

# An IoT-Oriented Data Placement Method with Privacy Preservation in Cloud Environment

Xiaolong Xu<sup>a,b,c</sup>, Shucun Fu<sup>a,b</sup>, Lianyong Qi<sup>d,\*</sup>, Xuyun Zhang<sup>e</sup>, Qingxiang Liu<sup>a,b</sup>, Qiang He<sup>f</sup>, Shancang Li<sup>g</sup>

<sup>a</sup>*School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China*

<sup>b</sup>*Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing, China*

<sup>c</sup>*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China*

<sup>d</sup>*School of Information Science and Engineering, Qufu Normal University, China*

<sup>e</sup>*Department of Electrical and Computer Engineering, University of Auckland, New Zealand*

<sup>f</sup>*Faculty of Information and Communication Technologies, Swinburne University of Technology*

<sup>g</sup>*Computer Science and Creative Technologies Department, University of the West of England*

---

## Abstract

IoT (Internet of Things) devices generate huge amount of data which require rich resources for data storage and processing. Cloud computing is one of the most popular paradigms to accommodate such IoT data. However, the privacy conflicts combined in the IoT data makes the data placement problem more complicated, and the resource manager needs to take into account the resource efficiency, the power consumption of cloud data centers, and the data access time for the IoT applications while allocating the resources for the IoT data. In view of this challenge, an IoT-oriented Data Placement method with privacy preservation, named IDP, is designed in this paper. Technically, the resource utilization, energy consumption and data access time in the cloud data center with the fat-tree topology are analyzed first. Then a corresponding data placement method, based on the Non-dominated Sorting Genetic Algorithm II

---

\*Corresponding author

*Email addresses:* [njuxlxu@gmail.com](mailto:njuxlxu@gmail.com) (Xiaolong Xu), [shucunfu@gmail.com](mailto:shucunfu@gmail.com) (Shucun Fu), [lianyongqi@gmail.com](mailto:lianyongqi@gmail.com) (Lianyong Qi), [xuyun.zhang@auckland.ac.nz](mailto:xuyun.zhang@auckland.ac.nz) (Xuyun Zhang), [qingxiangliu737@gmail.com](mailto:qingxiangliu737@gmail.com) (Qingxiang Liu), [qhe@swin.edu.au](mailto:qhe@swin.edu.au) (Qiang He), [Shancang.Li@uwe.ac.uk](mailto:Shancang.Li@uwe.ac.uk) (Shancang Li)

(NSGA-II), is designed to achieve high resource usage, energy saving and efficient data access, and meanwhile realize privacy preservation of the IoT data. Finally, extensive experimental evaluations validate the efficiency and effectiveness of our proposed method.

*Keywords:* IoT, data placement, privacy, energy consumption, cloud

---

## 1. Introduction

Nowadays, Internet of Things (IoT) has been used in many areas including transportation, nursing, monitoring, etc., where smart devices equipped with sensors are connected together via the Internet to efficiently collect data from physical environment [1] [2] [3]. IoT devices produce large amount of real-time data, which is stored for analysis and processing to guarantee the service requirements of IOT applications [4]. However, due to the storage limitation of IoT devices, IoT data requires being placed on the remote data centers for further processing. Generally, data placement is divided into random placement and planned placement. In random placement, the sensors are scattered randomly, otherwise the sensors are deployed selectively for certain objectives in planned placement [5].

Cloud computing provides elastic and on-demand resources for IoT-oriented data placement. Compared to the IoT devices, cloud data centers (CDCs) have much more physical resources including storage resources, computing resources and communication resources [6] [7] [8]. Deploying IoT data on CDCs alleviates resource limitation of the IoT devices in a great deal. When allocating the cloud resources to accommodate IoT data, the resources are provided in the form of virtual machines (VMs). The resource managers monitor current cloud resource usage and map the VMs to physical hosts [9]. From the perspective of resource management, the resource utilization needs to be taken into consideration to avoid the overload or underload of CDCs.

However, with the cloud computing widespread used to support the IoT applications, it is a popular trend to deploy the IoT data on the cloud, which makes

25 the energy consumption of CDCs greatly increased. Moreover, the expenditure  
of electricity in CDCs has a serious impact on the emission of carbon diox-  
ide which plays an important part in ecological balance. Therefore, to reduce  
carbon emission and realize sustainable development of cloud computing, it is  
of great importance to layout the IoT data on the cloud in an energy-efficient  
30 manner.

On the other hand, the collected IoT data stored on the cloud is in need  
of security [10] [11] [12]. If the IoT data contains private information such as  
healthcare, finance, etc., when placing the data with privacy conflicts combined  
on the same CDC, the data owners' privacy is easy to be invaded [13] [14] [15].  
35 Data encryption is the straightforward way to deal with it, but it may cost  
much more time and energy without a legitimate placement method [16] [17].  
Hence, it is necessary to satisfy the privacy constraints for IoT-oriented data  
placement.

Energy-efficient network architecture has been investigated for IoT appli-  
40 cations [18], but it neglects the resource utilization of CDCs, access time and  
privacy for IoT data placement. With the observation above, it is still a chal-  
lenge to realize IoT data placement to optimize the resource utilization, energy  
consumption and access time, while satisfying the privacy constraints. In view  
of this challenge, an IoT-oriented data placement method with privacy preser-  
45 vation in cloud environment is proposed this paper.

The main contributions of this paper include the following:

- (1) Analyze the resource utilization, energy consumption and access time of  
CDCs and build a corresponding systematic model while satisfying the privacy  
constraints of IoT data.
- 50 (2) Employ Non-dominated Sorting Genetic Algorithm II (NSGA-II) to im-  
prove the average resource utilization of CDCs and reduce the energy consump-  
tion of CDCs while considering the privacy preservation in cloud environment.
- (3) Conduct adequate experimental evaluation and comparison analysis to  
validate the efficiency and effectiveness of our proposed method.

55 The rest of this paper is organized as follows. In Section 2, formalized

concepts, a systematic model of privacy in data placement, resource utilization, energy consumption and access time of CDCs is presented. Section 3 elaborates the proposed data placement method. Section 4 illustrates the comparison analysis and performance evaluation. Section 5 summarizes the related work, and Section 6 concludes the paper and presents the future work.

## 2. Preliminary Knowledge

Table 1: Key Notations and Descriptions

| Notation | Description                                      |
|----------|--|
| $D$      | The dataset collection obtained from IoT devices |
| $N$      | The number of datasets                           |
| $A$      | The IoT application collection                   |
| $Q$      | The number of IoT applications                   |
| $M$      | The number of pods in fat tree                   |
| $P$      | The pod collection                               |
| $H$      | The physical host collection                     |
| $W$      | The number of physical hosts                     |
| $X$      | The data placement policy collection for $D$     |
| $Y$      | The deployment policy collection for $A$         |

In this section, we present a system model that closely approximates the storage environment for the datasets collected from the IoT devices in fat-tree constructed CDC. Some key notations and descriptions used in the paper are listed in Table 1.

### 2.1. System Model

We focus on the IoT-oriented data placement in the cloud environment that is a cloud platform providing elastic resources for storing the datasets from the IoT devices. Suppose there are  $N$  datasets collected from IoT devices, denoted

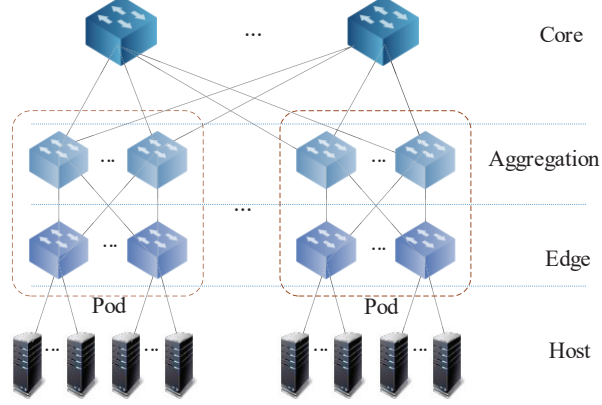


Figure 1: Fat-tree topology in CDCs.

as  $D = \{d_1, d_2, \dots, d_N\}$ , that need to be deployed in the CDC. And there  
 are  $Q$  IoT applications, denoted as  $A = \{a_1, a_2, \dots, a_Q\}$ , running in the data  
 center which requires multiple datasets from  $D$  for processing support. The  
 CDC employs the fat-tree topology to organize the physical hosts and switches,  
 as illustrated in Fig. 1. Through fat-tree topology, the CDC achieves timely  
 processing of loads, avoids network hotspots by multiple links at the core layer,  
 and eliminates overload by reasonably diverting traffic within pods.

