Applied Energy 324 (2022) 119665

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Data-driven flexibility assessment for internet data center towards periodic batch workloads

Yujie Cao [a], Ming Cheng [b], Sufang Zhang [a], Hongju Mao [b], Peng Wang [a], Chao Li [b], Yihui Feng [b], Zhaohao Ding [a,*]

[a] North China Electric Power University, Beijing 102206, China
[b] Alibaba Group, Hangzhou 311121, China

HIGHLIGHTS

- A data-driven flexibility assessment scheme of internet data center is proposed by investigating the temporal shifting capability of periodic batch workloads.
- The power consumption model of data center is modeled by a data-driven manner based real-world workload processing and energy consumption data.
- The periodic arrival patterns and job dependencies are incorporated into the assessment model to avoid infeasible job scheduling.
- The effectiveness and accuracy of the proposed scheme is illustrated with real-world production workload trace from one of the major cloud service providers.

ARTICLE INFO

Keywords:
Data-driven
Flexibility assessment
Data center
Periodic jobs

ABSTRACT

Considering its unique operational and power consumption characteristics, internet data center (IDC) has been intensively investigated as a promising candidate to provide flexibility for electric power system. In this paper, a data-driven flexibility assessment scheme for IDC is proposed by investigating the temporal shifting capability of periodic batch workloads, which are the major flexibility source in the workload scheduling and execution process. We develop a four-step assessment procedure by identifying the periodic jobs, extracting key operational patterns, mapping the power consumption with workload execution, and quantifying the flexibility associated with power system operation, all of which are established in a data-driven manner. In addition, we adopt real-world production workload trace to verify and demonstrate the effectiveness of the proposed flexibility assessment scheme.

1. Introduction

As the key infrastructure in the information era, internet data center (IDC) has witnessed unprecedented growth in the past decade, introducing numbers of opportunities and challenges in both technology and economic fields. The energy demand of data centers, in particular, is continuously increasing, attracting more and more attention. According to IEA, the power consumption of data centers worldwide is about to reach 270 TWh by 2022, representing around 2 % of global electricity consumption [1]. In China, the power consumption of data centers is projected to be around 78 TWh by 2035, accounting for 4 % of total power consumption in the country [2]. Among all the power-consuming devices in IDCs, information technology (IT) equipment generally contributes the most by providing intensive computation services [3]. Other devices (such as cooling systems, storage systems, and distribution systems) are responsible for assisting the IT equipment in secure and efficient operation, which consequently correlates with the workload execution in IT devices as well.

Workloads of IDCs are application programs performing myriad different computing services submitted by clients [4]. Considering various execution requirements, priorities, and current queues [5], some of the workloads could provide flexibilities in terms of the service level agreements (SLA) [6,7]. As widely accepted in various researches [8–11] and production [12,13], the classification of workloads is described as Table 1 based on [14–16].

Interactive workloads [17] (e.g., web service and online payment) are generally random and low-latency tolerant, which indicates that their power consumption flexibility is typically insignificant. On the contrary, batch workloads [7,18] (e.g., data mining and machine

* Corresponding author.
E-mail address: zhaohao.ding@ncepu.edu.cn (Z. Ding).

https://doi.org/10.1016/j.apenergy.2022.119665
Received 14 November 2021; Received in revised form 4 June 2022; Accepted 8 July 2022
Available online 4 August 2022

Nomenclature

$j, \forall j \in \mathbb{P} \in \mathbb{L}$ Indexes for periodic jobs.
$l, \forall l \in \mathbb{L}$ Indexes for jobs.
$t, \forall t \in T$ Indexes for time slots.
$i, \forall i \in I$ Indexes for fossil-fuel generators.
T_l^{AT}, T_l^{ET} Start time, and end processing time of job l.
T_l^{PT} Processing time duration for job l.
T_j^{ST} Slack time of periodic job j.
T_j^{DDL} Execution deadline of periodic job j.
r_j^{CPU}, r_j^{mem} Average CPU and memory usage of periodic job j.
R_j^{CPU}, R_j^{mem} Overall CPU and memory usage of periodic job j.
$x_{j,t}^{job}$ Binary decision variable indicating whether periodic job j is executed ($x_{j,t}^{job} = 1$) or not ($x_{j,t}^{job} = 0$).
P_j^{job} Power consumption of periodic job j.
$p_t^{\mathbb{P}}$ Power consumption of all periodic job at time t.
p_t^{dc} Power consumption of IDC clusters at time t.
P_t^{other} Power consumption of other load in IDC at time t.
$x_{i,t}^{gen,on}$ Binary decision variable indicating whether fossil-fuel generator i is on or not at time t.
$x_{i,t}^{gen,su}, x_{i,t}^{gen,sd}$ Binary decision variables indicating whether fossil-fuel generator i is start-up or shut-down at time t.
$P_i^{gen,min}, P_i^{gen,max}$ Minimum and Maximum output of fossil-fuel generator i.
$T_i^{gen,MU}, T_i^{gen,MD}$ Ramp-up/down time of fossil-fuel generator i.
$P_i^{gen,ru}, P_i^{gen,rd}$ Ramp-up/down power limitation of fossil-fuel generator i.
$T_i^{gen,on}, T_i^{gen,off}$ Minimum on/off time limitation of fossil-fuel generator i.
p_t^{wind}, P_t^{wind} Integrated and maximum wind power at time t.
p_t^{PV}, P_t^{PV} Integrated and maximum solar power at time t.
D_t^{flex} Flexible demand of IDC clusters at time t.
R_t^{flex} Flexible demand ratio of IDC clusters at time t.
P_t^{ori} Original demand of IDC clusters at time t.

Table 1
Workload Classification and Proportion.

Workload Type		Quantity Proportion (%)	Power Consumption (%)
Interactive workloads		60 %	30 %
Batch Workloads	Periodic	24 %	70 %
	Aperiodic	16 %	

learning) are computationally expensive and usually with longer and more flexible execution time [19]. As illustrated in Table 1, batch workloads account for about 40 % in terms of number but contribute to about 70 % of the total power consumption. Batch workloads could be further divided as aperiodic batch workloads and periodic batch workloads. Aperiodic batch workloads (e.g., user-facing bulk downloading) refer to those submitted at arbitrary time intervals. Generally, they have no explicit deadlines and should be executed as soon as possible [20,21], which results in inflexible processing and power consumption characteristics. In contrast, periodic workloads (e.g., periodical data collection and operation monitoring) are those kept being submitted cyclically. This kind of workload accounts for about 24 % in terms of number of all workloads (about 60 % of the batch workloads) [22] and has been considered as one of the most promising power consumption flexibility sources from IDCs as their execution can be optimally deferred by a centralized job scheduler [21].

The power consumption flexibility potential of IDCs, which a number of researchers have investigated, is defined in this paper as the ability of IDCs to adjust the power load responding to the change in power systems, while maintaining acceptable service quality of the loads at the same time [23,24]. A few research works have been conducted on the flexibility quantification of IDCs. Generally, the flexibility is given monetarily with hypothetical data. For example, Cioara *et al.* [25] exploited the flexibility of IDCs in a designed energy market and developed several business cases to quantify the energy flexibility potentials of the cases in different working conditions. Chen *et al.* [26] established an operational flexibility analysis model of active distribution networks with data centers. The model of IT equipment is modeled based on the given numbers of executed workloads. Few works have answered how much power flexibility that IDCs could provide through workload scheduling based on real-world data trace and actual working situations.

One of the most popular objectives in this field is to minimize the electricity cost of IDCs. For example, Qureshi *et al.* [27] firstly studied the problem of reducing the energy cost for geo-distributed IDCs. Rao *et al.* [28] continued the exploration of minimizing the total electricity cost of IDCs under multiple electricity market environments, while guaranteeing the requirement of SLA of workloads. Based on [28], Yao *et al.* [7] designed a workload balancing framework for a data center to adjust the temporal distribution of batch workload by responding to the real-time electricity prices. Guenter *et al.* [29] presented an automated server provisioning system to meet workload demand while achieving key tradeoffs between cost, performance, and reliability. Some research has applied the optimal framework to various operating conditions in data centers and improved the performance from different aspects, such as water footprint optimization [30], carbon reduction [31,32], and cooling system optimization [33,34].

Another interesting line of the work on temporal shifting flexibility of IDCs is to enhance renewable energy integration. As many IDCs are equipped with on-site renewable generation resources, workload flexibility could be adopted as an effective tool to utilize more renewable energy [35–40]. Le *et al.* [35] proposed an optimization-based framework for IDCs to cap the brown energy while minimizing energy costs. Goiri *et al.* [37] presented a parallel batch job scheduler for IDCs to increase the use of renewable energies while avoiding deadline violations. Zhang *et al.* [38] designed a scheduling system that conducts dynamic workload request dispatching to increase renewable energy usage for IDCs, subject to the desired cost budget. Chen *et al.* [39] proposed a holistic workload scheduling scheme to increase renewable energy consumption across geo-distributed IDCs. The experiments showed a 40 % increment in renewable energy integration. Wang *et al.* [40] designed a combined power and thermal operation scheme for IDC to introduce operational flexibility to power systems. The results showed that the scheme could achieve a multi-energy co-optimization for IDCs and enhance wind power integration.

The aforementioned works demonstrate promising results on the demand response potentials of IDCs by re-scheduling the IT jobs at the workload level. However, most of the existing literatures oversimplified the flexible workload scheduling process. To be specific, the limitations can be summarized as follows.

(1) Most of those papers consider the workload as the smallest scheduling unit while current industry-scale IDC job schedulers typically divide the workloads into more minor and dependent jobs, tasks, and instances [13,41–44] during the execution

process. Generally, there may exist data dependencies among jobs within a workload. Each job will be assigned a deadline for execution, while shifting the workload as a whole may interfere with the execution principles of each job.

