Bin-Packing scheduling of delay tolerant tasks for zero-carbon data centers

Yunfeng Peng[1], Wenjie Si[1], Yuhang Zou[1], Yunhao Zhang[1], Xueying Zhai[1], Xiuping Guo[2], and Wei Zhang[3]

[1]University of Science and Technology Beijing
[2]Beijing University of Posts and Telecommunications School of Economics and Management
[3]Qilu University of Technology

January 16, 2025

Abstract

Currently, the electricity to run cloud computers is usually generated from fossil fuels (e.g., petroleum, natural gas), which will cause carbon pollution. Therefore, data centers, as places to accommodate cloud computers, are now facing a serious problem of high carbon pollution. In this letter, an operational method is proposed to achieve zero-carbon data centers by carefully matching delay tolerant tasks to computing resources (e.g., CPU) when zero-carbon electricity (wind and solar energy) is available. We designed a unified measurement called CPU×Time, by which the complex matching problem involving tasks, computing resources, and zero-carbon electricity is simplified into a bin-packing scheduling. Simulations show that the proposed bin-packing scheduling method can achieve high resource utilization without carbon pollution.

Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

LETTER

Bin-Packing scheduling of delay tolerant tasks for zero-carbon data centers

Yunfeng Peng[1] | **Wenjie Si**[1] | **Yuhang Zou**[1] | **Yunhao Zhang**[1] | **Xueying Zhai**[1] | **Xiuping Guo**[2] | **Wei Zhang**[3]

[1]School of Computer & Communication Engineering, University of Science and Technology Beijing, Beijing, China

[2]School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing, China

[3]Key Laboratory of Computing Power Network and Information Security, Qilu University of Technology (Shandong Academy of Sciences), Jinanny, China

Correspondence

Corresponding author Yunfeng Peng.
Email: pengyf@ustb.edu.cn

Abstract

Currently, the electricity to run cloud computers is usually generated from fossil fuels (e.g., petroleum, natural gas), which will cause carbon pollution. Therefore, data centers, as places to accommodate cloud computers, are now facing a serious problem of high carbon pollution. In this letter, an operational method is proposed to achieve zero-carbon data centers by carefully matching delay tolerant tasks to computing resources (e.g., CPU) when zero-carbon electricity (wind and solar energy) is available. We designed a unified measurement called CPU×Time, by which the complex matching problem involving tasks, computing resources, and zero-carbon electricity is simplified into a bin-packing scheduling. Simulations show that the proposed bin-packing scheduling method can achieve high resource utilization without carbon pollution.

KEYWORDS

cloud computers, carbon pollution, delay tolerant task, zero-carbon electricity

1 | INTRODUCTION

In the era of digital economy, data centers have become a powerful engine to support innovation and development of cloud computing industry[1]. As a result, data centers will consume volumes of electricity energy to keep cloud computers working[2]. Because the supported electricity is mainly generated by carbon-emission plants burning coal, oil, and natural gas, data centers are indirectly becoming carbon-emission plants. For instance, in 2023, the total equivalent carbon emission of data centers in China reaches 163 million tons[3]. Therefore, to make use of the rich zero-carbon electricity generated by wind and solar energy in western China, a national project called Channeling Computing Resources from the East to the West is put forward in China (called the CCREW project in this paper), which will push zero-carbon data centers in China[4]. However, the wind and solar electricity (referred as zero-carbon electricity in this paper) depends on instable weather. Simultaneously, computing tasks from users are inherently instable because it is difficult to surely know who and when will use the computers in data centers. Unfortunately, they both will cause the availability of computing resources (e.g., CPUs) also fluctuate irregularly over time. Therefore, to find a good match among three instable parts (i.e., instable zero-carbon electricity, instable computing tasks, and instable computing resources) is complex, and we formulate it as a multi-dimensional uncertainty matching problem.

To deal with the instability of zero-carbon electricity, Khosravi et al. proposed a composite VM (Virtual Machine) placement approach by scheduling hybrid electricity to achieve low-carbon[5]. Abbas Kiani et al. proposed a workload distribution algorithm that maximizes the use of zero-carbon electricity to achieve low-carbon work through load distribution[6]. R. Tripathi et al. designed an electricity usage model to integrate zero-carbon electricity into data centers to reduce carbon footprint[7]. To solve the instability of computing tasks and computing resources, many works focused on harnessing algorithms, particularly machine learning, to achieve efficient scheduling by aligning tasks with available computing resources[8,9]. These complex algorithms depend heavily on parameter adjustment by trial and error, and historical data for training, which will easily fall into local optima and unreliable random solution.