In the fat-tree topology, the switches are classified as three layers, including  
 the core, aggregation and the edge. The switches in the aggregation and edge  
 layer are divided to several pods, denoted as  $P = \{p_1, p_2, \dots, p_M\}$ . According  
 to the rules of fat-tree topology, the number of the aggregation switches is  
 equal to the number of the edge switches in each pod, and they are both set  
 to  $(\frac{M}{2})^2$ . Besides, the number of core switches is set to  $\frac{M}{2}$ , and the number  
 of ports of each switch is set to  $\frac{M}{2}$ . The physical hosts are connected to the  
 edge servers and suppose the CDC consists of  $W$  physical hosts, denoted as  
 $H = \{h_1, h_2, \dots, h_W\}$ .

Let  $X = \{x_1, x_2, \dots, x_N\}$  be the placement policy for the datasets in  $D$ ,  
 where  $x_n \in H$  ( $n = \{1, 2, \dots, N\}$ ) is the host which the dataset  $d_n$  is placed  
 on. Let  $Y = \{y_1, y_2, \dots, y_Q\}$  be the deployment policy for the application set

$A$ , where  $y_q \in H (q = \{1, 2, \dots, Q\})$  is the host which the application  $a_q$  is  
90 deployed on.

## 2.2. Resource Utilization Model

In the CDC, the applications and datasets are hosted by VMs. In a cloud platform, there are multiple VM instances created for resource provisioning. The resource requirements of the datasets and the capacity of the hosts both  
95 could be quantified by the number of VM instances. Let  $c_w$  be the capacity of the  $w$ -th host  $h_w$  and  $\lambda_n$  be the requirements of the dataset  $d_n$ .

The resource utilization is a key metric for the resource managers to manage the CDCs. According to the data placement policy in  $X$ , the resource utilization of each host could be detected by detecting the resource usage of the VM instances. Let  $u_w(X)$  be the resource utilization with  $X$ , which is calculated by

$$u_w(X) = \frac{1}{c_w} \sum_{n=1}^N \theta_n \cdot I_{n,w}(X), \quad (1)$$

where  $\theta_n$  is the requested amount of VM instances for  $d_n$ , and  $f = I_{n,w}$  is a binary variable to judge whether  $d_n$  is placed on  $h_w$ , which is measured by

$$I_{n,w}(X) = \begin{cases} 1, & \text{if } w = x_n, \\ 0, & \text{Otherwise.} \end{cases} \quad (2)$$

Then the overall resource utilization of the CDC refers to the resource usage status of all the occupied hosts. The amount of occupied hosts, denoted as  $\lambda(X)$ , is calculated by

$$\lambda(X) = \sum_{w=1}^W F_w(X), \quad (3)$$

where  $F_w(X)$  is a binary flag to judge whether  $h_w$  is occupied which is calculated by

$$l_{q,n} = \begin{cases} 1, & \text{if } a_q \text{ needs to access } d_n, \\ 0, & \text{Otherwise} \end{cases} \quad (4)$$

Then the average resource utilization of the CDC is calculated by

$$U(X) = \frac{1}{\lambda(X)} \sum_{w=1}^W u_w(X). \quad (5)$$

### 2.3. Data Access Model

The IoT applications need to access and analyze the stored IoT data deployed in the cloud platforms. Generally, the IoT applications are also deployed in the cloud, thus while considering data placement for the IoT datasets, the access time for the IoT applications to obtain the datasets should be taken into account.

Let  $l_{q,n}$  be the flag to judge whether  $a_q$  needs to access  $d_n$ , which is measured by

$$l_{q,n} = \begin{cases} 1, & \text{if } a_q \text{ needs to access } d_n, \\ 0, & \text{Otherwise,} \end{cases} \quad (6)$$

where  $l_{q,n} = 1$  means  $a_q$  needs to access  $d_n$ , while  $l_{q,n} = 0$  means  $a_q$  need not to access  $d_n$ .

Denote  $g_{q,n}$  as the access frequency for  $a_q$  to get  $d_n$  during the execution period, and then the total access time in this period is calculated by

$$K = \sum_{q=1}^Q \sum_{n=1}^N l_{q,n} \cdot g_{q,n}. \quad (7)$$

In the fat-tree topology, the access time of the datasets for the IoT applications is closely relevant to the distributed locations among the datasets and the IoT applications. Let  $ES(h)$  be the edge switch which a physical host  $h$  connect to. Let  $Pod(h)$  be the pod where  $ES(h)$  locates. The data access time in the fat tree could be classified as four conditions: 1) the application  $a_q$  and the dataset  $d_n$  are deployed in the same host, i.e.,  $x_n = y_q$ . 2) the hosts for accommodating  $a_q$  and  $d_n$  are different and they both connect to the same edge switch, i.e.,  $x_n \neq y_q$ , and  $ES(x_n) = ES(y_q)$ . 3) the hosts for accommodating  $a_q$  and  $d_n$  are connected to different edge switches, but they are in the same pod, i.e.,  $ES(x_n) \neq ES(y_q)$ . and  $P(x_n) = P(y_q)$ . 4) the hosts for accommodating  $a_q$  and  $d_n$  are not in the same pod, i.e.,  $Pod(x_n) \neq Pod(y_q)$ . Based on the above analysis and the fat-tree topology, the data access time  $t_{q,n}(X)$  for  $a_q$  to

access  $d_n$  is calculated as

$$t_{q,n}(X) = \begin{cases} 0, & \text{if } x_n = y_q, \\ 2\frac{\rho_n}{b_{SE}} \cdot l_{q,n}, & \text{if } x_n \neq y_q, ES(x_n) = ES(y_q), \\ (2\frac{\rho_n}{b_{SE}} + 2\frac{\rho_n}{b_{EA}}) \cdot l_{q,n}, & \text{if } ES(x_n) \neq ES(y_q), P(x_n) = P(y_q), \\ (2\frac{\rho_n}{b_{SE}} + 2\frac{\rho_n}{b_{EA}} + 2\frac{\rho_n}{b_{AC}}) \cdot l_{q,n}, & \text{if } P(x_n) \neq P(y_q), \end{cases} \quad (8)$$

where  $\rho_n$  is the data size of  $d_n$ ,  $b_{SE}$  is the bandwidth between hosts and edge switches,  $b_{EA}$  is the bandwidth between edge switches and aggregation switches, and  $b_{AC}$  be the bandwidth between aggregation switches and core switches.

Then the average data access time  $T(X)$  can be calculated by

$$T(X) = \frac{1}{K} \sum_{n=1}^N \sum_{q=1}^Q t_{q,n}(X) \cdot g_{q,n}. \quad (9)$$

#### 2.4. Energy Consumption Model

In this paper, we mainly focus on the energy consumption for data placement and access generated by the physical hosts, the VMs and the switches.

All the running hosts consume the baseline energy during the tracked execution period  $R$  in the CDC. Such baseline energy consumption for  $H$  is denoted as  $BE_{Host}$ , calculated by

$$BE_{Host}(X) = \sum_{w=1}^W R \cdot \alpha_w \cdot F_w(X), \quad (10)$$

where  $\alpha_w$  is the baseline energy consumption rate of  $h_w$ .

