon, as illustrated in Fig. 8. Despite of the conservativeness, an advantage of this partition strategy is the feasibility guarantee for the next-horizon problem, since this problem does not carry out more computing workload than that before flexibility provision. This also means that optimization problems under non-overlapping horizons are independent, so these problems for different date ranges can be solved in parallel. One may consider another partition strategy such that jobs only need to complete a portion of the computing workload within the current optimization horizon. However, this may lead to an overestimate of the flexibility cost because some jobs are forced to be delayed unnecessarily.

4) Job aggregation: Some of our 14 HPC datasets contain numerous jobs that are running simultaneously, which leads to complicated optimization problems used to find the maximum amount of flexibility (1) and the minimum flexibility cost (15). Since this paper focuses on high-level evaluation rather than production-level implementation, we use a job aggregation strategy to reduce the computational complexity.

We aggregate daily jobs into 100 groups using K-means clustering based on the start time and completion time, then replace each group with a single aggregated job, resulting in only 100 jobs per day after aggregation. For the aggregated job of each group, the start time and the submission time are the earliest start and submission times of the original jobs in the group, and the completion time is the latest. The computing time of the aggregated job is the difference between the completion and start times. The number of resources (GPUs/CPUs) used for each aggregated job is calculated by summing the products of computing time and the number of resource for each job j in the group, then dividing by the computing time of the aggregated job. This process ensures that the computing workload of the aggregated job is equal to the total workloads of the original jobs in that group.

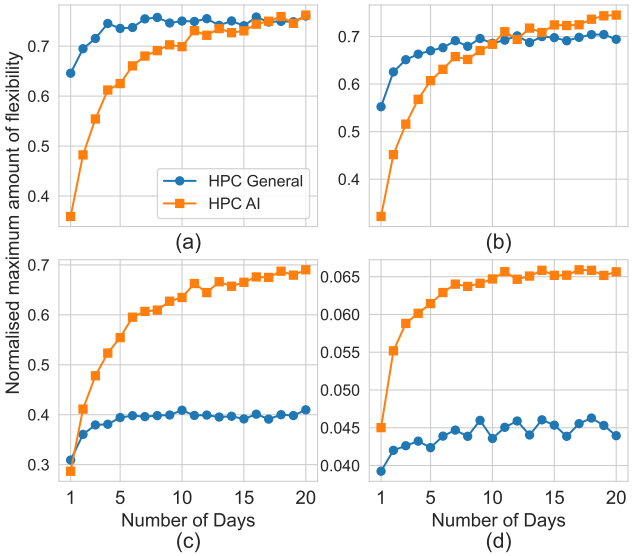

Figure 8. The maximum amounts of flexibility calculated by solving optimization problems with varying horizons (number of days). The calculation is based on the two HPC data centers in Fig. 1. (a) Results for a power system service with duration of 0.25 hours and frequency of 365 times/year. (b) Results when the duration is 0.25 hours and the frequency is 2,920 times per year. (c) Results when the duration is 2 hours and the frequency is 365 times per year. (d) Results when the duration is 2 hours and the frequency is 2,920 times per year. The increasing trend of the maximum amount of flexibility with the increasing optimization horizon confirms our anticipation of reduced conservativeness. This figure shows that the AI-focused HPC data center in Figs. 1c and 1d should have maximum flexibility similar to the general-purpose HPC data center. This holds true even for short-duration, low-frequency services when the optimization horizon increases beyond 15 days.

The aggregation strategy improves computational efficiency while preserving utilization fidelity. Our supplementary file plots the utilization curves based on the aggregated jobs and those based on the original jobs.

E. Problem settings

The optimization horizon is set to 10 days with a resolution Δt of 15 minutes, resulting in $T = 960$. In the 10-day optimization horizon, we randomly pick $|\mathcal{N}^{\mathrm{A}}|$ non-overlapping flexibility activation periods from a uniform distribution, where each period contains $|\mathcal{T}_i^{\mathrm{A}}|$ consecutive time steps. This achieves an annual flexibility frequency of $365/10 \times |\mathcal{N}^{\mathrm{A}}|$ with a duration of $|\mathcal{T}_i^{\mathrm{A}}|$ time steps per activation. We repeat our 10-day optimization for the whole time span of the dataset, and our final results are the average based on these runs.

The overhead M^{P} in (1f) per preemption is 1.5 minutes for AI-focused HPC data centers based on [44], and 0.5 minutes for general-purpose HPC data centers, considering the avoidance of GPU communication compared to AI-focused HPC data centers. The extra workload due to preemption is limited below 1% of the original computing workload, namely $\epsilon = 1\%$ in (1f).

Our nominal parameters are set to $\overline{G} = 1$ kW, $\overline{G_0} = 0$, $\overline{A} = 0.5$, $\overline{R} = 1$ USD per hour per computing resource. The energy price $\overline{\pi}$ is set to the wholesale market price (front-of-

the-meter) of 0.05 USD/kWh [26], considering the excessive power demand of hyperscale data centers.