Fortunately, we find that the computing tasks for the CCREW project are mainly delay tolerant, such as large model training, which can tolerate the computing latency in minutes, hours, even in days[10]. Consequently, the delay tolerant tasks can be easily matched to available computing resources during the time when zero-carbon electricity is available. Reference[11] demonstrates the feasibility and efficacy of this method. Based on the above ideas, a measurement spelled as CPU × Time is proposed to quantify the volume of a computing task and resources by multiplying the number of CPUs used and the time these CPUs worked. With the CPU × Time measurement, the tasks and the available computing resources will be modeled as items and rectangle bins respectively. Because the volume of each bin is restricted by the availability of zero-carbon electricity, the task can be allocated with available resources within the available time window of zero-carbon electricity.

The remainder of this letter is organized as follows: The proposed bin-packing scheduling method is presented in Section II, and its performance is simulated in Section III. Finally, in Section IV, we draw conclusions.

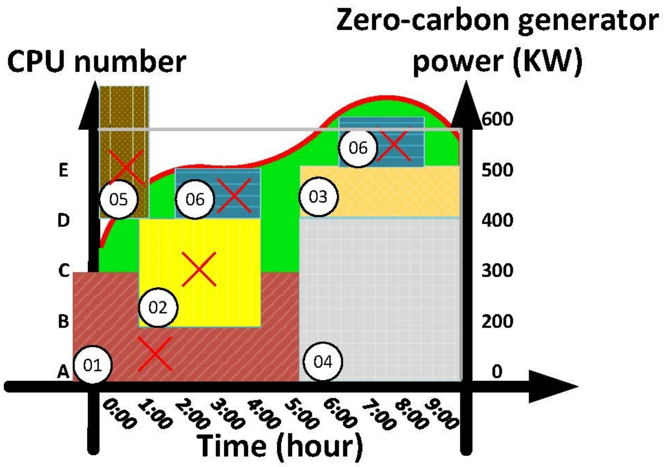

FIGURE 1 Schematic diagram of the bin-packing scheduling solutions for the multi-dimensional matching

2 | THE BIN-PACKING SCHEDULING METHOD

This section introduces the CPU × Time measurement, how to use the bin-packing scheduling method to solve the proposed multi-dimensional uncertainty matching problem. Based on the matching problem and bin-packing scheduling method, a mathematical model is established, which will be solved by Gurobi.

2.1 | CPU × Time measurement

The energy consumed by computers in data centers mainly comes from CPUs and memories[12] where the part for CPUs is usually more than that for memory, and the cost of memories is lower than that of CPUs. Therefore, in this work, we assume that memories are always available, and we do not consider memory availability but its energy consumption. The volumes of computing resources and computing tasks both can be unified by the measurement CPU × Time. For one piece of computing resource, its volume can be measured by the number of available CPUs and their available time. For example, when a data center has 5 available CPUs from 0:00 to 9:00 as illustrated in Figure 2, we model these computing resources as a rectangle bin with volume as a size of 5 × 9. Similarly, for a computing task, its volume can be predicted based on its inherent attribute[13] and measured by the number of CPUs assigned and their working time. As an example, illustrated in Fig. 2, where a task No.4 is assigned 3 CPUs and each will work 4 hours, then we quantify this task as a rectangle item with volume of 3 × 4, where the CPU side of the rectangle is 3 and the Time side is 4.

The electricity to fuel computers in data centers, can also be measured by CPU × Time × P, where P denotes the working power for a CPU quantified in watt. When all CPUs are working at the same power P, the electricity consumed by these CPUs can be measured by CPU × Time. Consequently, the size of the aforementioned rectangular can also be used to quantify the electricity used and the zero-carbon power supply. And in our method, the data center should forecast the zero-carbon electricity generation over an upcoming period based on weather predictions (i.e., wind force and solar radiation)[14]. For this way, if the zero-carbon electricity supply from 3:00 p.m. to 4:00 p.m. can afford up to four computers with the same power P, the maximum allowable size of total items to be placed in bin within this time span would be 4 × 1.