The running VMs also consumes energy consumption, denoted as  $AE_{VM}$  which is calculated by

$$AE_{VM}(X) = \sum_{w=1}^W \sum_{n=1}^N \theta_n \cdot I_{n,w}(X) \cdot R \cdot \eta, \quad (11)$$

where  $\eta$  is the power rate for the running VM instances.

The energy consumption produced by the unused resource units, i.e., VM instances, also generate a certain amount of energy consumption, which is calculated by

$$SE_{VM}(X) = \sum_{w=1}^W (c_w - \sum_{n=1}^N \theta_n \cdot I_{n,w}(X)) \cdot R \cdot \tau. \quad (12)$$



Then the total energy consumption for all the VM instances in the CDC is calculated by

$$E_{VM} = SE_{VM}(X) + AE_{VM}(X). \quad (13)$$

In fat-tree topology, the total amount of switches in the topology is  $\frac{5}{4}M^2$  and from the access time model, the number of switches each dataset passes while invoking the IoT data from the cloud could be detected. The number of switches for  $a_q$  accessing  $d_n$  is calculated by

$$\delta_{q,n}(X) = \begin{cases} 0, & \text{if } x_n = y_q, \\ 1, & \text{if } x_n \neq y_q, ES(x_n) = ES(y_q), \\ 3, & \text{if } ES(x_n) \neq ES(y_q), P(x_n) = P(y_q), \\ 5, & \text{if } P(x_n) \neq P(y_q). \end{cases} \quad (14)$$

In this way, the energy consumption produced by switches, denoted as  $E_{switch}$ , is calculated by

$$E_{switch} = \frac{5}{4}k^2 \cdot R \cdot \beta + \sum_{q=1}^Q \sum_{n=1}^N l_{q,n} \cdot g_{q,n}(X) \cdot t_{q,n}(X) \cdot \delta_{q,n}(X) \cdot \gamma, \quad (15)$$

where  $\beta$  is the baseline power rate for each switch and  $\gamma$  is the power rate of each port.

Hence, the total energy consumption, denoted as  $E(X)$ , is calculated by

$$E(X) = BE_{Host} + E_{VE} + E_{Switch}. \quad (16)$$

## 2.5. Data privacy Model

115 There are several attributes in each IoT dataset. And some attributes of different datasets have privacy conflicts as once these attributes are stored in the same physical host, the combined privacy in these datasets will be leaked with meaningful probability.

120 We model the privacy conflicts of the datasets by a graph  $G = (D, E)$ , where  $D$  is the set of datasets,  $E$  represents the conflicting relations between two datasets in  $D$  and  $(d_n, d_{n'}) \in E$  represents there is a privacy conflict between  $d_n$  and  $d_{n'}$ .

Then the conflicting datasets of  $d_n$  is obtained by

$$cd_n = \{d_{n'} | (d_n, d_{n'}) \in E, n' = \{1, 2, \dots, N\}\}. \quad (17)$$

In this paper, to realize privacy preservation for the IoT datasets in cloud, the privacy-conflict datasets in  $CD_n$  could not be deployed with  $d_n$  in the same host. For thus we can obtain the conflicting location set of  $x_n$ , which is measured by

$$cl_n = \{x_j | x_j \in cd_n, j = \{1, 2, \dots, |cd_n|\}\}. \quad (18)$$

### 2.6. Problem Definition

In this paper, we focus on the data placement problem for IoT datasets in cloud environment, with fixed deployment policy for the IoT applications, to improve the resource utilization, reduce energy consumption and optimize the data access time while taking into account the data privacy constraints.

Then, the formalized problem is be defined as:

$$\max U(X), \min T(X), \min E(X). \quad (19)$$

$$s. \ t. \quad x_n = \{1, 2, \dots, W\}, \quad (20)$$

$$\sum_{n=1}^N \theta_n \cdot f_{n,w} \leq c_w, \quad (21)$$

$$x_n \notin cl_n. \quad (22)$$

## 3. An IoT-Oriented Data Placement Method with Privacy Preservation

An IoT-oriented data placement method with privacy preservation is proposed in this section. NSGA-II can find the global optimal solution in the feasible solution quickly and accurately compared with the traditional genetic algorithm. Thus, NSGA-II is adopted to solve the multi-objective optimization problem in this paper. First, the placement strategies for IoT datasets are encoded and fitness functions are given for the optimization problem. Second,

the *fast non-dominated sorting* approach and the *crowded-comparison* operation are used in selection. Then, the *crossover and mutation* operation of traditional genetic algorithm (GA) are adopted. Finally, the overview of our method is described in detail.

### 140 3.1. Encoding

In the GA, the gene represents the value of each decision variable. In this paper, a gene reflects the data placement strategy for a IoT dataset. A group of genes make up a chromosome which denotes a set of placement strategies of IoT datasets collection  $D$ . There are various methods to encode genes in genetic algorithms, including binary encoding, Gray code encoding, real encoding, 145 integer encoding, etc. We need to adaptively select different encoding methods according to the actual situation in different application scenarios. In this paper, as the strategy of data placement for each IoT dataset is an integer, the integer coding method is used and data placement strategies for IoT datasets are numbered as 0, 1, 2, ..., etc. 150

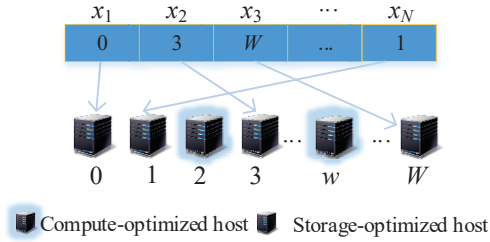


Figure 2: An encoding example of data placement for datasets.

As IoT applications need to access the placed IoT datasets in CDCs, two types of VM instances which are the computation-intensive optimized VM instances and the storage-optimized VM instances are employed in this paper. The datasets can only be placed on storage-optimized VM instances. Fig. 2 shows an example of data placement for IoT datasets. In Fig. 2, the blue shaded hosts represent computation-intensive optimized VM instances and the unshaded hosts represent storage-optimized VM instances. IoT applications have been placed 155

in computation-intensive optimized VM instances, and the IoT datasets have been chosen to be placed in those storage-optimized VM instances according to the encoding result of placement strategies.

### 3.2. Fitness Functions

The fitness function is a criterion for judging the merits of solutions in GA, each individual represents a solution of problems and all solutions form the sets called population. The fitness functions in this paper are given by Eq.(5), Eq.(9) and Eq.(16) which represent the average resources utilization, the average access time and the energy consumption. These fitness functions are indicators to measure the performance of CDCs.

The average resource utilization of hosts used is a fitness function which is calculated according to the coding result of data placement strategies for IoT datasets. Algorithm 1 specifies the process of calculating the average resources utilization of hosts used. In this algorithm, the inputs are IoT datasets collection  $D$ , IoT applications collection  $A$ , and the coding result of data placement strategies. We first calculate the resource utilization of each host used according to the input data placement strategy (Lines 3 to 6), then the average resource utilization of employed hosts is obtained (Line 12).

The average access time of IoT applications is another fitness function which is calculated according to the coding result of data placement strategies for IoT datasets and fat-tree topology of the CDC. Algorithm 2 specifies the process of calculating the average data access time of IoT applications. In this algorithm, a fat-tree network topology is employed to construct the cloud data center architecture. IoT applications and datasets share the same type of switch based on the assigned host seats. The access time of each application is calculated (Lines 2 to 5), then we get the average data access time (Line 10).

The energy consumption is the last fitness function. This function is calculated according to the coding result of data placement strategies for IoT datasets, as well as the number of hosts and switches employed. Algorithm 3 specifies the process of calculating the energy consumption. In this algorithm,

---

**Algorithm 1** Average resources utilization

---

**Require:**  $D, A, X, Y, H$ .**Ensure:** Average resource utilization

```
1:  $u = 0$ 
2:  $\lambda = 0$ 
3: for  $w = 1$  to  $W$  do
4:   for  $n = 1$  to  $N$  do
5:     if  $x_n == h_w$  then
6:       Calculate  $u_w(X)$  by Eq.(1)
7:        $\lambda = \lambda + 1$ 
8:        $u = u + u_w(X)$ 
9:     end if
10:   end for
11: end for
12: Calculate  $U(X)$  by Eq.(5)
13: return  $U(X)$ 
```

---

the total energy consumption includes the basic energy consumption of hosts, the energy consumption of the working VMs, and the energy consumption of switches. We first calculate the basic energy consumption of hosts according to the coding result of data placement strategies (Lines 1 to 8). Secondly, the energy consumption of the working VMs according to the resource requirements of IoT datasets is calculated, then calculate the energy consumption of switches (Lines 9 to 14). Finally, the total energy consumption is obtained (Line 15).