(2) Most of those works model the IDC power consumption as static and linear formulations w.r.t. workload execution. In reality, the mapping between power consumption and workload execution are highly dynamic and non-linear. Furthermore, most of those works use hypothetical or experimental data sets to illustrate the effectiveness. However, the real-world production data could be quite large-scale and heterogeneous, which may jeopardize the effectiveness of an oversimplified theoretical model.

To address those limitations, we propose a data-driven assessment framework for IDCs to investigate the aggregated power consumption flexibility of periodic batch workloads. We adopt a real-world production workload trace to illustrate its effectiveness. The contributions of this paper are summarized as follows:

(1) A data-driven flexibility assessment scheme of IDCs is proposed. We develop a four-step procedure that includes identifying the periodic jobs, extracting key operational patterns, mapping the power consumption with workload execution, and quantifying the flexibility in power system operation, all of which are established in a data-driven manner. The proposed data-driven scheme can overcome limitations of existing models on heterogeneous workloads and server configurations by using the field data to extract the power consumption mapping.

(2) We incorporate the periodic arrival pattern and data-dependency among jobs in the proposed flexibility assessment scheme. Three log-based algorithms are proposed to extract the arrival pattern of periodic workloads and execution sequence constraints from historical production data trace to accurately assess the power consumption flexibility. The proposed scheme can avoid irrational expectations on the flexibility potentials of IDCs by considering realistic job scheduling and inter-job dependencies in the algorithms, realizing practical adaptation and reliable estimation.

The rest of this paper is organized as follows: Section II introduces a data-driven flexibility assessment scheme to evaluate the power flexibility of periodic batch workloads in the face of typical computation logs. A historical production workload trace is utilized to test the feasibility and accuracy of the proposed scheme in Section III. And the conclusions are presented in Section IV.

2. Flexibility assessment scheme towards periodic batch workloads

2.1. System architecture

IDCs request thousands of IT devices to provide myriad internet services, including CPU for computing, memory devices for data storage, and I/O network devices for communication in servers [45], etc. Apart from that, cooling systems are relied upon to remove the heat generated along with intensive computation of IT equipment. On average, the direct electricity use in IDCs during operation mainly comes from these facilities [46]. Therein, the power demand of cooling systems is closely bound up with the IT equipment operation. That is, it mainly operates along with workload execution on servers [47], which emphasizes the significance of the workload execution process for flexibility assessment.

During execution, workloads, introducing a trace of computing requests and actions, will be separated into several dependent segments, with each in charge of a piece of computation target [11,48]. Some segments could be operated in parallel for efficient computation while some are executed in sequence. This process is shown in Fig. 1. Different

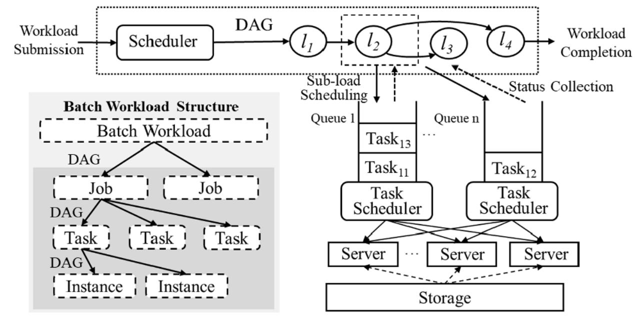

Fig. 1. Batch workload topology structure and executing diagram.

scheduling systems apply diverse workload structures. For instance, in Google's Borg [49], Microsoft's Azure [50], and Amazon's EC2 [51] management system, workloads are separated into jobs, and each job consists of multiple tasks. In Alibaba's Fuxi scheduling system, each task is further divided as instances executing simplist binary computation when executing [43].

The computation of some jobs is dependent on the results of others, which naturally generates computing sequences based on the data dependencies. In production, the sequence of workloads is defined as *Workflow*. This technique is widely deployed in most of the leading internet service providers (such as Google, IBM, Microsoft, Alibaba, etc.) for effective workload management, which instructs the steps of each proceeding execution. For the periodic workloads, their *Workflow* frameworks would be automatically operated several times during their lifetime. And the jobs within each periodic workload are repeatedly executed and performed as periodic jobs.

In the *Workflow*, data dependencies between jobs could be described as a directed acyclic graph (DAG) [11], as shown in Fig. 2. The *Workflow* framework designates a start time and an acceptable deadline for each job, and then they will be started to execute as scheduled. Under this circumstance, if the preceding jobs are finished before their deadlines, the subsequent jobs would still wait until their predefined start time [52]. This information indicates the promising temporal flexibility potentials of periodic workloads in production.

In Fig. 2, three kinds of data dependencies among jobs in *Workflow* are listed. α and β represent the recorded start time and end processing time of each job, respectively. $\beta - \alpha$ represents the processing duration of each job. The acceptable delay time is a determining indicator that describes the flexibility potential of the flexible periodic jobs, which is defined as slack time [53]. It could be calculated as follows, considering various kinds of data dependencies between two stages:

(1) In *Workflow1*, job l_2 relies on the results of its upstream job l_1. Then, the due time of l_1 is the start time of l_2, denoted as α_2. Thus, the slack time of l_1 is calculated as α_2-β_1, depicted in Fig. 3.

(2) In *Workflow2*, both l_1 and l_2 are upstream execution jobs of l_3. l_1 and l_2 could be processed in parallel since they would be assigned

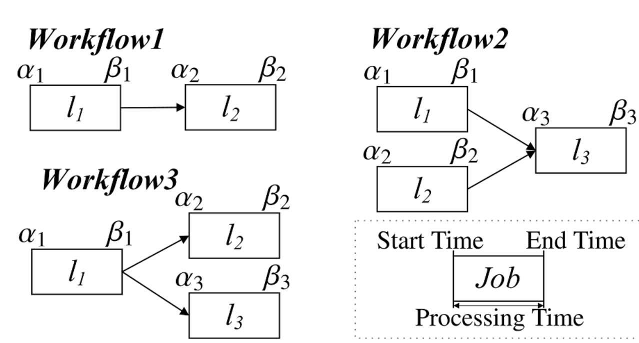

Fig. 2. Examples of dependencies among jobs in workflow.

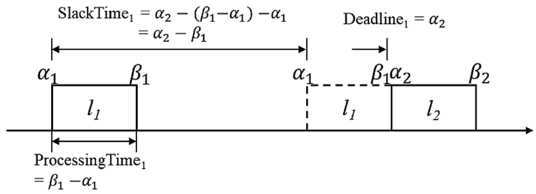

Fig. 3. Slack time calculation method of upstream job.

to separate virtual machines, which indicates that shifting any of them has little impact on the other job's computation process. Then, execution deadline constraints of l_1 and l_2 both depend on the start time of l_3. In this case, the acceptable delay time for l_1 is $\alpha_3 - \beta_1$, and l_2 is $\alpha_3 - \beta_2$, representing their temporal flexibility respectively.

(3) In *Workflow3*, l_1 is the upstream execution job for both l_2 and l_3. In this case, the execution deadline of l_1 depends on the earliest start time among its subsequent jobs. That is, the acceptable delay time for l_1 is calculated as $\min\{\alpha_2, \alpha_3\} - \beta_1$.

Based on the workload scheduling principles and dependencies among jobs in the *Workflow*, the time-shifting flexibility of periodic batch workloads would be assessed in detailed and accurate manners.

2.2. Flexibility assessment scheme description

All jobs' execution footprints are recorded in the computation logs, which will be stored and available for at least a decade on the storage [54]. The logs generally contain application-specific property information (such as job priority, signature, running server and clusters ID, etc.), resources occupation information (such as CPU, memory, disk, and network I/O usage, etc.), and execution information (such as start and end processing time, running state, etc.) [55]. Thus, the historical execution logs based on the raw context are parsed for realistic data mining and analysis on the temporal flexibility of periodic jobs in this paper. Generally speaking, log examining is a commonly employed method for engineers in maintenance of large-scale workload management systems. However, there are several challenges posed in evaluating power flexibility of IDCs based on raw logs in production:

(1) There is no existing algorithm to efficiently identify and label the periodic ones among the workloads. This is particularly challenging considering the heterogenous and large-scale nature of real-world data center workloads.
(2) Different workloads require varied sets of different resources and generate distinct DAG topologies during each execution, which introduces difficulties in power consumption modeling and temporal flexibility assessment.
(3) The existing power consumption models can hardly address the highly dynamic and non-linear characterization of the heterogenous and large-scale periodic workloads.

Furthermore, there is no existing power system operation models that combines those aforementioned features of periodic jobs with flexible job scheduling. To address these challenges, we propose a four-step data-driven flexibility assessment scheme of IDCs towards periodic batch workloads in this section. The framework is shown in Fig. 4. Three log-based algorithms are designed in the first three steps: (1) identification method of periodic jobs, (2) key pattern extraction method, and (3) power consumption fitting model with the data-driven method. And the fourth step of the scheme is to quantify the flexibility of IDCs towards periodic jobs associated with power system operation.

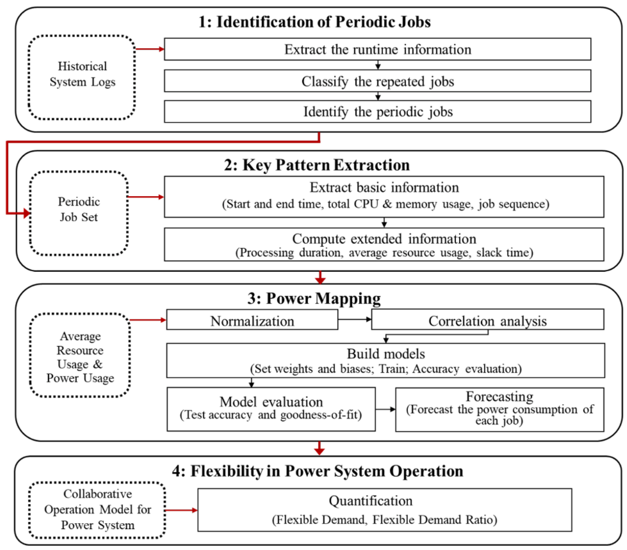

Fig. 4. Framework of the flexibility assessment scheme.