REFERENCES

[1] D. Demszky *et al.*, "Using large language models in psychology," *Nat. Rev. Psychol.*, vol. 2, no. 11, pp. 688–701, 2023.

[2] K. Singhal *et al.*, "Large language models encode clinical knowledge," *Nature*, vol. 620, no. 7972, pp. 172–180, 2023.

[3] Q. Hu *et al.*, "Characterization of large language model development in the datacenter," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, 2024, pp. 709–729.

[4] Run:AI, "CPU vs. GPU: Key Differences & Uses Explained," https://www.run.ai/guides/multi-gpu/cpu-vs-gpu#:~:text=Faster%20in%20many%20contexts%E2%80%94CPUs,important%20for%20many%20use%20cases., [Accessed 11-06-2024].

[5] "SMC 2021 Data Challenge: Analyzing Resource Utilization and User Behavior on Titan Supercomputer," https://doi.ccs.ornl.gov/ui/doi/334, [Accessed 11-06-2024].

[6] "ALCF Public Data," https://reports.alcf.anl.gov/data/, [Accessed 11-06-2024].

[7] M. Vestias and H. Neto, "Trends of cpu, gpu and fpga for high-performance computing," in *2014 24th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2014, pp. 1–6.

[8] D. Patterson *et al.*, "Carbon emissions and large neural network training," 2021.

[9] World Economic Forum, "How venture capital is investing in AI in the top five global economies — and shaping the AI ecosystem," https://www.weforum.org/agenda/2024/05/these-5-countries-are-leading-the-global-ai-race-heres-how-theyre-doing-it/, [Accessed 15-07-2024].

[10] Electric Power Research Institute (EPRI), "Powering intelligence: Analyzing artificial intelligence and data center energy consumption," , Tech. Rep., 2024. [Online]. Available: https://www.epri.com/research/products/3002028905

[11] D. Mytton, M. Ashtine, S. Wheeler, and D. Wallom, "Stretched grid? Managing data center energy demand and grid capacity," *Oxford Open Energy*, vol. 2, no. October, pp. 1–4, feb 2023.

[12] Y. Zhou, C. Essayeh, S. Darby, and T. Morstyn, "Evaluating the social benefits and network costs of heat-pumps as an energy crisis intervention," *iScience*, 2024.

[13] D. Patterson *et al.*, "The carbon footprint of machine learning training will plateau, then shrink," *Computer*, vol. 55, no. 7, pp. 18–28, 2022.

[14] "NVIDIA Blackwell Platform Arrives to Power a New Era of Computing," https://nvidianews.nvidia.com/news/nvidia-blackwell-platform-arrives-to-power-a-new-era-of-computing, [Accessed 11-06-2024].

[15] S. Sorrell, "Jevons' paradox revisited: The evidence for backfire from improved energy efficiency," *Energy Policy*, vol. 37, no. 4, pp. 1456–1469, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421508007428

[16] Á. Paredes, J. A. Aguado, C. Essayeh, Y. Xia, I. Savelli, and T. Morstyn, "Stacking revenues from flexible ders in multi-scale markets using tri-level optimization," *IEEE Transactions on Power Systems*, vol. 39, no. 2, pp. 3949–3961, 2023.

[17] I. Alaperä, S. Honkapuro, and J. Paananen, "Data centers as a source of dynamic flexibility in smart girds," *Appl. Energy*, vol. 229, no. April, pp. 69–79, 2018.

[18] K. Kaur, S. Garg, N. Kumar, G. S. Aujla, K. K. R. Choo, and M. S. Obaidat, "An Adaptive Grid Frequency Support Mechanism for Energy Management in Cloud Data Centers," *IEEE Syst. J.*, vol. 14, no. 1, pp. 1195–1205, 2020.

[19] S. Chen *et al.*, "Operational flexibility of active distribution networks with the potential from data centers," *Appl. Energy*, vol. 293, no. March, p. 116935, 2021.

[20] 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 Trans. Sustain. Comput.*, vol. 7, no. 1, pp. 157–171, 2022.

[21] Y. Cao, F. Cao, Y. Wang, J. Wang, L. Wu, and Z. Ding, "Managing data center cluster as non-wire alternative: A case in balancing market," *Applied Energy*, vol. 360, p. 122769, 2024.

[22] 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.

[23] 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.

[24] "An Interactive API for Scaling the Estimated DaCe Flexibility and Cost," https://colab.research.google.com/drive/1fkMINZtjUmjEtjBaS7jgH59GsjYsT0ov?usp=sharing, accessed: July 1, 2024.

[25] "Ancillary Services," PJM Interconnection, accessed: January 28, 2024. [Online]. Available: https://www.pjm.com/markets-and-operations/ancillary-services.aspx

[26] 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.

[27] 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.

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

[29] A. Krzywaniak, P. Czarnul, and J. Proficz, "Dynamic gpu power capping with online performance tracing for energy efficient gpu computing using depo tool," *Future Generation Computer Systems*, vol. 145, pp. 396–414, 2023.