The constraints in this paper are considering the capacity of each host and privacy constraints between IoT datasets. The constraints are given in Eq.(20), Eq.(21) and Eq.(22). Different datasets with privacy conflicts are stored in the same physical host will be leaked with meaningful probability. Thus, the selected strategies must be required to satisfy the privacy constraints. IoT datasets should be placed separately if they are privacy conflicted.

Fig. 3 shows an example of privacy-aware data placement. As shown in

---

**Algorithm 2** Average access time

---

**Require:**  $D, A, X, Y, H$ .**Ensure:** Average access time  $T(X)$ 

```
1:  $t = 0$ 
2: for  $n = 1$  to  $N$  do
3:   for  $q = 1$  to  $Q$  do
4:     if  $x_n == y_q$  then
5:       Calculate  $t_{q,n}(X)$  by Eq.(8)
6:        $t = t + t_{q,n}(X)$ 
7:     end if
8:   end for
9: end for
10: Calculate  $T(X)$  by Eq.(9)
11: return  $T(X)$ 
```

---

Fig. 3, the IoT application  $a_1$  needs to access four datasets, i.e.,  $x_1, x_2, x_3$  and  $x_4$ . The IoT application  $a_1$  is placed on the host  $h_4$  and IoT datasets  $x_1, x_2$  and  $x_3$  are placed on the hosts  $h_1, h_2$  and  $h_3$ , respectively. Suppose  $x_4$  has privacy conflicts with  $x_1, x_2$  and  $x_3$ , and consequently  $x_4$  must be placed on the host  $h_5$  to meet the privacy constraints.

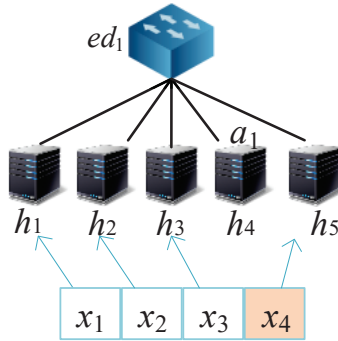


Figure 3: An example of privacy-aware data placement.

---

**Algorithm 3** Energy consumption

---

**Require:**  $D, A, X, Y, H$ .

**Ensure:** Energy consumption  $E(X)$

```
1: for  $w = 1$  to  $W$  do
2:   Calculate  $BE_{Host}$  by Eq.(10)
3:   for  $n = 1$  to  $N$  do
4:     if  $x_n == h_w$  then
5:       Calculate  $E_{VM}$  by Eq.(13)
6:     end if
7:   end for
8: end for
9: for  $n = 1$  to  $N$  do
10:  for  $q = 1$  to  $Q$  do
11:    Calculate  $\delta_{q,n}(X)$  by Eq.(14)
12:    Calculate  $E_{switch}$  by Eq.(15)
13:  end for
14: end for
15: Calculate  $E(X)$  by Eq.(16)
16: return  $E(X)$ 
```

---

### 3.3. Genetic Operation

NSGA-II is a multi-objective genetic algorithm without finding a solution that maximizes or minimizes each objective. Generally, there are multiple sets of solutions in multi-objective optimization problems. Each group of solutions we called non-dominated solution, these non-dominated solutions are inferred to as Pareto frontier. Every non-inferior solution is a chromosome composed of genes. According to the data placement model presented in the section 2, our goal is to find out the optimal solutions for data placement.

### 215 3.3.1. Initialization

In the initial stage of genetic operation, the parameters should be determined in advance, including the size of population  $Z_{pop}$ , the number of iterations  $Gen$ , the crossover probability  $\xi_c$  and the mutation probability  $\xi_m$ . In this paper, each chromosome consists of placement strategies for all of datasets, it is denoted by  $X_\alpha = \{x_1, x_2, \dots, x_N | 1 \leq \alpha \leq Z_{pop}\}$ , where  $X_\alpha$  represents the  $\alpha$ -th chromosome in the population and  $N$  represents the total number of datasets. Two populations  $S_j$  and  $O_j$  of size  $Z_{pop}$  are randomly generated and they are mixed together to form a population  $L_j$  with a population size of  $2Z_{pop}$ .

### 3.3.2. Selection

225 The population  $L_j$  with a population size of  $2Z_{pop}$  in initialization generates the multiple non-dominated layers ( $V_i, i = 1, 2, \dots$ ) using *fast non-dominated sorting* approach.

At the same time, all individuals in the non-dominated layer are calculated for the crowding-distance. The selection operation is to pick out some of the chromosomes from the population, carry out crossover and mutation operations, as well as generate a new population with better fitness. The selection method in NSGA-II is based on the *crowded-comparison* operator. the crowding-distance of data placement strategy is calculated by:

$$cd_\alpha = cd_\alpha^U + cd_\alpha^T + cd_\alpha^E = |U^{\alpha+1} - U^{\alpha-1}| + |T^{\alpha+1} - T^{\alpha-1}| + |E^{\alpha+1} - E^{\alpha-1}|, \quad (23)$$

where  $cd_\alpha$  represents the crowding-distance of  $\alpha$ -th data placement strategy  $S_\alpha$ ,  $cd_\alpha^U, cd_\alpha^T, cd_\alpha^E$  represents the objective functions, respectively.  $U^{\alpha+1}$  represents the value of  $(\alpha + 1)$ -th data placement strategy to the objective function  $U(X)$ .

According to the non-dominated rank and the crowding distance, we select the best individuals to form the parent population. The data placement strategies for IoT datasets selected meets the constraints of privacy preservation.

### 3.3.3. Crossover and Mutation

235 The crossover is to combine the two parental chromosomes in the population, trying to get better offspring chromosomes. Fig. 4 shows an example of crossover



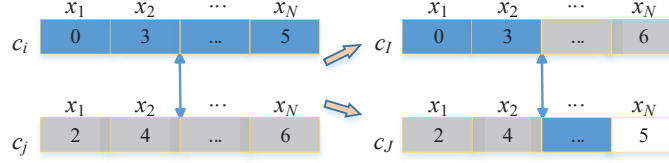


Figure 4: An example of crossover operation.

operation. As shown in Fig. 4, we firstly select a crossover point after  $x_2$  in the chromosome and then swap genes of two parental chromosomes on both sides of crossover point, then complete the crossover operation.

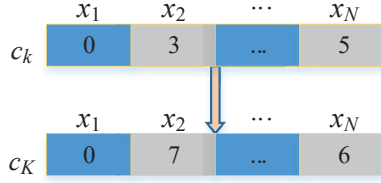


Figure 5: An example of mutation operation.

240 The mutation is to slightly modify some of the genes in a chromosome to produce a well-fitness individual as well as avoid early convergence. Fig. 5 shows an example of the standard mutation that each gene is changed with equal probability,  $x_2$  and  $x_N$  are mutated with the same mutation probability. After crossover and mutation operation, the data placement strategy generated  
245 still meets the constraints of privacy preservation.