1) Identification method of periodic jobs

Considering both aperiodic and periodic batch jobs are colocated on the batch servers, their execution logs are mixed without direct label. Therefore, the first step of the scheme attempts to identify the periodic ones from the record mixture.

Algorithm 1. (*Periodic Jobs Identification*)

Input: Time frame investigated T; Start time T_l^{AT}, end time T_l^{ET}, signature sig_l of job l, $\forall l \in \mathbb{L}$; $CV^{threshold}$.
Initialize: Repeated job set $\mathbb{R} = \varnothing$, periodic job set $\mathbb{P} = \varnothing$.
/* Step 1: Grouping the repeated jobs */
for all $l \in \mathbb{L}$ **do**
  $n_l^{count} \leftarrow$ job l signature recurrence times within T.
  if $n_l^{count} > 1$ **then**
    $\mathbb{R} = \mathbb{R} + l$
  end if
end for
/* Step 2: Identifying the periodic jobs */
for all $l \in \mathbb{R}$ **do**
  for $n = [2, n_l^{count}]$ **do**
    $d_{n,l}^{diff,\,AT} \leftarrow T_{n,l}^{AT} - T_{n-1,l}^{AT}$
    $d_{n,l}^{diff,\,ET} \leftarrow T_{n,l}^{ET} - T_{n-1,l}^{ET}$
    $d_{n,l}^{diff,\,PT} \leftarrow \left(T_{n,l}^{ET} - T_{n,l}^{AT}\right) - \left(T_{n-1,l}^{ET} - T_{n-1,l}^{AT}\right)$
  end for
  Compute CV and MAD using Eq. (1) and Eq. (2).
  if $CV\left(d_l^{diff,\,AT}\right), CV\left(d_l^{diff,\,ET}\right), CV\left(d_l^{diff,\,PT}\right) \langle CV^{threshold}$ **then**
    $\mathbb{P} = \mathbb{P} + l$
  end if
end for
return periodic job set $\mathbb{P}$.

Among the recorded information, signature and execution information are the main parameters suggesting the periodicity, which are taken as the identification criteria in this algorithm. After submission, each workload is marked with a unique signature based on its underlying source codes to indicate the execution logic [56]. Therefore, signature matching implies that the computation programs use same logic and activity patterns; that is, the corresponding generated workload is performed repeatedly at different times. Based on this, the method is designed as shown in Algorithm 1, which includes two steps explained as follows.

(1) Identification of repeated jobs.

After acquiring a period of log files from certain IDC clusters, the data needed in this approach should be collected as the input of the algorithm. The first step is to identify the jobs that have same signatures and group them as the repeated jobs in this given time period. These repititions need to be further determined to be periodic ones depending on their duration of repetitions and inter-arrival time.

(2) Identification of periodic jobs.

Since periodic jobs must have a near-constant execution duration and inter-arrival time between repetitions, the coefficient of variation (CV) is adopted as the identification criterion to measure the dispersion of these parameters. CV is a commonly used metric of measuring repeatability and interval estimation of recurring data [57], calculated as Eq. (1). Median Absolute Deviation (MAD) is defined as another measure of reproducibility, modeled as Eq. (2) [58]. If the inter-arrival time is relatively constant and the execution duration is the same, it indicatesthat the workload is executed periodically. In addition, the threshold for the CV value is needed to be set to represent the deviation acceptance.

$$CV = \frac{MAD(X)}{median(X)} \tag{1}$$

where,

$$MAD(X) = median\left|x_j - \overline{x}\right|,\ x_j \in X \tag{2}$$

2) Key pattern extraction of periodic jobs.

In the second step, we extract the key parameters that could reflect the power consumption and the temporal flexibility patterns. Runtime parameters (including start and end processing time, processing duration, and resource usage) and the slack time are defined as the key patterns of each periodic job. For the reason that only start and end processing time and total resource usage of each computation are commonly recorded in the raw logs, other parameters need to be calculated through statistical methods. The extraction algorithm is shown as Algorithm 2.

Algorithm 2. (*Key Pattern Extraction of Periodic Jobs*)

Input: Start time T_j^{AT}, end time T_j^{ET}, total resource usage R_j^{CPU} and R_j^{mem}, signature sig_j of periodic job $j, (j \in \mathbb{P})$; Dependent job pairs $j_a \rightarrow j_B, j_B = \{j_{b_1}, j_{b_2}, \cdots, j_{b_n}\}$.
Initialize: $T_j^{PT}, T_j^{ST}, r_j^{CPU}, r_j^{mem} \leftarrow 0$.

```
function RuntimeIndicators()
  if T_j^ET > T_j^AT then
    Compute processing duration time using Eq. (3).
    Compute average resources usage using Eq. (4) and Eq. (5).
  end if
return T_j^PT, r_j^CPU, r_j^mem.
end function

function SlackTime()
  for all j ∈ P do
    Compute slack time using Eq. (6).
  end for
return T_j^ST
end function
return j = < T_j^AT, T_j^ET, T_j^PT, T_j^ST, r_j^CPU, r_j^mem >.
```

(1) Runtime indicators.

Processing duration of each job impacts the duration of demand response of the aggregated loads, which is calculated as Eq. (3). What's more, the average amount of computation resource usage during execution is more related to the dynamic power. Considering fast-growing computation and storage requirements, CPU and memory have been considered as two dominant power consumers of servers during operation [59–61]. The average CPU and memory usage are calculated as Eqs. (4) and (5).

$$T_j^{PT} = T_j^{ET} - T_j^{AT}, \forall j \in \mathbb{P} \tag{3}$$

$$r_j^{CPU} = \frac{R_j^{CPU}}{T_j^{PT}}, \forall j \in \mathbb{P} \tag{4}$$

$$r_j^{mem} = \frac{R_j^{mem}}{T_j^{PT}}, \forall j \in \mathbb{P} \tag{5}$$

(2) Slack time.

As shown in Fig. 3, upstream jobs should meet the earliest read request of the subsequent jobs. According to its definition, the slack time of each job could be obtained by Eq. (6). Noting that if a job has no downstream jobs waiting for its results, then it is said that it has no deadline and would be processed as soon as possible. Then, its slack time is set as 0, and its time-shifting flexibility is not taken into consideration in the following works.

$$T_{j_a}^{ST} = \begin{cases} 0, & j_a \rightarrow \varnothing \\ T_{j_b}^{AT} - T_{j_a}^{ET}, & j_a \rightarrow j_b \\ \min\left(T_{j_B}^{AT}\right) - T_{j_a}^{ET}, & j_a \rightarrow j_B (j_B = j_{b_1}, j_{b_2}, \cdots, j_{b_n}) \end{cases} \tag{6}$$

In sum, the execution patterns of each periodic job are labeled with a tuple, which consists of 6 key runtime parameters, denoted as $j = < T_j^{AT}, T_j^{ET}, T_j^{PT}, T_j^{ST}, r_j^{CPU}, r_j^{mem} >$. Based on that, the power consumed by each job could be estimated and the temporal flexibility could be quantified in the next model.

3) Power consumption mapping model.

Power consumption estimation model of each job execution is more complex than static mathematical mappings considering the heterogeneous configuration of servers and workload types. In this section, a data-driven method using Back Propagation Neural Network (BPNN) to access the power consumption mapping is proposed in this section, as shown in Algorithm 3 [62].

CPU and memory usage are utilized as main input vectors in the model training framework. Based on [63], the total power consumption of a server could be divided into fixed part and dynamic part. The fixed part represents the power consumption when server is idle [64], which is generally considered as a constant value. The dynamic part represents the dynamic power demand for computation, data writing and reading, which is closely related to the CPU and memory usage [65]. Based on Eq. (7), the dynamic power of the servers is calculated and utilized as the output vectors in the network. During the training process, mean absolute error (MAE), mean square error (MSE) and R^2 score are set as the criteria to evaluate the performance of the trained model. If the evaluated performance is acceptable, the network would be utilized to estimate the dynamic power demand of periodic jobs based on their average resource usage extracted by the second model.

$$P_{server}^{total} = P_{server}^{fixed} + f\left(r_{server}^{CPU}, r_{server}^{mem}\right) \tag{7}$$

Algorithm 3. (*Power Mapping with BPNN*)

Input: (1) Training set: input vectors $\overline{X}$ (server-level CPU and memory usage), output values y (dynamic power consumption of servers, calculated by Eq. (7)); **(2) Forecasting:** average CPU usage r_j^{CPU}, average memory usage r_j^{mem} of periodic jobs.
Initialize: Weights and biases; Min-max normalization of data sets.
Set: Training set (80%), validation set (20%).
Create: BPNN network.
1: **Inputs:** Load the training and test data.
2: Choose the weights and biases.
3: **while** (stopping condition is not met) **do**
4: Implement train steps for each data point.
5: Implement for testing data points.
6: **end while**
7: Calculate MAE, MSE, R^2 for performance measure.
8: **for** all $j \in \mathbb{P}$ **do**
9: Compute power consumption of job j by trained model.
10: **end for**
11: **return** power consumption of periodic jobs P_j^{job}.

4) Flexibility in power system operation.

