Coordinated Placement of Meteorological Workflows and Data with Privacy Conflict Protection

Tao Huang[1], Shengjun Xue[1,2(✉)], Yumei Hu[3], Qing Yang[1], Yachong Tian[1], and Dan Zeng[4]

[1] School of Computer Science and Technology, Silicon Lake College, Suzhou, China
nuisthuangtao@163.com, sjxue@163.com, whuyq@163.com, 779273334@qq.com
[2] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China
[3] Shanghai Huanan Environmental Management Limited Company, Shanghai, China
shymhu@163.com
[4] Library of Wuhan University of Technology, Wuhan University of Technology, Hubei, China
zengd@whut.edu.cn

Abstract. Cloud computing is cited by various industries for its powerful computing power to solve complex calculations in the industry. The massive data of meteorological department has typical big data characteristics. Therefore, cloud computing has been gradually applied to deal with a large number of meteorological -services. Cloud computing increases the computational speed of meteorological services, but data transmission between nodes also generates additional data transmission time. At the same time, based on cloud computing technology, a large number of computing tasks are cooperatively processed by multiple nodes, so improving the resource utilization of each node is also an important evaluation indicator. In addition, with the increase of data confidentiality, there are some data conflicts between some data, so the conflicting data should be avoided being placed on the same node. To cope with this challenge, the meteorological application is modeled and a collaborative placement method for tasks and data based on Differential Evolution algorithm (CPDE) is proposed. The Non-dominated Sorting Differential Evolution (NSDE) algorithm is used to jointly optimize the average data access time, the average resource utilization of nodes and the data conflict degree. Finally, a large number of experimental evaluations and comparative analyses verify the efficiency of our proposed CPDE method.

Keywords: Meteorological · Coordinated placement · NSDE · Data access time · Resource utilization · Data conflict

X. Zhang et al. (Eds.): CloudComp 2019/SmartGift 2019, LNICST 322, pp. 553–568, 2020.
https://doi.org/10.1007/978-3-030-48513-9_44

1 Introduction

1.1 Background

With the advancement of meteorological data acquisition technology and the improvement of meteorological service requirements [1–3], the number and types of meteorological data continue to grow, and it has gradually become a typical industry big data [4,5]. At the same time, the computational complexity of meteorological applications is increasing [5], so the meteorological department offloads a large number of meteorological applications and data to cluster for execution and storage [6,7]. However, in order to improve the average response time of all meteorological applications, meteorological department analyzes the characteristics of massive meteorological data and rationally distributes meteorological big data to each storage node [8,9]. In addition, based on the overall placement of meteorological big data, meteorological department continues to study how to properly place all tasks and data to each node in cluster [8], thereby reducing the average data access time for all tasks in the application [10,11].

However, as the number of meteorological applications and data offloaded to cluster increases rapidly [12], the resource utilization of nodes in cluster is also being paid more and more attention [13], and it has become an important indicator to measure the performance of placement method [14,15]. In addition, with the improvement of the confidentiality of meteorological data, the placement of conflicting data has also received more and more attention. While improving the resource utilization of nodes, it is also necessary to avoid placing those conflicting data in the same storage node to ensure the security of meteorological data [16–18]. Therefore, the collaborative placement of tasks and data for each meteorological application has become a challenge. In response to this challenge, this paper proposes an optimization method for collaborative placement of tasks and data in the meteorological applications.

1.2 Paper Contributions

In this paper, the main contributions are as follows:

- We model the meteorological application in the meteorological fat-tree network as a workflow, and all operations in the meteorological application are modeled as a series of tasks in workflow.
- The coordinated placement problem of meteorological tasks and data is modeled as a multi-objective optimization problem.
- We propose a optimization method for the coordinated placement of meteorological tasks and data based on NSDE algorithm to optimize the object functions of model.

2 Analysis of Meteorological Scenarios

2.1 Meteorological Fat-Tree Network

Meteorological networks usually use the tree structure, but the bandwidth is layer-by-layer convergence in the traditional network, and the network conges-

tion is likely to occur. Therefore, based on the traditional tree network structure, the Fat-tree topology network structure has been proposed and has been widely adopted by the meteorological department. The Fat-tree network structure is divided into three layers from top to bottom: core layer, aggregation layer and edge layer, the aggregation layer switches and the edge layer switches form a pod. The bandwidth of Fat-tree topology network is not convergent, and there are multiple parallel paths between any two nodes, so that it can provide high throughput transmission service and high fault tolerance for the meteorological data center.

In actual meteorological applications, a public meteorological cloud data center is constructed based on the virtualization technology and the Fat-tree network topology. The switches in each department constitute a Pod, or all the switches of several adjacent departments constitute the same Pod, and each Pod connects to the servers of the department to which it belongs. According to the rules of Fat-tree, if the meteorological cloud data center contains N^{pod} Pods, the number of edge switches and aggregation switches in each pod is $N^{pod}/2$, the number of servers that can be connected in each pod is $(N^{pod}/2)^2$, and the number of core switches is also $(N^{pod}/2)^2$. Figure 1 shows a meteorological Fat-tree network topology with four Pods. In practical applications, the network size of meteorological department is usually much larger than this.