#### 3.3.4. Method Overview of IDP

In this paper, the optimized goal is to improve the resource utilization, reduce energy consumption and optimize the data access time while taking the data privacy constraints into account. The data placement problem is defined  
250 as a multi-objective problem, and the NSGA-II is used to obtain the optimal data placement strategy. Firstly, the data placement strategies for datasets are encoded as the number of the hosts, and fitness functions are given for the

---

**Algorithm 4** IoT-oriented Data Placement method

---

**Require:**  $D, A, X, Y, H$ .

**Ensure:** Optimal datasets placement strategy  $X^*$

```
1:  $G = 1$ 
2: while  $G \leq Gen$  do
3:    $L_j = S_j + O_j$ 
4:    $V = \text{Fast Non-dominated Sort } (L_j)$ 
5:    $S_j = \emptyset$ 
6:    $i = 1$ 
7:   while  $\text{sum}(S_j) < Z_{pop}$  do
8:      $\text{crowded-comparison}(V_i)$  by Eq.(23)
9:      $S_j += V_i$ 
10:     $i = i + 1$ 
11:  end while
12:   $O_j = \text{crossover and mutation operation}(S_j)$ 
13:   $G = G + 1$ 
14: end while
15: return  $X^*$ 
```

---

data placement problems. Then, the *fast non-dominated sorting* approach is used to generate multiple non-dominated layers for individuals and pretreat the population to better distinguish the merits of individuals. Crowding distance computation is used to identify individuals with better fitness.

The overview of IDP we proposed is shown in Algorithm 4. The inputs of the Algorithm 4 are IoT datasets collection  $D$ . The algorithm starts from the first iteration (Line 1). Two populations  $P_j$  and  $O_j$  of size  $Z_{pop}$  are randomly generated and form a population  $L_j$  with a population size of  $2Z_{pop}$ . The initialized population  $L_j$  is divided into multiple non-dominated layers (Lines 3 and 4). The selection operation follows two rules. (1): The individual with higher level of non-dominated layers is prioritized; (2): The individual with better crowding distance is prioritized when individuals are in the same non-dominated level.

265 The excellent individuals are selected to fill in a new population of size  $Z_{pop}$  by  
crowding distance (Lines 7 to 10). Then the offspring population is generated  
after the traditional crossover and mutation (Line 12). The offspring popula-  
tion is merged with the parent population and iterated again until the algorithm  
stops (Lines 2 to 14). Finally, the optimal placement strategies are output (Line  
270 15).

#### 4. Experimental Evaluation

In this section, we conduct extensive simulations to evaluate the performance  
optimization of the data placement method. Specifically, we first briefly intro-  
duce our simulation setup including simulation dataset and comparative meth-  
ods. Then, we evaluate the influence of the dataset scale on the performance  
275 of resources utilization and the energy consumption. Finally, we illustrate the  
evaluation results on our proposed data placement method.

##### 4.1. Simulation Setup

Table 2: Parameter Settings

| Parameter   | Value |
|---|-------|
| The number of tasks $Q$   | 40    |
| The time of running period $R$  | 24h   |
| The number of hosts $W$   | 3456  |
| The baseline energy consumption rate of $w$ -th host $\alpha_w$           | 342W  |
| The active energy consumption rate for each VM $\eta$                     | 23W   |
| The number of pods involved in fat-tree topology $M$                      | 288   |
| The bandwidth between severs and edge switches $b_{SE}$                   | 50M   |
| The bandwidth between switches in fat-tree topology $b_{EA}$ and $b_{AC}$ | 100M  |
| The idle energy consumption rate for each switch $\beta$                  | 100W  |
| The active energy consumption rate for each switch $\gamma$               | 120W  |

In our simulation, the fat-tree topology is employed to create a CDC with  
280 288 pods, including 144 core switches, 288 aggregations switches and 288 edge  
switches. Six scales of datasets are applied for experimenting the proposed IDP  
method which are  $\{200, 400, 600, 800, 1000, 1200\}$ . The specified parameter  
settings in this experiment are illustrated in Table 2.

To conduct the comparison analysis, we propose some other basic data place-  
285 ment methods in addition to our IDP method. The data placement problem in  
this paper is similar to the packing problems. Traditional algorithms are mainly  
used to solve the packing problems, including the First Fit Decreasing (FFD)  
and Best Fit Decreasing (BFD) algorithms [19]. Hence, we will compare our  
method with them .

290 **Benchmark:** In this method, the IoT datasets are placed in the order of the  
hosts. When the former host is full, the datasets left are placed on the next  
host. When the datasets have privacy conflicts, they are not placed on the same  
hosts. This process is repeated until all datasets have been placed.

**First Fit Decreasing-Privacy Reservation (FFD-PR):** The IoT datasets  
295 are sorted in descending order according to the dataset requests first. Then the  
sorted datasets are placed on the hosts. If the left resources of current host are  
insufficient for the resource requirement of a dataset, the dataset is placed on  
the host with enough resource next to this host. Specially, when placing each  
dataset, the privacy-preservation are considered. This process is repeated until  
300 all datasets have been placed.

**Best Fit Decreasing- Privacy Reservation (BFD-PR):** The IoT datasets  
and the hosts are both sorted in descending order according to the dataset re-  
quest and the space of hosts first. Then the sorted datasets are placed on the  
sorted hosts. If the left resources of the host are insufficient for the resource re-  
305 quirement of the dataset, the dataset is placed on the host with enough resource  
next to this host in optimal principle. Specially, when placing each dataset, the  
privacy-preservation are considered. This process is repeated until all datasets  
have been placed.

The methods are implemented under the simulation tools by the CloudSim

310 on a PC machine with 2 Intel Core i5-6500U 3.20GHz processors and 8GB RAM.  
The corresponding evaluation results are detailedly depicted in the following  
sections.

#### 4.2. Performance evaluation on IDP

Our proposed method tends to achieve a trade-off between multiple objec-  
315 tives, which are improving the average resources utilization of hosts, optimizing  
the average data access time and reducing the energy consumption. We con-  
ducted 50 experiments in the case of convergence for each dataset scale and  
get multiple sets of results. In order to pick out a relatively best solution, we  
use Simple Additive Weighting (SAW) and Multiple Criteria Decision Making  
320 (MCDM) to construct the utility function, which is measured by [19]:

$$\begin{aligned} V'(c_i) = & \frac{1}{3} \cdot \frac{U(c_i) - U^{\min}(X)}{U^{\max}(X) - U^{\min}(X)} + \frac{1}{3} \cdot \frac{T^{\max}(X) - T(c_i)}{T^{\max}(X) - T^{\min}(X)} \\ & + \frac{1}{3} \cdot \frac{E^{\max}(X) - E(c_i)}{E^{\max}(X) - E^{\min}(X)}, \end{aligned} \quad (24)$$

where  $U(c_i)$ ,  $T(c_i)$  and  $E(c_i)$  represent the fitness of three objective functions  
with the data placement strategy  $c_i$ , respectively.  $U^{\max}(X)$  and  $U^{\min}(X)$  rep-  
resent the maximum and minimum fitness for the resource utilization,  $T^{\max}(X)$   
and  $T^{\min}(X)$  represent the maximum and minimum fitness for the access time,  
325  $E^{\max}(X)$  and  $E^{\min}(X)$  represent the maximum and minimum fitness for the  
energy consumption. If  $U^{\max}(X) = U^{\min}(X)$ , let  $\frac{U(c_i) - U^{\min}(X)}{U^{\max}(X) - U^{\min}(X)} = 1$ . Anal-  
ogously, if  $T^{\max}(X) = T^{\min}(X)$  and  $E^{\max}(X) = E^{\min}(X)$ , we adopt the  
same calculation manner.