Typically, the interaction between IDCs and the bulk power system can be simulated with model-based approaches (e.g., mathematical programming) or model-free approaches (e.g., deep reinforcement learning). Without loss of generality, we adopt a mixed-integer linear programming (MILP) model to evaluate the flexibility potential of IDCs. The supply side of the system is composed of wind, solar, and fossil-fuel generators, while IDC clusters are considered as the end-users in the system. The objective is to minimize the operation cost of the system, i.e., the sum of operating cost and start-up/down cost of conventional fossil-fuel generators, as shown in Eq. (8). In this case, the flexible operation of IDCs could play an important role in power system operation, particularly for those with a high penetration level of intermitted resources.

$$\min \sum_{t}^{T} x_{i,t}^{gen,on}\left(a_i + b_i p_{i,t}^{gen}\right) + x_{i,t}^{gen,su} c_i^{gen,su} + x_{i,t}^{gen,sd} c_i^{gen,sd} \tag{8}$$

(1) Job scheduling and power load model of IDCs.

In this system, periodic jobs with slack time could be delayed to other time slots in response to the signals from the power system. As mentioned, time constraints are the top priorities during each rescheduling process, which are modeled in equations shown as follows.

$$\sum_{t}^{T} x_{j,t}^{job} = \sum_{t=T_j^{AT}}^{T_j^{DDL}} x_{j,t}^{job} = T_j^{PT}, \forall j \in \mathbb{P} \tag{9}$$

$$T_j^{DDL} = T_j^{ET} + T_j^{ST}, \forall j \in \mathbb{P} \tag{10}$$

Eq. (9) guarantees that each periodic job would be dispatched to a time slot between $[T_j^{\mathrm{AT}}, T_j^{\mathrm{DDL}}]$, and it also guarantees that the processing time lasts for T_j^{PT}. $x_{j,t}^{\mathrm{job}}$ indicates the execution actions of job j in the optimal situation. Eq. (10) demonstrate the calculation of execution deadline for each periodic job. The variable vector indicating execution actions for each periodic job of the submission day is shown as:

$$[x_{j,t}^{job}] = \underbrace{[0, \cdots, 1, 1, \cdots, 1, 1, 0, \cdots]}_{\sum T_j^{PT}}$$

Generally, a job execution process, which is responsible for the automatic separation, assignment and execution of multiple tasks and/or instances, could not be manually interrupted [66]. Therefore, each job execution is considered as a continuous process, as defined by Eqs. (11) and (12). Eq. (11) is used for the subsequent periods after arrival time T_j^{AT} to satisfy the processing duration during all the possible sets of consecutive periods of size T_j^{PT}. Eq. (12) constrains the final periods before deadlines T_j^{DDL} in which periodic job j is about to compute, it remains processing until the end of its possible time span [67].

$$\sum_{k=t}^{t+T_j^{PT}-1} x_{j,k}^{job} \geqslant \left(x_{j,t}^{job} - x_{j,t-1}^{job}\right) T_j^{PT}, \forall t \in \left[T_j^{AT}+1, T_j^{DDL} - T_j^{PT} + 1\right], \quad \forall j \in \mathbb{P} \tag{11}$$

$$\sum_{k=t}^{T_j^{DDL}} \left(x_{j,k}^{job} - \left(x_{j,t}^{job} - x_{j,t-1}^{job}\right)\right) \geqslant 0, \forall t \in \left(T_j^{DDL} - T_j^{PT} + 1, T_j^{DDL}\right], \quad \forall j \in \mathbb{P} \tag{12}$$

Processing different jobs introduces varied power loads. Therefore, the system optimization problem shall not only consider various constraints in temporal aspects, but also the power load that each periodic job representing. The aggregated power load of all periodic jobs at each time slot is shown in Fig. 5, modeled as Eq. (13). Apart from that, the power consumption introduced by other devices in the batch clusters are considered as fixed load, including power consumption of other types of workloads, and associated power demand for cooling, lighting, power distribution, etc. The total demand for IDC clusters is shown in Eq. (14).

$$p_t^{\mathbb{P}} = \sum_{j}^{\mathbb{P}} \mathrm{x}_{j,t}^{job} \mathrm{P}_j^{job}, \forall \mathrm{t} \in \mathrm{T} \tag{13}$$

$$p_t^{dc} = p_t^{\mathbb{P}} + \mathrm{P}_t^{other}, \forall \mathrm{t} \in \mathrm{T} \tag{14}$$

(2) Power generation constraints.

Conventional fossil-fuel generators are used to serve data center clusters' power demand when no sufficient renewable output is available in the coordinative system. To consider the dispatch inflexibility introduced to the system by fossil-fuel generators in general operation manners, we use the following unit commitment (UC) model, shown in Eqs. (15) – (23). Renewable generation is limited by the capacity of the generators, shown in Eqs. (24) – (25). Power balance constraint is shown as Eq. (26).

$$x_{i,t}^{gen,on} P_i^{gen,\min} \leqslant p_{i,t}^{gen} \leqslant x_{i,t}^{gen,on} P_i^{gen,\max}, \forall i \in I, \forall t \in T \tag{15}$$

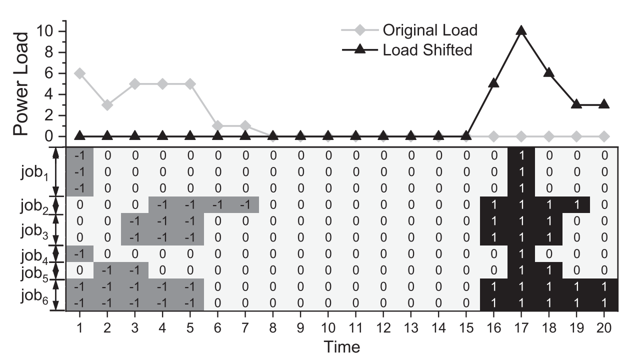

Fig. 5. Binary variables representing execution actions of periodic jobs and aggregation of load profile. In this figure, $x_{j,t}^{job} = -1$ represents that the job j was originally designed to be executed at time t but was shifted to other time slots.

$$x_{i,t}^{gen,on} - x_{i,k}^{gen,on} - x_{i,t-1}^{gen,on} \leqslant 0, \forall i \in I, \forall t \in T, \quad \forall k \in \left[t+1, \min\left(T, T_i^{gen,MU} + t - 1\right)\right] \tag{16}$$

$$-x_{i,t}^{gen,on} + x_{i,k}^{gen,on} + x_{i,t-1}^{gen,on} \leqslant 1, \forall i \in I, \forall t \in T, \quad \forall k \in \left[t+1, \min\left(T, T_i^{gen,MD} + t - 1\right)\right] \tag{17}$$

$$x_{i,t}^{gen,on} - x_{i,t}^{gen,su} - x_{i,t-1}^{gen,on} \leqslant 0, \forall i \in I, \forall t \in T \tag{18}$$

$$-x_{i,t}^{gen,on} - x_{i,t}^{gen,sd} + x_{i,t-1}^{gen,on} \leqslant 0, \forall i \in I, \forall t \in T \tag{19}$$

$$\sum_{k=1}^{t} x_{i,k}^{gen,on} - \left(x_{i,t-1}^{gen,on} - x_{i,t}^{gen,on}\right) T_i^{gen,on} \geqslant 0, \forall i \in I, \forall t \in T \tag{20}$$

$$\sum_{k=1}^{t} x_{i,k}^{gen,on} - \left(x_{i,t-1}^{gen,on} - x_{i,t}^{gen,on}\right) T_i^{gen,on} \geqslant 0, \forall i \in I, \forall t \in T \tag{21}$$

$$p_{i,t}^{gen} - p_{i,t-1}^{gen} \leqslant x_{i,t-1}^{gen,on} P_i^{gen,ru} + \left(1 - x_{i,t-1}^{gen,on}\right) P_i^{gen,\min}, \forall i \in I, \forall t \in T \tag{22}$$

$$p_{i,t-1}^{gen} - p_{i,t}^{gen} \leqslant x_{i,t}^{gen,on} P_i^{gen,rd} + \left(1 - x_{i,t}^{gen,on}\right) P_i^{gen,\min}, \forall i \in I, \forall t \in T \tag{23}$$

$$0 \leqslant p_t^{wind} \leqslant P_t^{wind}, \forall t \in T \tag{24}$$

$$0 \leqslant p_t^{PV} \leqslant P_t^{PV}, \forall t \in T \tag{25}$$

$$\sum_{i}^{I} x_{i,t}^{gen,on} p_{i,t}^{gen} + p_t^{wind} + p_t^{PV} = p_t^{dc}, \forall t \in T \tag{26}$$

(3) Flexible demand evaluation.

In this paper, we propose two types of indexes, flexible demand, and flexible demand ratio, to assess the flexibility associated with power system operation. The flexible demand D_t^{flex} refers to the flexibility potential of all periodic jobs in time slot t, which could be measured by Eq. (27) [68]. The flexible demand ratio R_t^{flex} could be calculated by Eq. (28), which is presented as the ratio of flexible demand to the original power consumption. These flexibility indexes could be positive or negative, indicating adjusting the load upward or downward in this time interval, respectively.

$$D_t^{flex} = p_t^{dc} - P_t^{ori}, \forall t \in T \tag{27}$$

$$R_t^{flex} = \frac{D_t^{flex}}{P_t^{ori}}, \forall t \in T \tag{28}$$

3. Case study: Alibaba production data set

In this section, the proposed flexibility assessment scheme of IDCs considering periodic workloads is evaluated by a real production workload trace from Alibaba. And the time-shifting flexibility of data centers is quantified.

3.1. Overview of alibaba workload trace.

The workload trace studied in this section records the execution footprints of the batch jobs generated on 8272 servers in 3 batch clusters across one week in Alibaba. Batch clusters are designed for computing batch workloads only. In this case, this workload trace contains both periodic batch jobs and aperiodic batch jobs. Without loss of generality, no holidays or e-commerce shopping carnivals are included in this selected time frame.