2.2 | Bin-packing scheduling process

The process of solving the matching problem through the Bin-packing scheduling method is divided into six stages: shown in Figure 2.

Step 1 Collect data of tasks, computing resources, and zero-carbon electricity from users, data center operators, and grid operators respectively.

Step 2 Translate the data into items or bin using the united CPU × Time measurement.

Step 3 Set CPU side as the vertical axis and set the time side as the horizontal axis, by which, the items and bin are

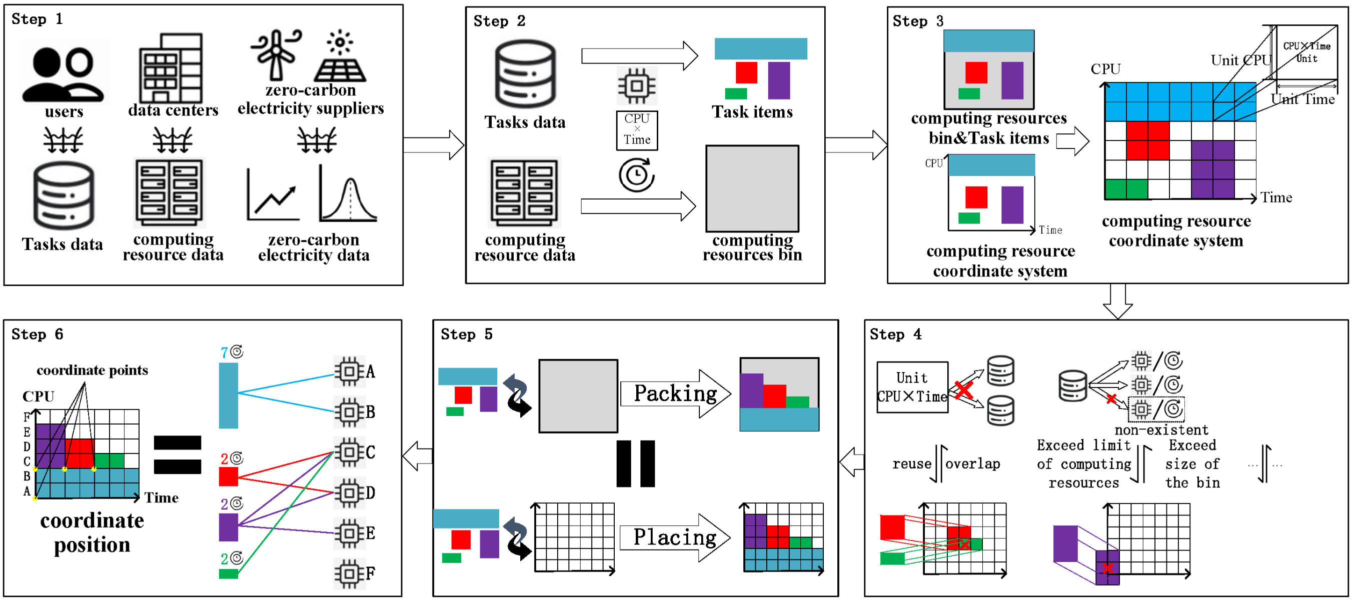

FIGURE 2 The process of bin-packing scheduling method.

mapped as a planar coordinate system and divided into CPU × Time units.

Step 4 Match constraints to bin-packing constraints.(Reference section 2.3 for detail).

Step 5 Use Gurobi to place items in bin and obtain their position information in coordinate system meeting constrains.

Step 6 Translate the bin-packing scheduling result back to the multi-dimensional matching solution. Taking Figure 1 as an example, five CPUs labeled as A to E and their working time in hours are resources to be matched to items indexed as 01 to 06. Item No. 3 in Figure 1 occupies four Unit Time and one Unit CPU, with its bottom-left coordinate point is (5,3), which means Task No. 3 is processed by CPU D from 5:00 (to 9:00).

TABLE 1 The symbols used in the model.