It is intuitive from Fig. 6 that the multiple sets of solutions are generated  
330 in each dataset scale, the solutions are calculated with SAW and MCDM re-  
spectively and we obtain a relatively goal solution. Fig. 6a shows that goal  
solution is solution-1 when the dataset scale is 200. Fig. 6b illustrates that goal  
solution is solution-3 when the dataset scale is 400. Based on Fig. 6c, the goal  
solution is solution-2 when the dataset scale is 600. When the dataset scale  
335 is 800, based on Fig. 6d, the goal solution is solution-1. Fig. 6e shows that

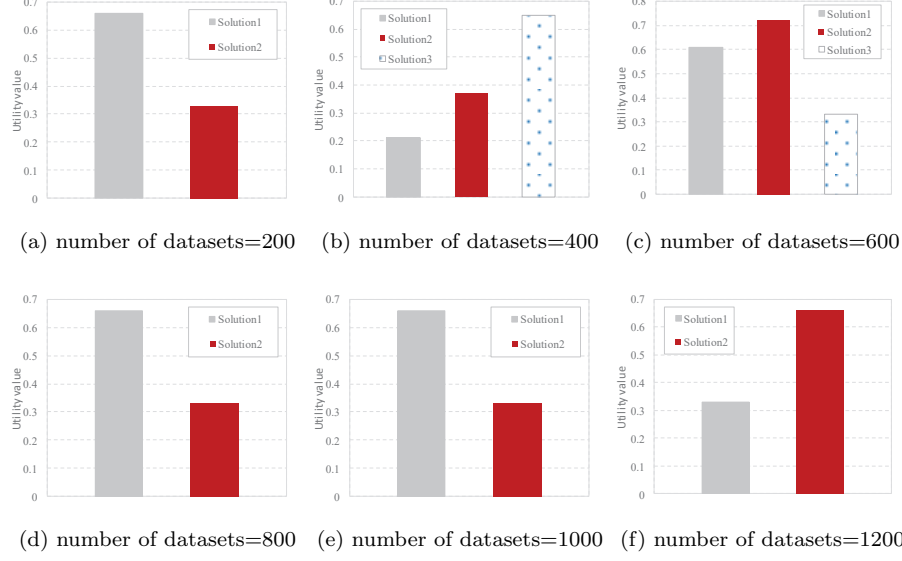


Figure 6: Comparison of utility value with different scales of datasets by the generated solutions of IDP.

goal solution is solution-1 when dataset scale is 1000. Fig. 6f shows that goal solution is solution-2 when dataset scale is 1200. The experiments prove that utility function can obtain a relatively balanced solution for arbitrary scale of IoT datasets.

#### 4.3. Comparison analysis

In this subsection, we focus on evaluating the performance of our method and making comparison with multiple methods under the same configuration. The resource utilization and the energy consumption are two main metrics of the resource usage in CDCs. Furthermore, the number of employed hosts is closely related to the resource utilization and the average data access time is relevant to the energy consumption. The corresponding results are shown in Fig. 7, Fig. 8, Fig. 9 and Fig. 10.

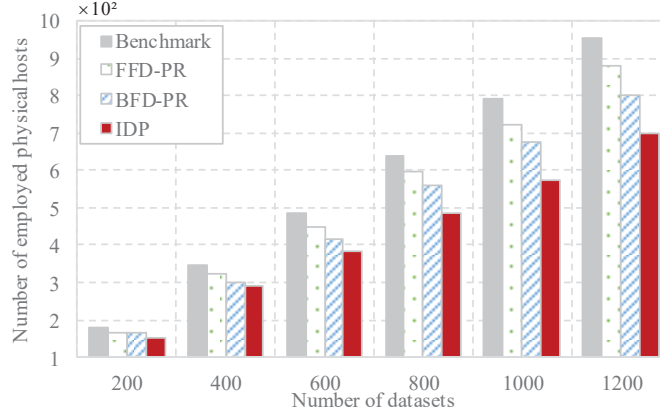


Figure 7: Comparison of the number of employed physical hosts with different dataset scale by Benchmark, FFD-PR, BFD-PR and IDP.

#### 4.3.1. Comparison of the number of employed hosts

In Fig. 7, we first compare the number of employed hosts of four data placement methods. Based on Fig. 7, it is indicated that IDP employs fewer physical hosts than the other basic method. With the increase of dataset scale, the number of employed hosts of IDP is increasing, but the increasing is in a more gentler way than other methods.

#### 4.3.2. Comparison of resource utilization

Then we compare the average resource utilization of four data placement methods in Fig. 8. We find that with the increase of dataset scale, our proposed method IDP can achieve better average resource utilization and maintain the utility at a high level. This confirms our intuition: The increase of datasets means that one host can actually be placed with more datasets (less space wasted). Therefore, the resource utilization can be improved.

#### 4.3.3. Comparison of average data access time

Then the average data access time is compared in Fig. 9. From Fig. 9, IDP cost fewer average data access time than the comparison method. However, with the increase of dataset scale, the average data access time of IDP is increasing.

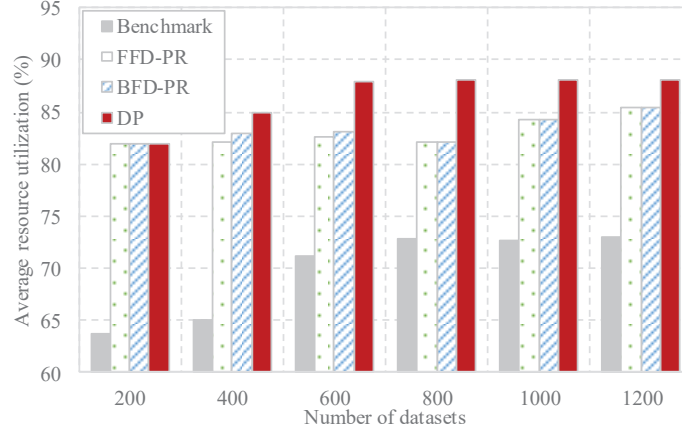


Figure 8: Comparison of average resource utilization with different dataset scale by Benchmark, FFD-PR, BFD-PR and IDP.

365 This may be because the hosts seem to be more fully utilized with the increase of datasets, which means that in reality we do not need to employ too datasets in a limited host.

#### 4.3.4. Comparison of energy consumption

From section 2, the energy consumption is composed with the VM energy consumption, the host energy consumption and the switche energy consumption. In Fig. 10, we first compare this three-energy consumption with different dataset scale. Based on Fig. 10a, we find that with the increase of dataset scale, all these methods have the increase of baseline energy consumption of VMs but in the approach level. Fig. 10b illustrates IDP has less energy consumption of active hosts. This may be because IDP employed fewer physical hosts. Fig. 10c indicates IDP has less energy consumption of switches for the reason of less average data access time IDP taking.

Then the comparison of total energy consumption is conducted in Fig. 11, which is intuitive that IDP obtains a better performance on the energy consumption. With the increasing of dataset scale, the difference between IDP and the other methods is enlarging. This is because the energy consumption is



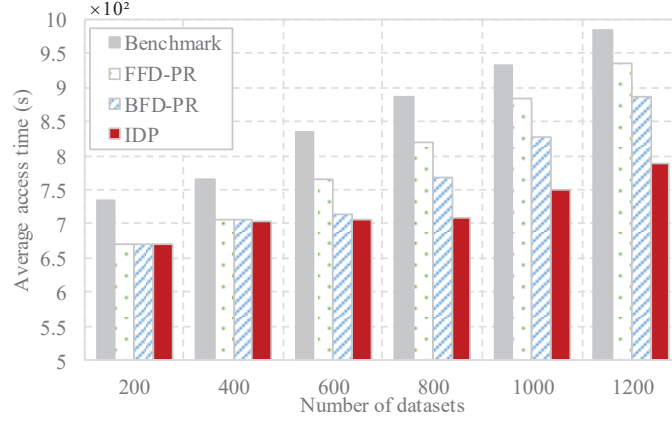
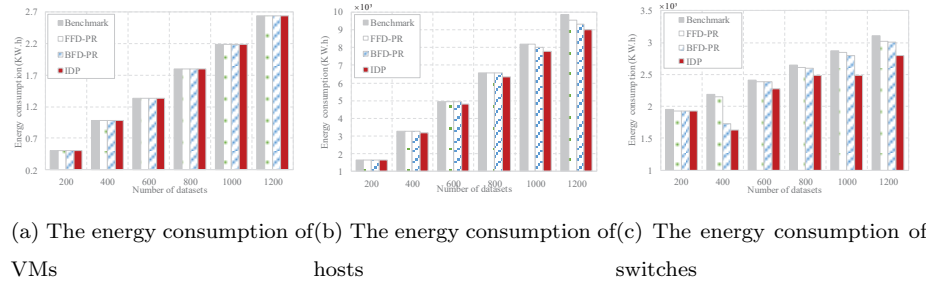


Figure 9: Comparison of average data access time with different dataset scale by Benchmark, FFD-PR, BFD-PR and IDP.