We use Algorithm 1 to identify the periodic jobs from the mixture. The results indicate that the periodic jobs account for about 55 % of the batch jobs on these clusters, and more than 97 % of them are repetitive

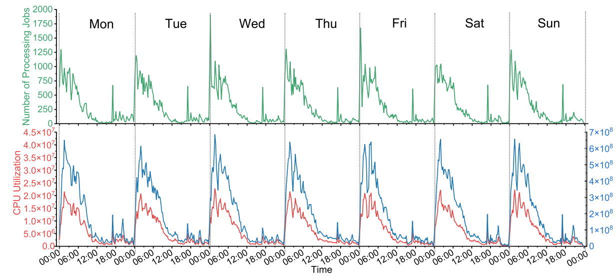

Fig. 6. Runtime parameters of jobs with daily periodicity.

on a daily basis, with the number more than 26,000 on average. Therefore, we mainly focus on the daily routine jobs with periodic arrival patterns and data dependencies for the flexibility assessment model.

1) Analysis on periodic jobs.

We use Algorithm 2 to extract the critical runtime parameters of the periodic jobs and the results are reported in Fig. 6. From this figure, we can observe clear peak-to-valley differences in the distribution of the quantity, CPU and memory usage during a one-day period. A large portion of periodic jobs are processed in the early morning (from 0:00 to 8:00). For example, there is a surge that can be observed at 0:00. For hours after 8:00, the quantity stays at a relatively low level. The computation resources usage has a similar trend. The main reason is that most of the periodic jobs in this cluster are SQL query jobs for data collection and statistics for business activities of the last day. These works typically need to be completed before 8:00 for business management purposes in the next day. On the other hand, aperiodic batch jobs are processed as quickly as possible because they have higher priority in computation resource preemption on the batch clusters owing to their user-facing features. Most aperiodic batch jobs are generated temporarily and submitted by application developers manually, such as ad-hoc testing and queries [69]. After 8:00, more aperiodic batch jobs are submitted and assigned to these clusters, requesting bulk computation resources and prior execution. Considering the resource preempted by those aperiodic jobs, the quantity and computation resources usage of the periodic jobs stays at a lower level in those hours.

In addition, some periodic jobs are automatically submitted at about 18:00 each day, forming a surge in job quantity and resource consumption. This is because some of them are defined for generating business reports for the day. Furthermore, the arrival rate of aperiodic jobs decreases after 18:00, and the resource occupied are released.

2) Execution patterns variation.

As shown, the execution of periodic jobs does not follow a constant pattern across the week in production. This is because the stochastic submission of higher-priority aperiodic jobs introduces uncertainty to the availability of computation resources for lower-priority periodic jobs. Therefore, the execution patterns may fluctuate within a certain

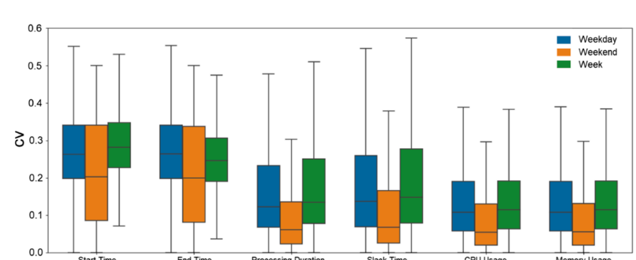

Fig. 7. CV value of the execution patterns of the periodic batch jobs.

range.

The CVs for the execution patterns, including 6 key runtime parameters of the periodic jobs in this week are shown in Fig. 7. The workload trace is divided into weekdays and weekends in this section to compare the distribution of CVs with the entire week trace. In this figure, shorter boxes indicate that the variation in execution patterns of the periodic jobs is more concentrated during this time period, while longer boxes indicate more dispersed distribution of the execution pattern.

From this figure, we can observe that start and end processing time have the most significant variation in different subsets compared with other parameters. This is because the centralized job schedulers may postpone the execution of periodic jobs due to insufficient resources. Moreover, the variation of processing duration, resources usage, and slack time of the periodic jobs appears relatively low. This indicates that the execution sequence and content do not vary much no matter when the periodic jobs start processing. In addition, the variation of execution patterns on weekends is much lesser than those on weekdays, suggesting that job execution during weekends follows the predefined schedule in a strict manner. The reason is that the quantity of user-facing aperiodic jobs with higher priority from application developers is less during weekends, so there would be less resource preemption and execution delay for the periodic jobs.

3.2. Flexibility analysis.

1) Temporal flexibility.

The temporal flexibility of job execution is analyzed in this subsection. To better illustrate the relationship between data dependency, execution sequence, and slack time, an example periodic workload is depicted in Fig. 8. In this figure, each circle denotes a job. Its color refers to the average processing duration, and its size refers to the slack time. The arrows indicate the data dependencies and execution sequence between jobs. Among them, the orange arrows indicate the temporal flexibility of the upstream jobs is determined by the earliest downstream jobs. For example, the slack time of Job 9 is determined by Job 12, which is the earliest one among the downstream jobs.

Fig. 9 shows the slack time distribution of periodic jobs on weekdays and weekends, respectively. As can be seen, the slack time of periodic jobs decreases as their end processing time approaches 24:00. This finding reveals that the daily routine jobs generally are completed within the submission day. Moreover, a large portion (66 % on weekdays and 68 % on weekends) of periodic jobs have a deadline at 8:00. This observation is consistent with the early observation that a majority of periodic jobs in the studied clusters are scheduled to be finished before 8:00 for business management purposes.

2) Power profiling.

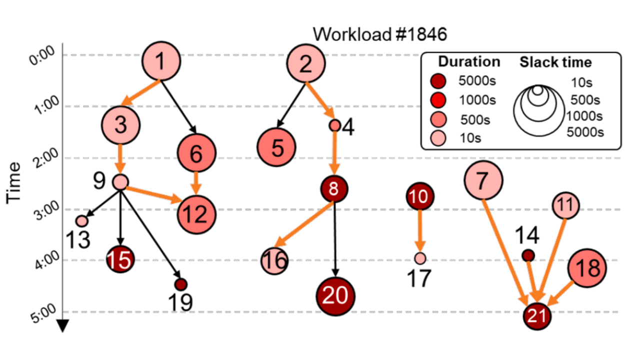

Fig. 8. The decomposition of a periodic workload into jobs with data dependencies.

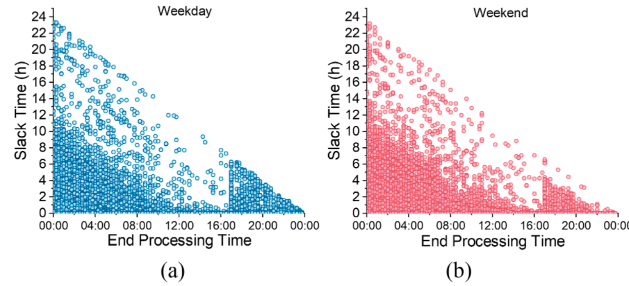

Fig. 9. Slack time distribution of periodic jobs on weekdays and weekends.

By applying Algorithm 3, we use the BPNN-based method to establish a data-driven power consumption mapping model at the job level. We extract the CPU and memory data and the corresponding server power data from the production log files at the granularity of 10 s. With 8272 servers investigated and a week of data trace studied, more than 500 million input–output data pairs could be utilized for the data-driven mapping progress. To accelerate the progress and ease the computation burden, we use the increment learning technique in which we first train the mapping model with a portion of data, and then update the trained model with new data to extend the existing model's knowledge [70]. This step is taken via *Tensorflow* 8.0. After training and validation, the network is then utilized to estimate the power consumption of each periodic job.

The aggregated power demand of periodic jobs per 15-minute interval during a one-day period on the weekday and weekend are reported in Fig. 10. As could be seen, the power consumption profile and the resource usage profile of all periodic jobs have similar trends, i.e., they peak during the early morning hours and then continue to decline until a spike occurs at around 18:00. We can observe a nonlinear correlation between CPU and memory utilization and the power consumption from 2:00 to 6:00. The main reason is that the CPU and memory utilization do not contribute in the same proportion to the server power consumption for different kinds of workloads [65]. To be specific, the periodic jobs during this time generally require writing the results to the disks, introducing more power demand to the hard disks on the server. Moreover, the interference between the execution process of various jobs may result in the total resource usage not matching the sum of the individual resource consumption of each job. The power consumption profile on weekday is slightly different from the weekend situation due to the change in users' requests.

Moreover, we compare the performance of aforementioned method with three other commonly used machine learning models to address the accuracy and goodness-of-fit, including Gradient Boost Regressor (GBR), Support Vector Machine Regression (SVR), and Convolutional Neural Network (CNN). These models are trained with same data set. The MAE, MSE, and R^2 of these models are compared in Table 2. As could be seen,

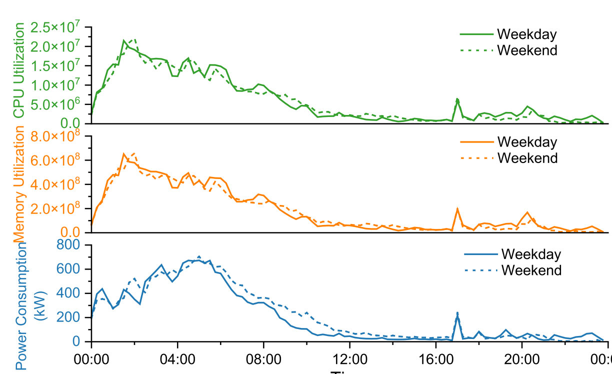

Fig. 10. Power consumption of periodic jobs on weekdays and weekends.