[30] B. Dutta, V. Adhinarayanan, and W.-c. Feng, "Gpu power prediction via ensemble machine learning for dvfs space exploration," in *Proceedings of the 15th ACM International Conference on Computing Frontiers*, 2018, pp. 240–243.

[31] A. Krzywaniak, P. Czarnul, and J. Proficz, "Depo: A dynamic energy-performance optimizer tool for automatic power capping for energy efficient high-performance computing," *Software: Practice and Experience*, vol. 52, no. 12, pp. 2598–2634, 2022.

[32] "Imbalance price modules (IP)," https://www.netztransparenz.de/en/Balancing-Capacity/Imbalance-price/Imbalance-price-modules-IP, [Accessed 11-06-2024].

[33] AEMO, "Market Data Nemweb," https://visualisations.aemo.com.au/aemo/nemweb/index.html#mms-data-model, [Accessed 11-06-2024].

[34] "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.

[35] "On-demand GPU cloud pricing," https://lambdalabs.com/service/gpu-cloud, accessed: September 2, 2024.

[36] "Introducing Run:ai's CPU Scheduling: Improved Productivity and Utilization for CPU-only Clusters," Run:AI, https://www.run.ai/blog/introducing-run-ais-cpu-scheduling-improved-productivity-and-utilization-for-cpu-only-clusters#:~:text=Enhanced%20Cluster%20Utilization%3A%20Run%3Aai's,memory%20unit%20is%20leveraged%20effectively., accessed: March 4, 2024.

[37] J. F. D. Guerreiro, "DVFS modeling for energy-efficient GPU computing," Ph.D. dissertation, UNIVERSIDADE DE LISBOA, Jun. 2020.

[38] P. Czarnul, J. Proficz, and A. Krzywaniak, "Energy-aware high-performance computing: survey of state-of-the-art tools, techniques, and environments," *Scientific Programming*, vol. 2019, no. 1, p. 8348791, 2019.

[39] Run:ai, "Allocation of gpu fractions," https://docs.run.ai/v2.13/Researcher/scheduling/fractions/, 2024, accessed on 26 Jan 2024.

[40] NVIDIA, "NVIDIA Multi-Instance GPU: Seven independent instances in a single gpu," https://www.nvidia.com/en-gb/technologies/multi-instance-gpu/, 2024.

[41] PassMark Software, "CPU Benchmarks," https://www.cpubenchmark.net/cpu_list.php, [Accessed 29-08-2024].

[42] ——, "I don't understand the results. What do all these numbers mean?" https://www.passmark.com/support/performancetest_faq/understanding-results.php, [Accessed 29-08-2024].

[43] "Deep Learning GPU Benchmarks," https://lambdalabs.com/gpu-benchmarks, accessed: June 6, 2024.

[44] 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.

[45] "GitHub InternLM/AcmeTrace," https://github.com/InternLM/AcmeTrace, [Accessed 11-06-2024].

[46] M. Jeon, S. Venkataraman, A. Phanishayee, J. Qian, W. Xiao, and F. Yang, "Analysis of large-scale multi-tenant gpu clusters for dnn training workloads," in *2019 USENIX Annual Technical Conference (USENIX ATC 19)*, 2019, pp. 947–960.

[47] "GitHub - msr-fiddle/philly-traces," https://github.com/msr-fiddle/philly-traces, [Accessed 11-06-2024].

[48] Q. Weng *et al.*, "Mlaas in the wild: Workload analysis and scheduling in large-scale heterogeneous gpu clusters," in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 2022, pp. 945–960.

[49] Alibaba, "Alibaba Cluster Trace Program," https://github.com/alibaba/clusterdata, [Accessed 11-06-2024].

[50] "GitHub - S-Lab-System-Group/HeliosData: Helios Traces from SenseTime," https://github.com/S-Lab-System-Group/HeliosData, [Accessed 11-06-2024].

[51] 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.

[52] Wikipedia, "Mira (supercomputer)," https://en.wikipedia.org/wiki/Mira_(supercomputer), [Accessed 11-06-2024].

[53] "Theta/ThetaGPU — Argonne Leadership Computing Facility," https://www.alcf.anl.gov/alcf-resources/theta, [Accessed 11-06-2024].

[54] "Theta Supercomputer Set to Retire: A Look Back at Its Impact on Science at Argonne and Beyond," https://www.hpcwire.com/off-the-wire/theta-supercomputer-set-to-retire-a-look-back-at-its-impact-on-science-at-argonne-and-beyond/, [Accessed 11-06-2024].

[55] CIEMAT, "CIEMAT - Resumen Anual 2022," https://www.ciemat.es/informesanuales/resumen_anual2022.pdf, [Accessed 11-06-2024].

[56] D. G. Feitelson and D. Tsafrir, "Parallel Workloads Archive: Logs," https://www.cs.huji.ac.il/labs/parallel/workload/logs.html, [Accessed 11-06-2024].

[57] "User Statistics," https://www.hpc.cineca.it/our-activities/services/service-statistics/user-statistics/, [Accessed 11-06-2024].