Symbol	Definition
T_i	number task i
P_j	number time slice j
C_k	number CPU k
NP	total time slice of scheduling cycle
NC	the total amount of CPU
NI	the total amount of task
CC_i	required cpus for task i
CT_i	completed time for task i
Pr_i	Priority for completing task i
P_i	the energy consumption of task i
E_j	available zero-carbon electricity at time silce j
$x_{i,j,k}$	task i is processed by begin CPU k at begin time slice j

2.3 | Formulation of Bin-packing scheduling problem

We formulate the bin-packing problem solved by Gurobi as a mathematical model. Table1 lists the symbols used in the model.

We set maximize the quility of completing tasks in the Bin-packing scheduling problem as the objective function. In CCREW, some tasks are strong delay-tolerant with long deadline, while others are weak delay-tolerant needed to be scheduled quickly. We set Pr_i to reflect the priority of tasks. This means that the objective function needs to maximize the sum of the Pr_i values for packing items. In the model, it is reflected as (1):

$$\max \sum_{i=1}^{NI} \sum_{j=1}^{NP} \sum_{k=1}^{NC} (x_{i,j,k} \times Pr_i) \tag{1}$$

Constraint 1 (Excess constraints)**.**

The time span of a task must be scheduled within the CPU available time window. And the number of CPUs allocated for all simultaneous tasks cannot exceed the number of available CPUs. This means that items cannot go out of the boundary of the bin. As illustrated in Figure 1, Task i = 1 is scheduled before the CPU available time, so it makes a packing error.

In the model, it is reflected as (2):

$$\begin{aligned}\sum_{j=NP-CT_i}^{NP} x_{i,j,k} &= 0, \forall k \in NC \forall i \in NI \\ \sum_{k=NC-CC_i}^{NC} x_{i,j,k} &= 0, \forall j \in NP \forall i \in NI\end{aligned} \tag{2}$$

Constraint 2 (Task match constraints)**.**

Each task can only be matched once. As illustrated in Figure 1, Task $i = 6$ is matched twice, so it makes a packing error.

In the model, it is reflected as (3):

$$\sum_{j=1}^{NP} \sum_{k=1}^{NC} x_{i,j,k} \leq 1, \forall i \in NI \tag{3}$$

Constraint 3 (CPU usage constraints)**.**

One CPU should not work for more than one tasks at the same time. This means that items cannot overlap each other. As in Figure 1, Task $i = 2$ shares CPU $k = 1$ with task $i = 1$ from time $j = 1$ to $j = 4$, so they make a packing error.

In the model, it is reflected as (4):

$$\sum_{i=1}^{NI} \sum_{k0=k-CC_i+1}^{k} \sum_{j0=j-CT_i+1}^{j} x_{i,j0,k0} \leq 1, \forall k \in NC, \forall j \in NP \tag{4}$$

Constraint 4 (Zero-carbon electricity constraints)**.**

The scheduled working time must be fall inside the available time window of zero-carbon electricity. As illustrated in Figure 1, the red line is the forecast zero-carbon generator power. So, the green area is the available zero-carbon electricity according to the concept of calculus. The electricity consumed by processing tasks at any time must be less than the current available zero-carbon electricity. This means that at any time slice, the total area of all items must be less than the green coverage area. The $c = 6$ CPU does not exist. And Task $i = 5$ results in the electricity consumption from $t = 0$ to $t = 1$ to be larger than current zero-carbon electricity, so it makes a packing error.

In the model, it is reflected as (5):

$$\sum_{i=1}^{NI} \frac{\left(\sum_{j0=j-CT_i+1}^{j} \sum_{k=1}^{NC} x_{i,j0,k}\right) \times P_i}{CC_i} \leq E_j, \forall j \in NP \tag{5}$$

3 | EVALUATION

We simulate the proposed bin-packing scheduling with the objective of maximizing the number of tasks packed by Gurobi 11.02 Optimization Tools. The number of CPUs required and their working time for 154 tasks(NI=154) is taken from the open-source data set of Alibaba[15]. The power of each single-CPU computers is 193.3 W according to literature[16]. Each single time slice is set to 30 minutes and total simulation time is set to 24 hours(NP=48). Based on the zero-carbon electricity prediction model in literature[17], we produced available zero-carbon electricity as shown in Figure 4 following the zero-carbon energy data set in literature[18].