(a) The energy consumption of VMs (b) The energy consumption of hosts (c) The energy consumption of switches

Figure 10: Comparison on specific energy consumption with different dataset scale by Benchmark, FFD-PR, BFD-PR and IDP.

affected by the amount of hosts and switches.

## 5. Related Work

In recent years, with the development of IoT, large volume of data streams from billions of interconnected heterogeneous devices are emitted for the requirements of processing in real time [20] [21] [22] [23]. Cloud computing which has comparative advantages for data processing is of great benefit to process the data from IoT applications [24] [25] [26] [27].

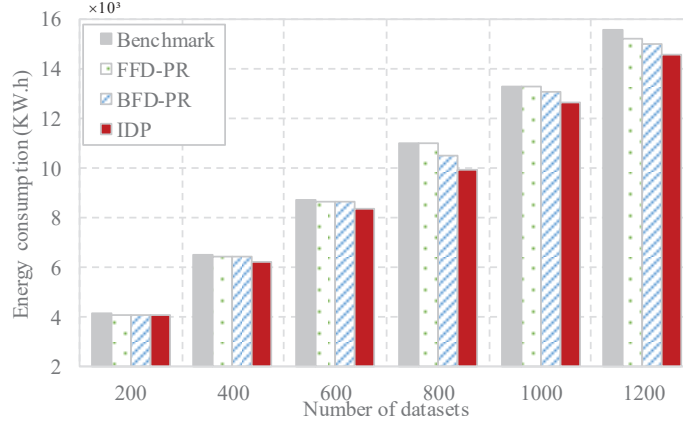


Figure 11: Comparison of total energy consumption with different dataset scale by Benchmark, FFD-PR, BFD-PR and IDP.

Due to the direct impact of data placement in cloud environment on data access and data storage, there are multiple methods to improve the utilization efficiency of cloud resources [28] [29] [30] [31] [32]. Hajisami et al. [28] proposed a reconfigurable solution based on C-RAN, which can be dynamically and effectively adapt to the fluctuation of user demand to reduce the waste of resources and improve the efficiency of resource utilization. Guo et al. [29] simulated the resource allocation problem as a Vector Bin Packing Problem (VBPP) and designed a resource allocation strategy that minimizes the number of servers used to maximize resources utilization rate. Fan et al. [30] proposed a VM-container hybrid hierarchical resource scheduling mechanism to alleviate resource utilization efficiency issues, dividing tasks into different levels and formulating initial scheduling strategies for different levels of tasks. Agarwal et al. [31] presented a system named Volley which can analyze the logs of datacenter requests and output migration recommendations to address data placement problem. Yu et al. [32] proposed the sketch-based data placement (SDP) to lower the overhead and keep the benefits of the data placement.

Energy consumption which is generated by the management of resources in the cloud datacenter is a key factor to consider for data placement. Hajisami et

al. [33] proposed an elastic resource deployment framework called Elastic-Net to solve the user demand fluctuations while minimizing energy consumption in the cloud. Aksanli et al. [34] introduced an online job migration algorithm between data centers to reduce overall energy consumption based on assessments of the advantages of short-term green energy forecasting for data center scale. Access time is another key factor. Jiao et al. [35] proposed an optimization approach by leveraging graph cuts to optimize multi-objective data placement across clouds for socially aware services.

More and more attentions have been paid on data security [36] [37]. Several works have been investigated for the privacy preservation in the cloud environment [38] [39]. Some were discussed about separating private data from public data and placing them in trusted private cloud and untrusted public cloud respectively [40] [41] [42] [43] [44] [45]. Zhou et al. [41] presented a set of techniques for privacy-aware data retrieval by splitting data and storing on hybrid cloud. Huang et al. [42] proposed a scheme to achieve image data privacy over hybrid cloud efficiently and proposed a one-to-one mapping function for image encryption. Wang et al. [43] described several methods about protecting data security in hybrid cloud and discussed an authentication intercloud model. Abrishami et al. [44] presented a scheduling algorithm to protect data privacy while minimizing the cost and satisfy the users limitation. Stout et al. [45] surveyed the shortcomings and challenges of securing IoT devices and their interactions with cloud and enterprise applications.

Currently, an energy efficient network architecture for IoT applications has been proposed [18]. The method is robust, however, to the best of our knowledge, the existing work has not taken the privacy in data placement into consideration. With the observations above, it is still a challenge to realize IoT data placement to optimize the resource utilization, energy consumption and accessing time, while taking the data privacy into account. In view of this challenge, an IoT data placement method with privacy preservation in cloud environment is proposed this paper.

## 6. Conclusion and Future Work

The existing works of IoT data placement in cloud environment have not taken privacy into account. In this paper, an IoT-oriented data placement  
440 method with privacy preservation in cloud environment is proposed. We construct a systematic model of resource utilization, energy consumption and access time of CDCs when deploying the IoT data of privacy conflict in the cloud environment. The proposed method is designed to improve the average resource utilization of CDCs and access performance while reduce the energy consumption  
445 of CDCs based on the Non-dominated Sorting Genetic Algorithm II (NSGA-II). Through adequate experimental evaluation and comparison analysis, the efficiency and effectiveness of our proposed method are validated.

For future work, we will adjust and extent our method to implement the IoT data placement in exact physical environment. accordingly, the privacy  
450 preservation strategy will be improved on the privacy constraints of the real-time IoT data.

## 7. Acknowledgement

This research is supported by the National Science Foundation of China under grant no. 61702277, no. 61872219 and no. 61772283. Besides, this work is  
455 also supported by The Startup Foundation for Introducing Talent of NUIST, the open project from State Key Laboratory for Novel Software Technology, Nanjing University under grant no. KFKT2017B04, the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment  
460 Technology (CICAEET), and the project “Six Talent Peaks Project in Jiangsu Province” under grant no. XYDXXJS-040.

## References

- [1] Q. Fan, N. Ansari, Application aware workload allocation for edge computing-based iot, *IEEE Internet of Things Journal* 5 (3) (2018) 2146–2153.
- [2] H. Yan, X. Li, Y. Wang, C. Jia, Centralized duplicate removal video storage system with privacy preservation in iot., *Sensors* 18 (6) (2018) 1814.
- [3] R. H. Jhaveri, N. M. Patel, Y. Zhong, A. K. Sangaiah, Sensitivity analysis of an attack-pattern discovery based trusted routing scheme for mobile ad-hoc networks in industrial iot, *IEEE Access* (2018) 1–1.
- [4] T. Wang, M. Z. A. Bhuiyan, G. Wang, M. A. Rahman, J. Wu, J. Cao, Big data reduction for a smart citys critical infrastructural health monitoring, *IEEE Communications Magazine* 56 (3) (2018) 128–133.
- [5] T. Wang, J. Zeng, Y. Lai, Y. Cai, H. Tian, Y. Chen, B. Wang, Data collection from wsns to the cloud based on mobile fog elements, *Future Generation Computer Systems*.
- [6] Z. Wu, L. Tian, P. Li, T. Wu, M. Jiang, C. Wu, Generating stable biometric keys for flexible cloud computing authentication using finger vein, *Information Sciences*.
- [7] M. Su, L. Zhang, Y. Wu, K. Chen, K. Li, Systematic data placement optimization in multi-cloud storage for complex requirements, *IEEE Transactions on Computers* 65 (6) (2016) 1964–1977.
- [8] L. Yang, Z. Han, Z. Huang, J. Ma, A remotely keyed file encryption scheme under mobile cloud computing, *Journal of Network & Computer Applications* 106.
- [9] S. Mustafa, K. Bilal, S. U. R. Malik, S. A. Madani, Sla-aware energy efficient resource management for cloud environments, *IEEE Access* 6 (2018) 15004–15020.