[58] "The Aspen Institute Italia 2024 Award to Research Conducted With Super Computers — Leonardo Pre-exascale Supercomputer," https://leonardo-supercomputer.cineca.eu/the-aspen-institute-italia-2024-award-to-research-conducted-with-super-computers/, [Accessed 11-06-2024].

[59] S. Erotokritou, "PRACE 22nd Call for Proposals Supports New Innovatory Research and Introduces a New Peer-Review Platform," https://prace-ri.eu/prace-22nd-call-for-proposals-supports-new-innovatory-research-and-introduces-a-new-peer-review-platform/, [Accessed 11-06-2024].

[60] "CINECA's Marconi100 Supercomputer Accelerates Drug Discovery at University of Nottingham," https://www.hpcwire.com/off-the-wire/cinecas-marconi100-supercomputer-accelerates-drug-discovery-at-university-of-nottingham/, [Accessed 11-06-2024].

[61] F. Antici, M. Seyedkazemi Ardebili, A. Bartolini, and Z. Kiziltan, "PM100: A Job Power Consumption Dataset of a Large- Scale HPC System," Nov. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.10127767

[62] "Amber - Metacentrum Documentation," https://docs.metacentrum.cz/software/sw-list/amber/, [Accessed 11-06-2024].

[63] Google Cloud, "CPU platforms," https://cloud.google.com/compute/docs/cpu-platforms, [Accessed 02-09-2024].

[64] Amazon Cloud Service, "On-Demand Plans for Amazon EC2," https://aws.amazon.com/ec2/pricing/on-demand/?nc1=h_ls, [Accessed 02-09-2024].

[65] ——, "Amazon EC2 Instance types," https://aws.amazon.com/ec2/instance-types/?nc1=h_ls, [Accessed 02-09-2024].

[66] Oracle, "Configure and estimate costs for OCI services ," https://www.oracle.com/uk/cloud/costestimator.html, [Accessed 02-09-2024].

[67] Oracle Cloud Infrastructure Blog, "Bare Metal vs. Virtual Machines: Which is Best for HPC in the Cloud?" https://blogs.oracle.com/cloud-infrastructure/post/bare-metal-vs-virtual-machines-which-is-best-for-hpc-in-the-cloud, [Accessed 02-09-2024].

Table I
SUMMARY OF THE ANALYZED HPC DATA CENTERS

Name	Type	Main applications	Timespan	Source	System specs	Commercial	Academic
Seren	AI	LLM (training/development), 7B to over 123B [3]	03/2023 to 08/2023	Shanghai AI Lab [45]	8*286 GPUs (A100), 128*286vCPUs, 1024*286 GB RAM	No	Yes
Philly	AI	NN training only, CNN, RNN, LSTM [46]	08/2017 to 12/2017	Microsoft's internal Philly clusters [47]	2490 GPUs	Yes	No
PAI	AI	DL, RL, training and inference [48]	07/2020 to 08/2020	Alibaba [49]	6742 GPUs (T4/P100/V100), 156576vCPUs	Yes	No
Saturn	AI	DL for CV, NLP. and RL. various types of jobs in the DL development pipeline, e.g., data preprocessing, model training, inference, quantization, etc, but the majority is for training [22].	04/2020 to 09/2020	SenseTime [50]	2096 GPUs (Pascal & Volta), 64*262vCPUs, 256*262GB RAM	Yes	No
Earth	AI				1144 GPUs (V100), 48*143vCPUs, 376*143GB RAM,		
Venus	AI				1064 GPUs (V100), 48*133vCPUs, 376*133GB RAM,		
Uranus	AI				2112 GPUs (Pascal), 64*264vCPUs, 256*264GB RAM		
ORNL	General	General scientific computing, like Fluid Dynamics and physics accelerators (recorded in their data files) [5]	01/2015 to 12/2019	ORNL [5]	18688 nodes, 299,008 CPU cores, 18688 Kepler K20X GPUs [51]	No	Yes
MIRA	General	General scientific computing, scientific research, including studies in the fields of material science, climatology, seismology, and computational chemistry [52].	04/2013 to 03/2020	Argonne Leadership Computing Facility [6]	786432 CPU cores	No	Yes
THETA	General	Simulation, data analytics, AI [53], detailed molecular simulations to massive cosmological models [54]	07/2017 to 01/2024		281088 CPU cores for THETA + 24*128 (EPYC 7742) for THETA GPU		
CIEMAT-Euler	General	Data science, AI applied to medicine, dark matter, proton collider analysis and Virgo gravitational waves identification, modelling and simulating of fluid mechanics: combustion, flame dynamics, acoustics [55]	11/2008 to 12/2017	Parallel Workloads Archive [56]	1920 CPU cores (may include GPUs as the application includes AI, but not clearly stated)	No	Yes
PM100	General	Computational Chemistry, Condensed Matter Physics and Computational Fluid Dynamics [57], advanced fluid-dynamic simulations [58], Nuclear Physics [59], drug discovery [60].	05/2020 to 10/2020	Zenodo [61]	980 nodes, 32 CPU cores, 4 V100 GPUs	No	Yes
University of Luxemburg Gaia Cluster	General	It is used mainly by biologists working with large data problems and engineering people working with physical simulations [56].	05/2014 to 08/2014	Parallel Workloads Archive [56]	2004 CPU cores	No	Yes
MetaCentrum 2	General	Computer science, computational chemistry, biomedical computing, bioinformatics, physics, bioinformatics, computer science (middleware technologies), interoperability (research infrastructures) computer science (data platforms) [62].	01/2013 to 12/2014	Parallel Workloads Archive [56]	19 clusters, with 495 nodes and 8412 cores (include some GPU nodes)	No	Yes