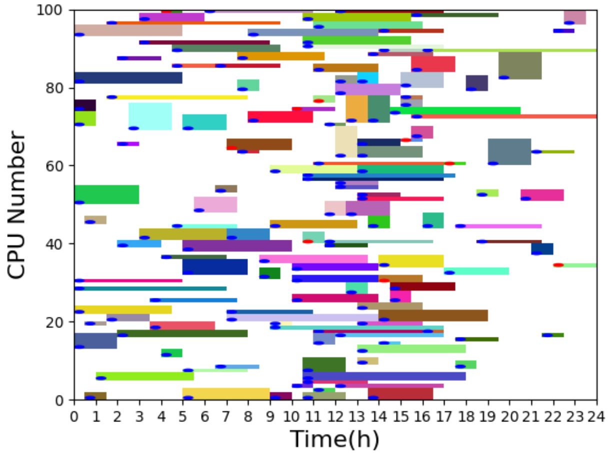

FIGURE 3 The matching scheme of tasks to available resources

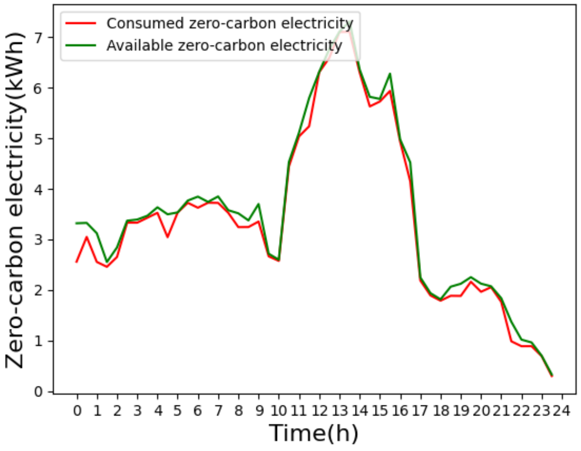

FIGURE 4 Utilization of zero-carbon electricity

TABLE 2 Carbon Emission Parameter

Type of energy	Carbon emission parameter
Natural gas	440 g/(kWh)
Petroleum	890 g/(kWh)

As illustrated in Figure 3 Items are distinguished by color. The bottom-left cells of the high-priority or low-priority items

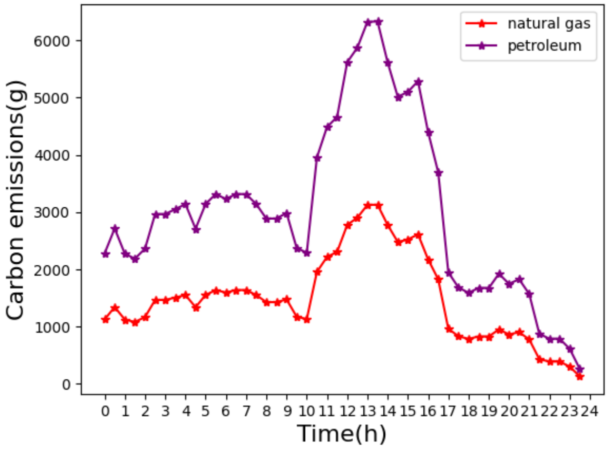

FIGURE 5 Carbon emissions

are marked with red circle or blue circle, respectively. The blank part represents the unused time slice of computing resources. Based on the ratio of the area occupied by the item to the total area of the bin, we get the resource utilization efficiency of 34.4%. Items has a completion rate of 94.8%. In Figure 4, red line represents the consumed zero-carbon electricity by processing tasks. The 24-hours total available zero-carbon electricity is 171.88 kWh, of which the 162.72 kWh electricity is used. The zero-carbon power average utilization rate is 94.67%.

The carbon emission parameters are listed in Table 2[19]. If the 154 tasks are processed in 24-hour by a data center powered by plants burning fossil fuels, the carbon emissions generated per hour would be as shown in Figure 5. The 24-hour cumulative of the total carbon emissions are respectively 71595.18 g with natural gas or 144817.53 g with petroleum.

4 | CONCLUSIONS

To solve the multi-dimensional uncertainty matching problem for zero-carbon data centers in CCREW project, the CPU × Time measurement is proposed, by which the complex problem is simplified into a bin-packing scheduling.