Table 2
Performance Measures: MAE, MSE and R^2.

	GBR	SVR	CNN	BPNN
MAE	34.97	38.44	36.86	34.13
MSE	2364.1	2776.4	2486.5	2361.9
R^2	0.3940	0.2226	0.728	0.839

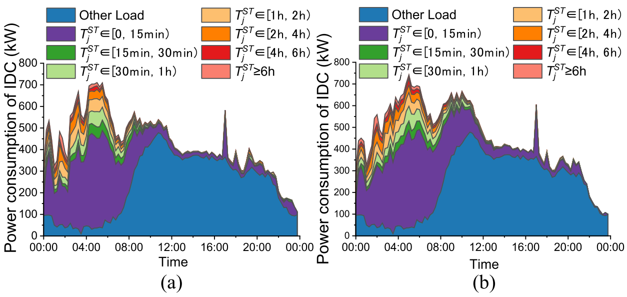

Fig. 11. Assessed flexibility towards periodic jobs on weekdays and weekends.

the BPNN model performs the best with highest value of R^2 and small value of MSE and MAE.

3) Flexibility quantification in power system operation.

The aggregated flexible load of periodic jobs and other power load of the studied IDCs is depicted in Fig. 11, which illustrates the demand shifting capability in the studied clusters. Since the other power load of the batch clusters is introduced by manually submitted aperiodic workloads and associated power demand for auxiliary devices, it shares similar load patterns with general load profiles of power systems. Thus, the value for other power demands of batch clusters is set based on [71]. To better demonstrate the demand shifting flexibility, we categorize the periodic jobs power consumption into 7 groups. As can be observed, approximately 68 % of the power load is deferrable within a period shorter than 15 min. The proportion of flexible load with a slack time of [1 h, 2 h), [30 min, 1 h), and [2 h, 4 h) are 11 %, 9 %, 8.9 %, respectively. The power and deferrable time distribution is similar between weekdays and weekends. Therefore, the power consumption introduced by the periodic jobs on the weekday is taken as the original load in the flexibility assessment process for simplicity.

The flexibility provided by periodic jobs in IDCs is quantified in this subsection. We consider a collaborative operation scenario for the high renewable penetrated power system where the studied clusters are located. In this model, IDCs adjust their power load profiles by rescheduling the execution patterns of periodic jobs under the limits of computation resource burden and execution deadline limitations. The model is tested with a one-year of wind and solar power generation observed in 2020 by a research program, and the profiles are shown in Fig. 12. We rescale the capacity of renewable generation to together meet the renewable energy penetration level of about 50 % of the total power capacity in the system. It is worth to mention that the proposed scheme is scalable, and also can be applied to other power system operation models with various objectives, such as maximizing social welfare [72], enhancing the PV integration [15], and reducing the operation costs [73]. Other operating parameters, such as marginal operating cost, on/off time limits, and ramping on/off limits of conventional generators, are set based on [74]. The model is then solved by Gurobi 7.0 in Python 3.8 on Intel(R) Core(TM) i7-9750H CPU.

To evaluate the flexibility of IDCs, two cases are designated as follows.

(a) Case I: Original Load - power system with wind and solar energy sources penetration level of 50 % and fixed power loads in the demand side.
(b) Case II: Flexible Load - power system with same electricity mix with Case I and flexible power loads in demand side.

The optimal dispatch strategies of two cases are shown in Fig. 13. To highlight the curtailment of renewable energies in two cases, the combined maximum wind and solar generation profiles are shown in this figure as reference. As could be seen from the results in Case II, about 400 kW of power load is shifted from 2:00–4:00, when the renewable energy supplies are in shortage, and about 80 % of them are scheduled to time period from 5:00 to 8:00, and the rest are scheduled separately to 8:00–17:00. When the solar energy is considerably generating, more periodic jobs with 1100 kWh of electricity usage are designated to be processed during daytime, compared with the results in the Case I. In sum, the flexibility of IDCs towards periodic jobs could reduce about 10 % of the renewable energy curtailment.

Results of the load shifting flexibility potential quantification for the periodic jobs are shown in Fig. 14. The black solid line represents the original load profile of periodic jobs. And the gray area represents the flexibility range between the maximum upward and downward load variation, indicating the time-shifting capability of IDCs at each time interval.

Fig. 15 shows the flexible demand deviations and the flexible demand ratio towards the periodic jobs compared with the original load. As could be seen, the load shifting flexibility of IDCs mainly lies before 8:00 a.m. It can also be concluded that the flexible demand decreases with time by the fact that the grey area in the figure narrows when approaching 24:00. The maximum upward load variation is about 400 kW, and downward is about 325 kW. Moreover, the maximum R_t^{flex} could reach to 80 % and about 30 % on average, and the maximum downward flexibility ratio could reach 50 % and about 20 % on average. This flexibility profile is favorable to wind power high integration because power consumption generally decreases during the early morning. The results show that the proposed flexibility assessment scheme can provide valuable and practical flexibility quantification on IDCs in production environments.

It is worth mentioning that this scheme focuses on flexibility quantification of periodic batch workloads on small-scale server clusters for data availability and confidentiality, though this method can also be

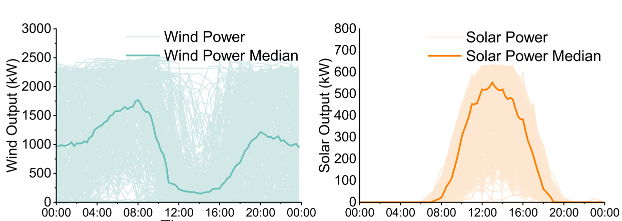

Fig. 12. Wind and solar power supply profiles.

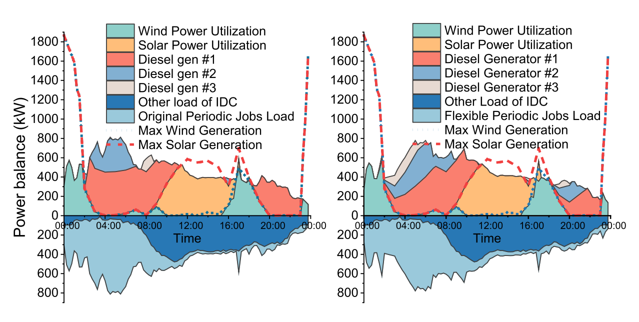

Fig. 13. Power balance. (a) Case I: optimal dispatch with fixed power loads, (b) Case II: optimal dispatch with flexible power loads.

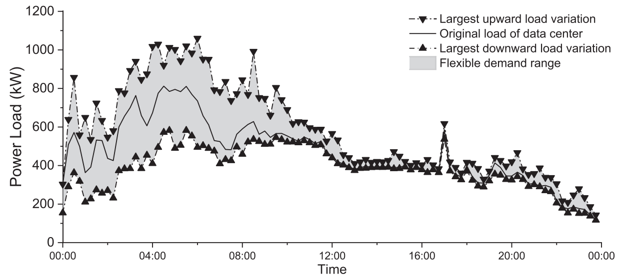

Fig. 14. Flexible power demand towards periodic jobs under various wind power generation scenarios.

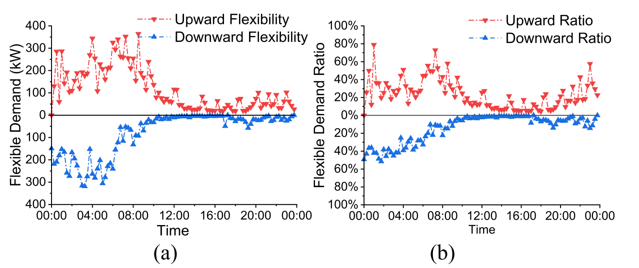

Fig. 15. Flexible power demand capacity and ratio under various wind power generation scenarios.

applicable on large-scale clusters.

3.3. Sensitivity analysis.

1) Impact of data-driven approach.

This subsection aims to illustrate the power consumption mapping accuracy of the proposed data-driven scheme in dealing with the heterogeneous configuration of servers and workloads. In this regard, six existing power mapping models of servers are tested with the historical production data for performance comparison, which are Xenpm [75], MBFD [76], STAMP [77], DCSC [78], DGLB [8], SMG [79], as shown in Table 3. In the first three models, the power demand of servers is established as mathematical models closely related to CPU or memory utilization. The last three models consider the quantity of executing workloads as a critical variable that affects power demand.

The accuracy of those models is measured by MAE and Normalized Root Mean Square Error (NRMSE). The results are reported in Fig. 16.

As shown in Fig. 16(a), the proposed data-driven scheme outperforms Xenpm, MBFD, and STAMP as it has the lowest MAE and NRMSE. The comparison results demonstrate that the heterogeneous servers, especially in production scales, are not well represented by those steady-state models. In contrast, the proposed scheme can overcome this limitation by training the network with historical data generated on all kinds of production servers.

Table 3
Configuration of The Power Mapping Models.

Model	Variables	Parameters	Works
Xenpm	CPU cores usage	–	[75]
MBFD	CPU utilization ratio	Idle power	[76]
STAMP	CPU and memory utilization	Idle power	[77]
DCSC	Executing workload quantity	PUE, service rate, idle power	[78]
DGLB	Executing workload quantity	PUE, service rate, peak power	[8]
SMG	Executing workload quantity	PUE, service rate, peak and idle power	[79]

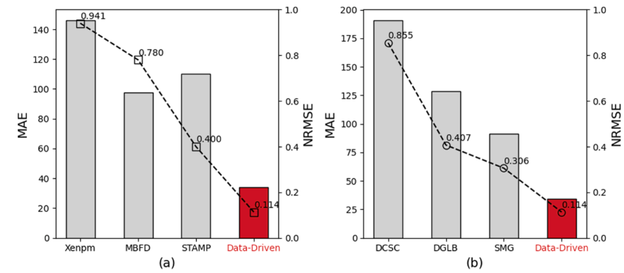

Fig. 16. Performance comparison of the existing power mapping models.