Supplementary File: AI-focused Data Centers Can Provide More Grid Flexibility and at Lower Cost

Yihong Zhou[1], Ángel Paredes[2], Chaimaa Essayeh[3], and Thomas Morstyn[4]

[1]School of Engineering, The University of Edinburgh, U.K., yihong.zhou@ed.ac.uk
[2]Department of Electrical Engineering, University of Málaga, Spain, angelparedes@uma.es
[3]Department of Engineering, Nottingham Trent University, U.K., chaimaa.essayeh@ntu.ac.uk
[4]Department of Engineering Science, University of Oxford, U.K., thomas.morstyn@eng.ox.ac.uk

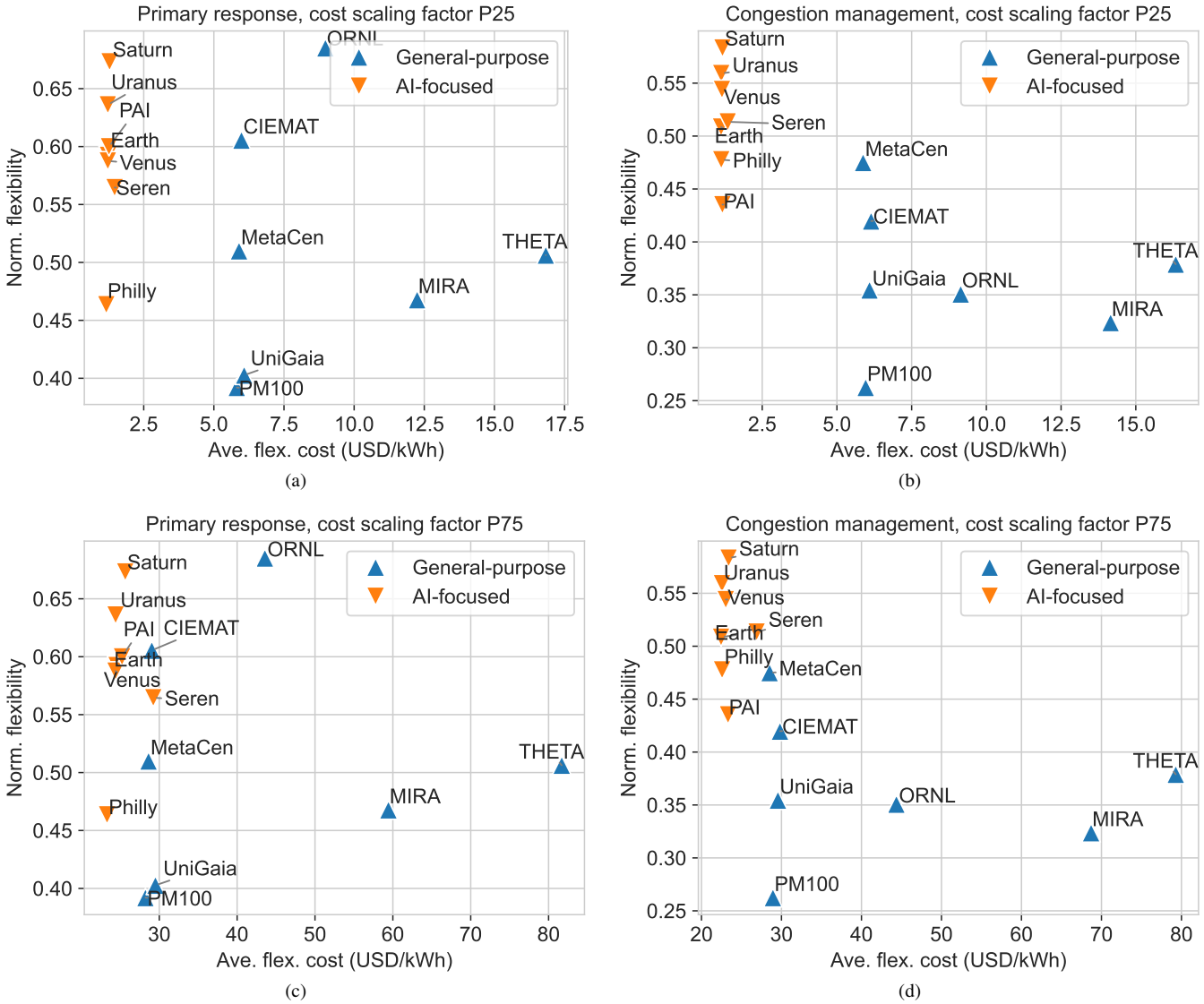

Figure 9. The normalized maximum amount of flexibility (Norm. flexibility) versus the average flexibility cost when providing 100% of the maximum flexibility (ave. flex. cost) for two power system services.
(a) Results for the primary response service under the 25th percentile (P25) of the cost scaling factor. (b) Results for the congestion management service under the 25th percentile (P25) of the cost scaling factor. (c) Results for the primary response service under the 75th percentile (P75) of the cost scaling factor. (d) Results for the congestion management service under the 75th percentile (P75) of the cost scaling factor. The cost scaling factor is estimated using data from Google Cloud, AWS, and Oracle.
This figure leads to the same conclusion as Fig. 5 of our main manuscript.