Considering all data centers across China will be connected by CCREW in the future, the method can be extended to schedule tasks, resources, and zero-carbon electricity among multiple data centers.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENTDATA

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

1. Tang X, Cao C, Wang Y, et al. Computing power network: The architecture of convergence of computing and networking towards 6G requirement. *China Communications.* 2021;18(2):175–185.
2. Buyya R, Ilager S, Arroba P. Energy-efficiency and sustainability in new generation cloud computing: A vision and directions for integrated management of data centre resources and workloads. *Softw: Pract Exper.* 2024;54(1):24–38. doi: 10.1002/spe.3248
3. Chen X, Cao L, Chen J, Zhang J, Cao W, Wang Y. Development demand, power energy consumption and green and low-carbon transition for computing power in China. *Bulletin of Chinese Academy of Sciences.* 2024;39(3):528-539. doi: 10.16418/j.issn.1000-3045.20230717002
4. Guo J. Five departments issue documents to accelerate the construction of a national integrated computing power network, which is conducive to accelerating the promotion of the CCREW' project. Tech. Rep. 1, Securities Daily; 2023.
5. Khosravi A, Andrew LLH, Buyya R. Dynamic VM Placement Method for Minimizing Energy and Carbon Cost in Geographically Distributed Cloud Data Centers. *IEEE Transactions on Sustainable Computing.* 2017;2(2):183-196. doi: 10.1109/TSUSC.2017.2709980
6. Kiani A, Ansari N. Toward Low-Cost Workload Distribution for Integrated Green Data Centers. *IEEE Communications Letters.* 2015;19(1):26-29. doi: 10.1109/LCOMM.2014.2369459
7. Tripathi R, Vignesh S, Tamarapalli V. Optimizing Green Energy, Cost, and Availability in Distributed Data Centers. *IEEE Communications Letters.* 2017;21(3):500-503. doi: 10.1109/LCOMM.2016.2631466
8. Belgacem A, Beghdad-Bey K, Nacer H, Bouznad S. Efficient dynamic resource allocation method for cloud computing environment. 2020
9. Lin QL, Yu SZ. A Distributed Green Networking Approach for Data Center Networks. *IEEE Communications Letters.* 2017;21(4):797-800. doi: 10.1109/LCOMM.2016.2642188
10. Wu HQ. The 'Mathematical Problems' and 'Arithmetic Problems' in the Development of Computing Power under the 'CCREW'" (in Chinese). *Science News.* 2022(5):11-13. doi: 10.1007/s10586-020-03053-x
11. Zhai X, Peng Y, Guo X. Edge-cloud collaboration for low-latency, low-carbon, and cost-efficient operations. *Computers and Electrical Engineering.* 2024;120:109758. doi: https://doi.org/10.1016/j.compeleceng.2024.109758
12. Vasques T, Moura P, Almeida dA. A review on energy efficiency and demand response with focus on small and medium data centers. *Energy Efficiency.* 2019:1399–1428. doi: 10.1007/s12053-018-9753-2
13. Xu D. Research on Resource Demand Prediction and Optimization Configuration Methods in Cloud Computing Environment. 2014. doi: 10.7666/d.Y2539016
14. Zhang Y, Wang Y, Wang X. GreenWare: Greening Cloud-Scale Data Centers to Maximize the Use of Renewable Energy. In: Kon F, Kermarrec AM., eds. *Middleware 2011*Springer Berlin Heidelberg 2011; Berlin, Heidelberg:143–164.
15. Inc A. Alibaba production cluster data v2018. .
16. Linfeng Z, Tong W, Qingfei S, et al. Electric Power Information and Communication Technology. *Research and Application of Testing Method for Actual Operating Power of Server (in Chinese).* 2021;19(6):64-69. doi: 10.16543/j.2095-641x.electric.power.ict.2021.06.010
17. Aksanli B, Venkatesh J, Zhang L, Rosing T. Utilizing green energy prediction to schedule mixed batch and service jobs in data centers. *SIGOPS Oper. Syst. Rev..* 2012;45(3):53–57. doi: 10.1145/2094091.2094105
18. Chen Y, Xu J. Solar and wind power data from the Chinese State Grid Renewable Energy Generation Forecasting Competition. *Sci Data.* 2022;9(1):577–577. doi: 10.1038/s41597-022-01696-6
19. W. Deng FLHJ, Li D. New Energy Applications in Cloud Computing Data Centers: Research Status and Trends*inChinese*. *Journal of Computer Science.* 2013;36(3):582-598. doi: 10.1038/s41597-022-01696-6