Similarly, the proposed scheme has a better performance than the last three models, as shown in Fig. 16(b). Since the workloads are highly diverse and myriad in production, the model of estimating server power by executing workload quantity is not as adaptive as the proposed model. In other words, those benchmark models are typically based on static PUE which is generally a yearly average value. Compared with that, the proposed scheme estimates the power consumption mapping based on the time-varying resource usage via the BPNN.

2) Impact of data dependency consideration.

In this subsection, we provide a comparison case to illustrate the impact of data dependency consideration. Without time constraints from the downstream jobs, jobs can be executed at any time within the submission day after arrival. In this case, the execution deadlines of the periodic jobs are designated as 24:00. This comparison case is tested with the same operation scenario for the high wind penetrated power system, and the simulation results are reported in Fig. 17.

As could be observed, the periodic jobs in this collaborative operation scenario are shifted to be processed in the morning and evening. The flexibility range is amplified with maximum upward and downward load variation of 2000 kW and 800 kW, respectively. However, this scenario substantially exaggerates the flexibility capacity since the periodic jobs need to be scheduled while considering the constraints of execution sequence induced by data dependency. Therefore, the upstream and downstream jobs cannot have temporal flexibility simultaneously. It shall be mentioned that this is a critical factor to investigate the flexibility of data centers as most of the batch jobs have data dependency constraints [80].

4. Conclusions

In this study, we propose a four-step data-driven flexibility assessment scheme for IDCs towards periodic batch jobs. The scheme is applied to a historical workload trace generated on 3 batch clusters in Alibaba in actual production environment. The periodic jobs are identified from the workload mixture, and the power consumption of each periodic job is estimated via a neural network-based approach. Based on

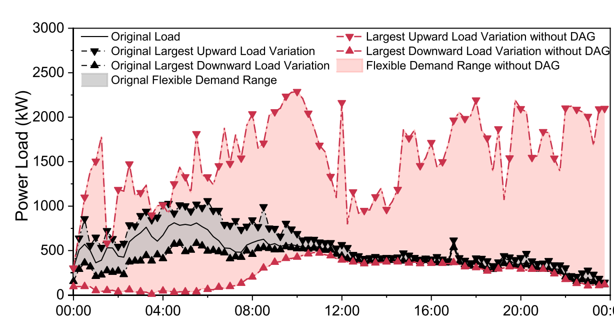

Fig. 17. Flexible power demand towards periodic jobs without DAG dependency.

the proposed scheme, the intrinsic temporal flexibility of IDCs introduced by periodic jobs could be quantified by finding the optimal schedule of job execution in operation scenarios of power systems with a high penetration level of renewable generation through two applicable metrics. The case study results reveal that the periodic batch jobs in the workload trace could provide power consumption flexibility from 2:00 to 9:00. The maximum upward and downward load variation is 400 kW and 325 kW, respectively. The comparison cases with steady-state power mapping models and the scheme without considering data dependencies indicate that the proposed scheme could avoid inaccurate estimation of the IDC flexibility by modeling the job scheduling in a more realistic manner.

CRediT authorship contribution statement

Yujie Cao: Methodology, Software, Validation, Investigation, Writing – original draft. **Ming Cheng:** Investigation, Writing – review & editing. **Sufang Zhang:** Writing – review & editing. **Hongju Mao:** Investigation, Writing – review & editing. **Peng Wang:** Investigation, Writing – review & editing. **Chao Li:** Investigation. **Yihui Feng:** Investigation. **Zhaohao Ding:** Conceptualization, Software, Writing – review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

This research was done while interning at Alibaba Group. This work was supported in part by the National Natural Science Foundation of China under Grant 51907063; in part by Alibaba Group through Alibaba Innovative Research (AIR) Program; in part by the Young Elite Scientists Sponsorship Program by CAST under Grant 2020QNRC001; in part by the Fundamental Research Funds for the Central Universities under Grant 2022MS001; and in part by National Social Science Foundation of China under Grant No.21BJY012.

References

[1] Masanet E, Shehabi A, Lei N, Smith S, Koomey J. Recalibrating global data center energy-use estimates. Science 2020;367:984–6.
[2] Ye R, Yuan Y, Wei J, Wu S, Zhao Y, Ma C, et al. The roadmap to decarbonization of China's digital infrastructure: data center and 5G carbon reduction potential and challenges (2020-2035). Greenpeace, China Electronic Product Reliability and Environmental Testing Research Institute (CEPREI); 2021.
[3] Sawyer RL. Calculating Total Power Requirements for Data Centers Schneider Electric; 2011. p. pp.1-10.
[4] Kubernetes. Workloads, https://kubernetes.io/docs/concepts/workloads/; 2021.
[5] Chang TW, Lin CC, Liu P, Wu JJ, Huang CW. Resource Provision for Batch and Interactive Workloads in Data Centers. IEEE International Conference on Parallel & Distributed Systems 2016.
[6] Li J, Bao Z, Li Z. Modeling demand response capability by internet data centers processing batch computing jobs. IEEE Trans Smart Grid 2014;6:737–47.
[7] Yuan Y, Huang L, Sharma A, Golubchik L, Neely M. Data centers power reduction: A two time scale approach for delay tolerant workloads. Infocom, IEEE2012. p. 1431-9.
[8] Chen T, Marques AG, Giannakis GB. DGLB: Distributed stochastic geographical load balancing over cloud networks. IEEE Trans Parallel Distrib Syst 2016;28: 1866–80.
[9] Liang Y, Tao J, Zou Y. Distributed Real-Time Energy Management in Data Center Microgrids. IEEE Transactions on Smart Grid. 2016;PP:1-.
[10] Luo J, Rao L, Liu X. Temporal Load Balancing with Service Delay Guarantee for Energy Cost Optimization in Internet Data Centers. Parallel and Distributed Systems, IEEE Transactions on. 2013;PP:1.
[11] Zhang Z, Li C, Tao Y, Yang R, Tang H, Xu J. Fuxi: a fault-tolerant resource management and job scheduling system at internet scale. In: Proceedings of the VLDB Endowment. VLDB Endowment Inc; 2014. p. 1393–404.
[12] Dong Z, Zhuang W, Rojas-Cessa R. Energy-aware scheduling schemes for cloud data centers on google trace data. In: 2014 IEEE Online Conference on Green Communications (OnlineGreenComm): IEEE; 2014. p. 1–6.
[13] Reiss C, Tumanov A, Ganger GR, Katz RH, Kozuch MA. Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In: Proceedings of the Third ACM Symposium on Cloud Computing; 2012. p. 1–13.
[14] Lu C, Ye K, Xu G, Xu CZ, Bai T. Imbalance in the cloud: An analysis on Alibaba cluster trace. 2017 IEEE International Conference on Big Data (Big Data)2018.
[15] Liu Z, Chen Y, Bash C, Wierman A, Gmach D, Wang Z, et al. Renewable and cooling aware workload management for sustainable data centers. In: Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems; 2012. p. 175–86.
[16] Zhao Y, Li S, Hu S, Wang H, Yao S, Shao H, et al. An experimental evaluation of datacenter workloads on low-power embedded micro servers. Proceedings of the VLDB Endowment 9, no 9 (2016): 696-707. 2016.
[17] Bigelow SJ. Workload, https://searchdatacenter.techtarget.com/definition/workload; 2020.
[18] Yao Y, Huang L, Sharma AB, Golubchik L, Neely MJ. Power cost reduction in distributed data centers: A two-time-scale approach for delay tolerant workloads. IEEE Trans Parallel Distrib Syst 2014;25:200–11.
[19] Li C. Fuxi 2.0: the core scheduling system of Alibaba facing challenges of big data and cloud computing, https://developer.aliyun.com/article/760062; 2020.
[20] Dahiya P. Tasks in real time systems, https://www.geeksforgeeks.org/tasks-in-real-time-systems/; 2019.
[21] Lee J, Lee S, Kim H. Scheduling soft aperiodic tasks in adaptable fixed-priority systems. SIGOPS Oper Syst Rev 1996;30:17–28.
[22] Guo J, Chang Z, Wang S, Ding H, Feng Y, Mao L, et al. Who limits the resource efficiency of my datacenter: An analysis of alibaba datacenter traces. In: 2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS): IEEE; 2019. p. 1–10.
[23] Lannoye E, Flynn D, Malley MO. Evaluation of Power System Flexibility. IEEE Trans Power Syst 2012;27:922–31.
[24] Chen Y, Chen Z, Xu P, Li W, Sha H, Yang Z, et al. Quantification of electricity flexibility in demand response: Office building case study. Energy. 2019;188: 116054.1-.18.
[25] Cioara T, Anghel I, Salomie I, Antal M, Pop C, Bertoncini M, et al. Exploiting data centres energy flexibility in smart cities: Business scenarios. Inf Sci 2019;476: 392–412.
[26] Chen S, Li P, Ji H, Yu H, Yan J, Wu J, et al. Operational flexibility of active distribution networks with the potential from data centers. Appl Energy 2021;293: 116935.
[27] Qureshi A, Weber R, Balakrishnan H, Guttag J, Maggs B. Cutting the electric bill for internet-scale systems. In: Proceedings of the ACM SIGCOMM 2009 conference on Data communication 2009. p. 123-34.
[28] Rao L, Liu X, Xie L, Liu W. Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment. In: INFOCOM, 2010 Proceedings IEEE: IEEE; 2010. p. 1-9.
[29] Guenter B, Jain N, Williams C. Managing cost, performance, and reliability tradeoffs for energy-aware server provisioning. 2011 Proceedings IEEE INFOCOM2011. p. 1332-40.
[30] Ren S. Batch job scheduling for Reducing Water footprints in data center. 2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)2013. p. 747-54.
[31] Ding Z, Lee WJ. A Stochastic Microgrid Operation Scheme to Balance Between System Reliability and Greenhouse Gas Emission. IEEE Trans Ind Appl 2016;52: 1157–66.
[32] Dou H, Qi Y, Wei W, Song H. Carbon-Aware Electricity Cost Minimization for Sustainable Data Centers. IEEE Trans Sustainable Comput 2017;2:211–23.
[33] Chen T, Wang X, Giannakis GB. Cooling-Aware Energy and Workload Management in Data Centers via Stochastic Optimization. IEEE J Sel Top Signal Process 2016;10: 402–15.
[34] Nadjahi C, Louahlia H, Lemasson S. A review of thermal management and innovative cooling strategies for data center. Sustainable Comput Inf Syst 2018.
[35] Le K, Bianchini R, Nguyen TD, Bilgir O, Martonosi M. Capping the brown energy consumption of Internet services at low cost. In: International Conference on Green Computing 2010.
[36] Liu Z, Lin M, Wierman A, Low S, Andrew LLH. Greening geographical load balancing. Acm Sigmetrics. Performance Evaluation Review 2011:657–71.
[37] Goiri Í, Le K, Haque ME, Beauchea R, Nguyen TD, Guitart J, et al. Greenslot: scheduling energy consumption in green datacenters. In: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis; 2011. p. 1–11.
[38] Zhang Y, Wang Y, Wang X. GreenWare: greening cloud-scale data centers to maximize the use of renewable energy. In: Proceedings of the 12th ACM/IFIP/USENIX international conference on Middleware. Lisbon, Portugal: Springer-Verlag; 2011. p. 143–64.
[39] Chen C, He B, Tang X. Green-aware workload scheduling in geographically distributed data centers. In: 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings; 2012. p. 82–9.
[40] Wang P, Cao Y, Ding Z. Flexible Multi-Energy Scheduling Scheme for Data Center to Facilitate Wind Power Integration. IEEE Access 2020;8:88876–91.
[41] Rasley J, Karanasos K, Kandula S, Fonseca R, Vojnovic M, Rao S. Efficient queue management for cluster scheduling. ACM2016. p. 1-15.