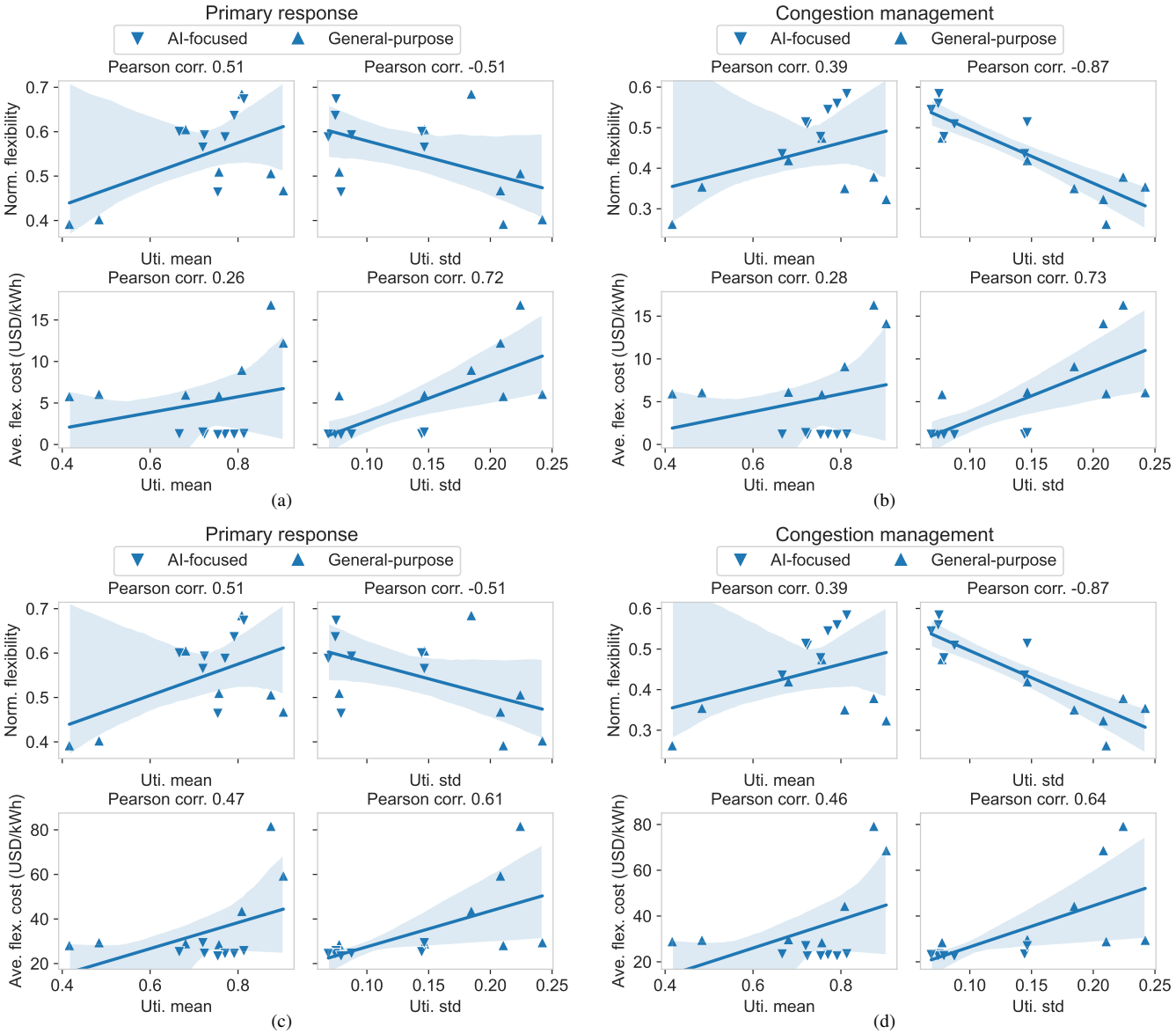

Figure 10. Correlation analysis of the flexibility results and data center utilization patterns. (a) Correlation between the normalized maximum amount of flexibility (Norm. flexibility) for primary response, the average flexibility cost (Ave. flex. cost) for primary response, the mean utilization rate (Uti. mean), and the standard deviation of utilization (Uti. std). Regression lines with confidence intervals (shaded areas) are plotted for highlighting the trends. (b) Correlation results where the flexibility and cost are for congestion management. Plots (a) and (b) are for the 25th percentile of the cost scaling factor. Plots (c) and (d) are for the 75th percentile of the cost scaling factor. The cost scaling factor is estimated using data from Google Cloud, AWS, and Oracle.
This figure leads to the same conclusion, as we have analyzed in Fig. 6 of our main manuscript.