- 490 [10] K. Wang, H. Yin, W. Quan, G. Min, Enabling collaborative edge computing for software defined vehicular networks, *IEEE Network PP* (99) (2018) 1–6.
- [11] X. Chen, J. Li, X. Huang, J. Ma, W. Lou, New publicly verifiable databases with efficient updates, *IEEE Transactions on Dependable Secure Computing* 12 (5) (2015) 546–556.
- 495 [12] T. Wang, Y. Li, W. Fang, W. Xu, J. Liang, Y. Chen, X. Liu, A comprehensive trustworthy data collection approach in sensor-cloud system, *IEEE Transactions on Big Data PP* (99) (2018) 1–1.
- [13] J. Cheng, R. Xu, X. Tang, V. S. Sheng, C. Cai, An abnormal network flow feature sequence prediction approach for ddos attacks detection in big data environment, *CMC: Computers, Materials & Continua* 55 (1) (2018) 095–119.
- 500 [14] L. Xiong, Y. Shi, On the privacy-preserving outsourcing scheme of reversible data hiding over encrypted image data in cloud computing, *CMC: Computers, Materials & Continua* 55 (3) (2018) 523–539.
- [15] X. Xie, T. Yuan, X. Zhou, X. Cheng, Research on trust model in container-based cloud service, *CMC: Computers, Materials & Continua* 56 (2) (2018) 273–283.
- 505 [16] K. He, J. Chen, R. Du, Q. Wu, G. Xue, X. Zhang, Deypos: Deduplicatable dynamic proof of storage for multi-user environments, *IEEE Transactions on Computers* 65 (12) (2016) 3631–3645.
- 510 [17] C. Esposito, A. De Santis, G. Tortora, H. Chang, K.-K. R. Choo, Blockchain: A panacea for healthcare cloud-based data security and privacy?, *IEEE Cloud Computing* 5 (1) (2018) 31–37.
- [18] P. Sarwesh, N. S. V. Shet, K. Chandrasekaran, Energy-efficient network architecture for iot applications, in: *Beyond the Internet of Things*, Springer, 2017, pp. 119–144.
- 515

- [19] X. Xu, W. Dou, X. Zhang, J. Chen, Enreal: An energy-aware resource allocation method for scientific workflow executions in cloud environment, *IEEE Transactions on Cloud Computing* 4 (2) (2016) 166–179.
- [20] C. Wu, E. Zapevalova, Y. Chen, F. Li, Time optimization of multiple knowledge transfers in the big data environment, *CMC: Computers, Materials & Continua* 54 (3) (2018) 269–285.
- [21] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, *Computer networks* 54 (15) (2010) 2787–2805.
- [22] T. Qiu, A. Zhao, F. Xia, W. Si, D. O. Wu, Rose: Robustness strategy for scale-free wireless sensor networks, *IEEE/ACM Transactions on Networking* PP (99) (2017) 1–16.
- [23] T. Qiu, R. Qiao, D. O. Wu, Eabs: An event-aware backpressure scheduling scheme for emergency internet of things, *IEEE Transactions on Mobile Computing* PP (99) (2018) 1–1.
- [24] F. Tao, Y. Cheng, L. Da Xu, L. Zhang, B. H. Li, Cciot-cmfg: cloud computing and internet of things-based cloud manufacturing service system, *IEEE Transactions on Industrial Informatics* 10 (2) (2014) 1435–1442.
- [25] M. Aazam, E.-N. Huh, Fog computing: The cloud-iot\ioe middleware paradigm, *IEEE Potentials* 35 (3) (2016) 40–44.
- [26] V. C. Emeakaro, N. Cafferkey, P. Healy, J. P. Morrison, A cloud-based iot data gathering and processing platform, in: *Future Internet of Things and Cloud (FiCloud)*, 2015 3rd International Conference on, IEEE, 2015, pp. 50–57.
- [27] A. Botta, W. De Donato, V. Persico, A. Pescapé, Integration of cloud computing and internet of things: a survey, *Future Generation Computer Systems* 56 (2016) 684–700.

- [28] A. Hajisami, T. X. Tran, D. Pompili, Dynamic provisioning for high energy efficiency and resource utilization in cloud rans, in: Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on, IEEE, 2015, pp. 471–472.
- [29] L. Guo, P. Du, A. Razaque, M. Almiani, A. Al Rahayfeh, Energy saving and maximize utilization cloud resources allocation via online multi-dimensional vector bin packing, in: Software Defined Systems (SDS), 2018 Fifth International Conference on, IEEE, 2018, pp. 160–165.
- [30] C. Fan, Y. Wang, Z. Wen, Research on improved 2d-bpso-based vm-container hybrid hierarchical cloud resource scheduling mechanism, in: Computer and Information Technology (CIT), 2016 IEEE International Conference on, IEEE, 2016, pp. 754–759.
- [31] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, H. Bhogan, Volley: Automated data placement for geo-distributed cloud services.
- [32] B. Yu, J. Pan, Sketch-based data placement among geo-distributed datacenters for cloud storages, in: Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on, IEEE, 2016, pp. 1–9.
- [33] A. Hajisami, T. X. Tran, D. Pompili, Elastic-net: Boosting energy efficiency and resource utilization in 5g c-rans, arXiv preprint arXiv:1710.00731.
- [34] B. Aksanli, J. Venkatesh, I. Monga, T. S. Rosing, Renewable energy prediction for improved utilization and efficiency in datacenters and backbone networks, in: Computational Sustainability, Springer, 2016, pp. 47–74.
- [35] L. Jiao, J. Lit, W. Du, X. Fu, Multi-objective data placement for multi-cloud socially aware services, in: INFOCOM, 2014 Proceedings IEEE, IEEE, 2014, pp. 28–36.



- [36] J. Cui, Y. Zhang, Z. Cai, A. Liu, Y. Li, Securing display path for security-sensitive applications on mobile devices, *CMC: Computers, Materials & Continua* 55 (1) (2018) 017–035.
- [37] W. Quan, Y. Liu, H. Zhang, S. Yu, Enhancing crowd collaborations for software defined vehicular networks, *IEEE Communications Magazine* 55 (8) (2017) 80–86.
- [38] X. Wu, C. Zhang, R. Zhang, Y. Wang, J. Cui, A distributed intrusion detection model via nondestructive partitioning and balanced allocation for big data, *CMC: Computers, Materials & Continua* 56 (1) (2018) 61–72.
- [39] T. Wang, J. Zhou, M. Huang, M. Bhuiyan, A. Liu, W. Xu, M. Xie, Fog-based storage technology to fight with cyber threat, *Future Generation Computer Systems* 83.
- [40] J. Shen, C. Wang, T. Li, X. Chen, X. Huang, Z. H. Zhan, Secure data uploading scheme for a smart home system, *Information Sciences* 453.
- [41] Z. Zhou, H. Zhang, X. Du, P. Li, X. Yu, Prometheus: Privacy-aware data retrieval on hybrid cloud, in: *INFOCOM, 2013 Proceedings IEEE, IEEE, 2013*, pp. 2643–2651.
- [42] X. Huang, X. Du, Achieving big data privacy via hybrid cloud, in: *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on, IEEE, 2014*, pp. 512–517.
- [43] J. K. Wang, X. Jia, Data security and authentication in hybrid cloud computing model, in: *Global High Tech Congress on Electronics (GHTCE), 2012 IEEE, IEEE, 2012*, pp. 117–120.
- [44] H. Abrishami, A. Rezaeian, G. K. Tousi, M. Naghibzadeh, Scheduling in hybrid cloud to maintain data privacy, in: *Innovative Computing Technology (INTECH), 2015 Fifth International Conference on, IEEE, 2015*, pp. 83–88.

- [45] W. M. Stout, V. E. Urias, Challenges to securing the internet of things, in: Security Technology (ICCST), 2016 IEEE International Carnahan Conference on, IEEE, 2016, pp. 1–8.