[42] Inc A. Cluster data collected from production clusters in alibaba for cluster management research, https://github.com/alibaba/clusterdata; 2017.
[43] Cheng Y, Anwar A, Duan X. Analyzing Alibaba's Co-located Datacenter Workloads. In: 2018 IEEE International Conference on Big Data (Big Data)2018.
[44] Verma A, Pedrosa L, Korupolu M, Oppenheimer D, Tune E, Wilkes J. Large-scale cluster management at Google with Borg. In: Proceedings of the Tenth European Conference on Computer Systems; 2015. p. 1–17.
[45] Cheung H, Wang S, Zhuang C, Gu J. A simplified power consumption model of information technology (IT) equipment in data centers for energy system real-time dynamic simulation. Appl Energy 2018;222:329–42.
[46] Ebrahimi K, Jones GF, Fleischer AS. A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities. Renew Sustain Energy Rev 2014;31:622–38.
[47] Dayarathna M, Wen Y, Fan R. Data Center Energy Consumption Modeling: A Survey. IEEE Commun Surv Tutorials 2017;18:732–94.
[48] Zhang P, Li Y, Lin H, Wang J, Zhang C. A Periodic Task-Oriented Scheduling Architecture in Cloud Computing. 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications. Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom): IEEE 2018: 788–94.
[49] Verma A, Pedrosa L, Korupolu M, Oppenheimer D, Tune E, Wilkes J. Large-scale cluster management at Google with Borg. Bordeaux, France: Association for Computing Machinery; 2015.
[50] Cortez E, Bonde A, Muzio A, Russinovich M, Fontoura M, Bianchini R. In: Resource Central: Understanding and Predicting Workloads for Improved Resource Management in Large Cloud Platforms. Shanghai, China: Association for Computing Machinery; 2017. p. 153–67.
[51] Tang S, Yuan J, Wang C, Li X-Y. A framework for amazon ec2 bidding strategy under sla constraints. IEEE Trans Parallel Distrib Syst 2013;25:2–11.
[52] Ilyushkin A, Ghit B, Epema D, Scheduling workloads of workflows with unknown task runtimes.. 15th IEEE/ACM International Symposium on Cluster. Cloud and Grid Computing: IEEE 2015;2015:606–16.
[53] Jyothi SA, Curino C, Menache I, Narayanamurthy SM, Tumanov A, Yaniv J, et al. Morpheus: Towards automated slos for enterprise clusters. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)2016. p. 117-34.
[54] Basak J, Wadhwani K, Voruganti K. Storage Workload Identification. ACM Trans Storage 2016;12:1–30.
[55] Hirales-Carbajal A, González-García J-L, Tchernykh A, México BC. Workload Generation for Trace Based Grid Simulations. carbajal.
[56] Fiskiran M. Workload characterization of elliptic curve cryptography and other network security algorithms for constrained environments. In: 2002 IEEE International Workshop on Workload Characterization: IEEE; 2002. p. 127–37.
[57] Arachchige CN, Prendergast LA, Staudte RG. Robust analogs to the coefficient of variation. Journal of Applied Statistics 2022;49:268–90.
[58] Selzer RH, Mack WJ, Lee PL, Kwong-Fu H, Hodis HN. Improved common carotid elasticity and intima-media thickness measurements from computer analysis of sequential ultrasound frames. Atherosclerosis 2001;154:185–93.
[59] Molka, Hackenberg, Schone, Muller. Characterizing the energy consumption of data transfers and arithmetic operations on x86-64 processors. In: International Conference on Green Computing2010.
[60] Karyakin A, Salem K. An analysis of memory power consumption in database systems. In: Proceedings of the 13th International Workshop on Data Management on New Hardware (DaMoN'17). Chicago, IL, USA: ACM; 2017. p. 1-9.
[61] Lewis AW, Ghosh S, Tzeng N-F. Run-time Energy Consumption Estimation Based on Workload in Server Systems. HotPower 2008;8:17–21.
[62] Robert H-N. Theory of the backpropagation neural network. Neural Networks 1988.
[63] Chen F, Grundy J, Yang Y, Schneider J-G, He Q. Experimental analysis of task-based energy consumption in cloud computing systems. In: Proceedings of the 4th acm/spec international conference on performance engineering2013. p. 295-306.
[64] Ghamkhari M, Wierman A, Mohsenian-Rad H. Energy Portfolio Optimization of Data Centers. IEEE Trans Smart Grid 2016;8:1–13.
[65] Lin W, Wu G, Wang X, Li K. An artificial neural network approach to power consumption model construction for servers in cloud data centers. IEEE Transactions on Sustainable Computing.PP:1-.
[66] Jiang C, Han G, Lin J, Jia G, Shi W, Wan J. Characteristics of Co-Allocated Online Services and Batch Jobs in Internet Data Centers: A Case Study from Alibaba Cloud. IEEE Access. 2019:1-.
[67] Carrión M, Arroyo JM. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. IEEE Trans Power Syst 2006;21:1371–8.
[68] Utama C, Troitzsch S, Thakur J. Demand-side flexibility and demand-side bidding for flexible loads in air-conditioned buildings. Appl Energy 2021;285:116418.
[69] Ren Z, Xu X, Wan J, Shi W, Zhou M. Workload characterization on a production hadoop cluster: A case study on taobao. In: 2012 IEEE International Symposium on Workload Characterization (IISWC): IEEE; 2012. p. 3–13.
[70] Mittal S, Sangwan OP. Big data analytics using machine learning techniques. In: 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence): IEEE; 2019. p. 203–7.
[71] Ding Z, Cao Y, Xie L, Lu Y, Wang P. Integrated stochastic energy management for data center microgrid considering waste heat recovery. IEEE Trans Ind Appl 2019; 55:2198–207.
[72] Liu Z, Liu I, Low S, Wierman A. Pricing data center demand response. Acm Sigmetrics. Performance Evaluation Review 2014.
[73] Dou H, Qi Y, Wei W, Song H. Minimizing electricity bills for geographically distributed data centers with renewable and cooling aware load balancing. In: 2015 International Conference on Identification, Information, and Knowledge in the internet of Things (IIKI): IEEE; 2015. p. 210–4.
[74] Schuller A, Flath CM, Gottwalt S. Quantifying load flexibility of electric vehicles for renewable energy integration. Appl Energy 2015;151:335–44.
[75] Takouna I, Dawoud W, Meinel C. Accurate mutlicore processor power models for power-aware resource management. In: 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing: IEEE; 2011. p. 419–26.
[76] Beloglazov A, Abawajy J, Buyya R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future generation computer systems 2012;28:755–68.
[77] Basmadjian R, Ali N, Niedermeier F, De Meer H, Giuliani G. A methodology to predict the power consumption of servers in data centres. In: Proceedings of the 2nd international conference on energy-efficient computing and networking2011. p. 1-10.
[78] Guo Y, Fang Y. Electricity cost saving strategy in data centers by using energy storage. IEEE Trans Parallel Distrib Syst 2012;24:1149–60.
[79] Yu L, Jiang T, Cao Y. Energy Cost Minimization for Distributed Internet Data Centers in Smart Microgrids Considering Power Outages. IEEE Trans Parallel Distrib Syst 2014;26:120–30.
[80] Tian H, Zheng Y, Wang W. Characterizing and synthesizing task dependencies of data-parallel jobs in alibaba cloud. In: Proceedings of the ACM Symposium on Cloud Computing; 2019. p. 139–51.