Collecting Cloud Platform CPU and GPU pricing information

A. *Computing prices and model specifications from cloud platforms*

We collect GPU and CPU rental options available on Google Cloud, Amazon Web Services (AWS), Oracle, and Lambda Cloud. These cloud platforms provide price and machine specifications, where the latter enables us to link the price information to device power and computing speed from other sources. Here, we detail the source of each price and specification information. Note that on cloud platforms, they usually use the term "instance" to refer to our "computing option".

- For Google Cloud, up-to-date price information and the GPU model can be found at their instance creation page, which requires a Google account log-in. The CPU model specification can be accessed here [63].
- For AWS, computing prices for all computing options can be found here [64]. Model specifications for both the GPU and CPU options are available here [65].
- For Oracle, the computing price and model specifications are available here [66].
- Lambda Cloud provides computing prices and model specifications here [35]. Note that, as mentioned in our main manuscript, GPU prices on Lambda Cloud are significantly lower than in other cloud platforms. Therefore, we analyzed the Lambda Cloud data separately as Figure 4.

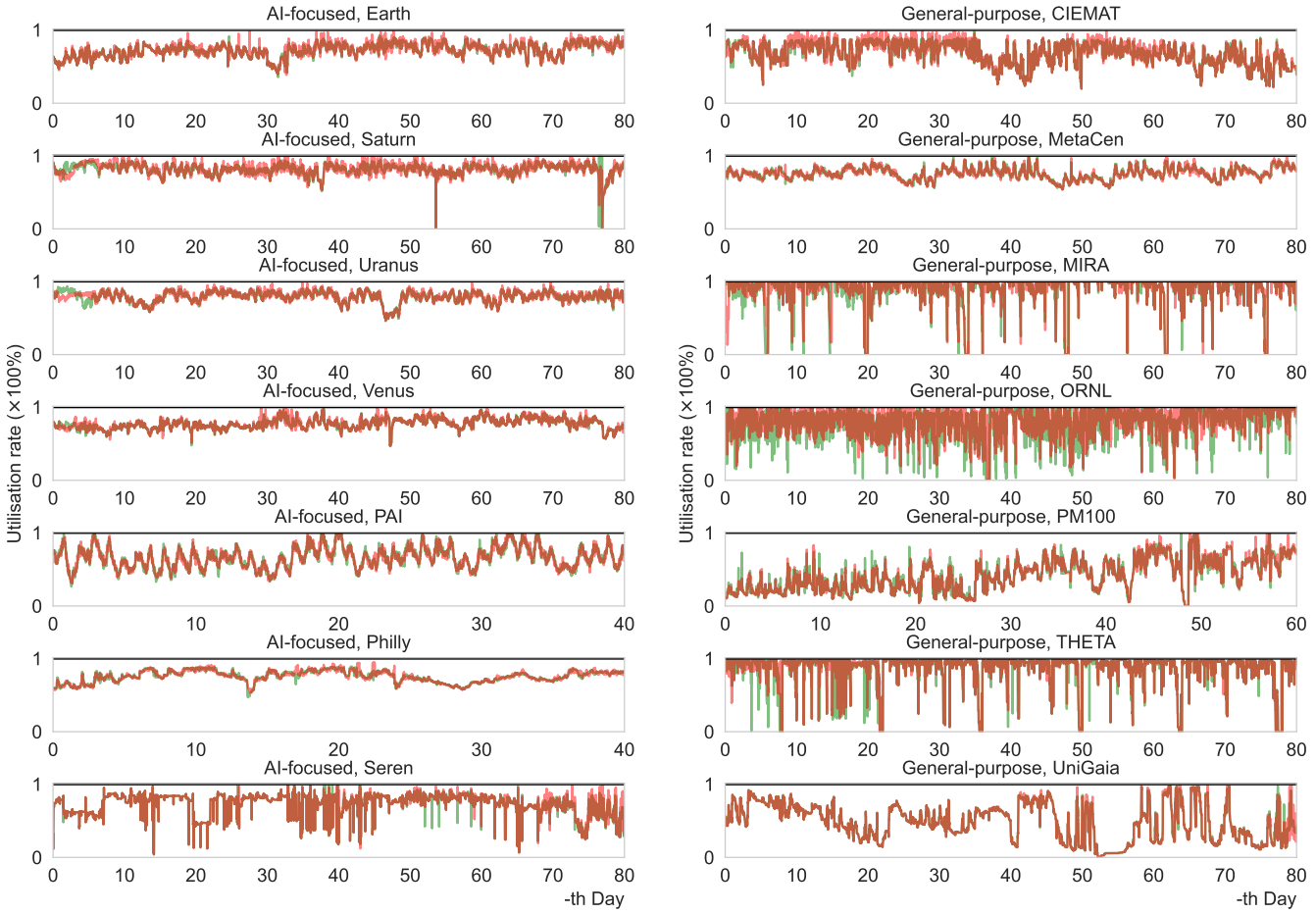

Figure 11. Baseline utilization time series of 14 data centers before and after job aggregation. (a) Utilization of the 7 AI-focused HPC data centers. (b) Utilization of the 7 general-purpose HPC data centers. The black lines refer to the utilization upper bound. The green lines are the utilization derived by the original job datasets. The red lines are the utilization derived by the aggregated jobs after job aggregation, which is a method used to reduce the computation complexity of solving optimization problems, as detailed in the Methods section. Is can be seen that the aggregated baseline profiles effectively capture the characteristics of the original profiles before aggregation (the green lines). Therefore, the applied aggregation strategy aims to improve computational efficiency while preserving utilization fidelity and potential flexibility.

Previous computing price information are for the whole machine. We divide the total price by the number of virtual CPUs (vCPUs) or GPUs to get the price per vCPU or per GPU. This can be justified by the observed proportional relationship between price and number of vCPUs/GPUs on cloud platforms.

On most cloud platforms, CPUs are rented on the basis of vCPUs, which typically represent one thread of a physical CPU. We follow their convention here. Details on the rated power of both CPUs and GPUs are available on the respective manufacturers' websites. The power of a vCPU is calculated by dividing the rated power of the physical CPU by the number of vCPUs.

It should be noted that cloud platforms provide CPUs not just for high-performance computing (HPC) but also for other purposes such as web services. As we focus on HPC data centers, we only collect a subset of CPU options on cloud platforms with optimized computing performance. The collection criteria are summarized below:

- Google Cloud: For CPUs, we only collect the "compute optimized" instances, and several "general" instances with description "consistently high performance". Discounts are not applied. The price region is "US-central-1".
- AWS: We collect "compute optimized" instances and "HPC optmized" instances. We select "on-demand" pricing with Linux operation systems. The price zone is "US East (N. Virginia)".
- Oracle: Oracle does not have dedicated "compute optimized" instances. However, they recommend the "bare metal" type for HPC computing [67]. Therefore, we only select CPU options within that type.
- Lambda Cloud only provides GPU rental, so is not discussed here.

B. Computing speed information

PassMark CPU benchmark test [41] provides multithread CPU ratings, which is used as our CPU speed. This rating is the weighted harmonic average of the computing speed for several benchmark tests, such as integer math, floating point math,

physics, etc., that are important for HPC applications [42].

GPU speed information is available at Lambda Lab [43], which is the weighted average speed for a range of AI tasks. The speed is relative to the speed of a single Tesla V100 GPU, and is provided on both the FP32 and FP16 precision basis. We use the harmonic average of the two precisions as the final GPU speed. We use the 2023 benchmark speed results for up-to-dateness, unless only the 2022 speed information is available for a few GPU options.

C. Data file description

In order to estimate the cost scaling factor (see Methods of the main manuscript), we collect price and speed information from cloud platforms and record them into "all_cloud_data_chart.xlsx" file in the supplementary material. The file has the following columns:

1) **Provider:** Specifies the cloud service provider (e.g., Google Cloud, AWS, Oracle Cloud).
2) **Type:** Indicates whether the entry refers to a CPU (Central Processing Unit) or a GPU (Graphics Processing Unit).
3) **Model:** The specific model of the CPU or GPU used in this computing option.
4) **Number of vCPU / GPU:** Lists the number of vCPUs or GPUs in the computing option. An asterisk (*) indicates that the option is estimated instead of actually available on the cloud platform. For example, a Cloud platform may only rent 4 GPUs or 32 vCPUs as a whole. We can then estimate the price of 1 GPU or 1 vCPU, as long as the corresponding speed information is available. The price is estimated by, e.g., dividing the 4 GPU price by 4. This can be justified by the observed proportional relationship between the price and the number of vCPUs/GPUs on the cloud platforms. This is to enrich our dataset to create more accurate estimate; otherwise, there are only limited computing options which can only support limited estimation samples of the cost scaling factor.
5) **Memory (RAM per CPU, VRAM per GPU):** Indicates the total RAM for CPUs or VRAM per GPU.
6) **Unit Price ($/(unit·h)):** The hourly cost in USD per vCPU or per GPU.
7) **Total device price ($/h):** The total hourly cost for the entire computing option, considering all vCPUs or GPUs.
8) **CPU Score:** The CPU speed for this option, which is only used for general-purpose HPC data centers.
9) **GPU FP32:** The FP32 GPU speed relative to a single "Tesla V100", which is only used for AI-focused HPC data centers.
10) **GPU FP16:** The FP16 GPU speed relative to a single "Tesla V100", which is only used for AI-focused HPC data centers.
11) **Unit Rated Power (W/(vCPU, GPU)):** The power consumption per vCPU or GPU, measured in watts (W).
12) **Total Rated Power (W):** The total power consumption for all vCPUs or GPUs in the computing option.
13) **Notes:** Provides additional information or clarifications about the data, such as assumptions or special conditions.