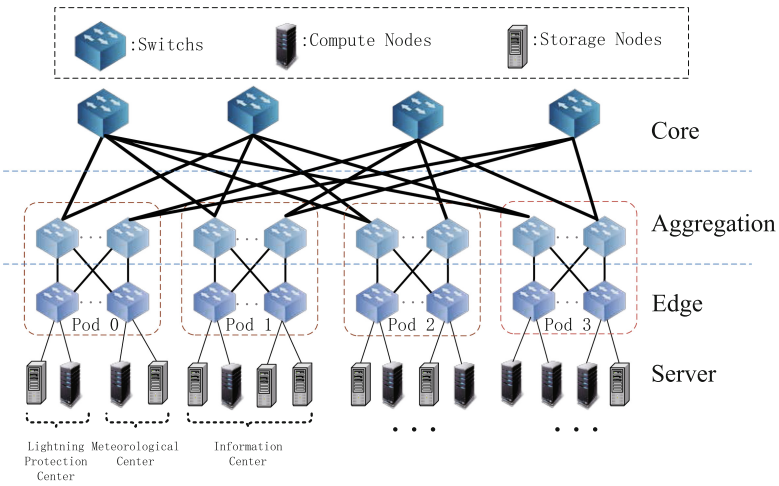

Fig. 1. A meteorological Fat-tree network with 4 pods.

2.2 Meteorological Scene and Workflow Description

In the current meteorological big data cloud processing mode, in order to improve the service efficiency of massive meteorological historical data and reduce the

average data access time. The meteorological department analyzes the characteristics of historical data that have been the most important data source for various meteorological applications, and reasonably stores the historical data in certain fixed storage nodes. In addition, user input data that can be dynamically placed during application execution is also an important data source for each application. Therefore, based on the placed meteorological historical data, the coordinated placement of tasks and input data in the meteorological applications is completed, so that the average data access time and the data conflict degree are minimized, the average resource utilization of all used nodes is maximized.

Based on workflow technology, each meteorological application can be modeled as a meteorological workflow, and operations in meteorological application can be modeled as a set of tasks in workflow. Figure 2 shows the workflow of weather forecast production.

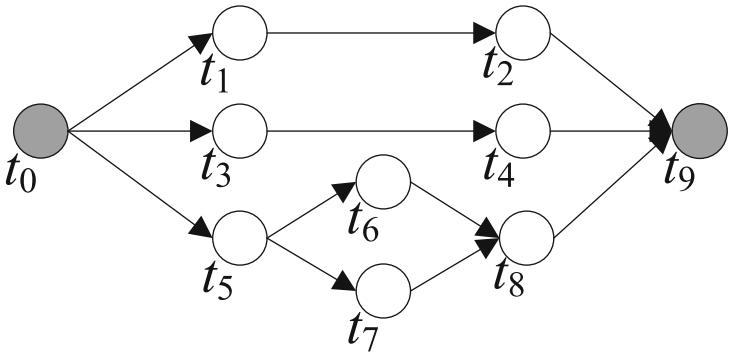

Fig. 2. The workflow of weather forecast production.

As the starting task, task t_0 represents *Data Collection* operation, including: automatic station data, radar data, satellite nephogram and so on. Task t_1 and task t_2 represent *Historical Weather Summary* operation and *Historical Weather Analysis* operation, that summarize the historical weather phenomena and analyze the causes of historical weather formation in the past 48 h, respectively. Task t_3 and task t_4 represent *Real-time Weather Summary* operation and *Real-time Weather Analysis* operation, that summarize the current weather phenomena and analyze the causes of current weather formation, respectively. Task t_5 represents *Forecast Mode Calculation* operation, the future weather is calculated in real time based on *European Centre for Medium-Range Weather Forecasts (ECMWF)* and *Global Forecasting System (GFS)*. Task t_6 and task t_7 represent *Weather Situation Analysis* operation and *Meteorological Elements Analysis* operation, that analyze the future weather situation and the future meteorological elements based on the calculation results of forecast model, respectively. Task t_8 represents *Generation of Forecast Model Conclusions* operation, based on the analysis for weather situation and meteorological elements, the final conclusion of forecast model is formed. As the termination task, task t_9 represents

Generation of Weather Forecast Conclusion operation, based on the analysis of historical weather and real-time weather, combined with the conclusion of forecast model, the final weather forecast conclusion is formed.

3 Problem Modeling and Formulation

3.1 Problem Modeling

In this section, we mainly model the coordinated placement problem of tasks and data, and formulate this model.

Assume that a meteorological workflow consists of M tasks, which can be defined as $TS = \{t_0,\ t_1,\ t_2,\ ...,\ t_{M-1}\}$. The data source of meteorological workflow mainly includes P input data and Q historical data, so the input data set and the historical data set can be defined as $D^{inp} = \{d_0^{inp},\ d_1^{inp},\ d_2^{inp},\ ...,\ d_{P-1}^{inp}\}$ and $D^{his} = \{d_0^{his},\ d_1^{his},\ d_2^{his},\ ...,\ d_{Q-1}^{his}\}$. Therefore, the relationship between tasks and data can be expressed as $\gamma = \{\gamma_0,\ \gamma_1,\ \gamma_2,\ ...,\ \gamma_{M-1}\}$, where $\gamma_m = \{d_a^{inp},\ ...,\ d_b^{inp}, d_c^{his},\ ...,\ d_e^{his}\}$ represents the data set required for the m-th task t_m. If there are K pairs of conflicting data, the conflicting relationship between these conflicting data can be expressed as $\beta = \{\beta_0,\ \beta_1,\ \beta_2,\ ...,\ \beta_k,\ ...,\ \beta_{K-1}\}$, where $\beta_k = \{d_x, d_y\}, (d_x, d_y \in \{D^{inp}, D^{his}\})$ represents the k-th pair of conflicting data.

3.2 Data Access Time Model

In the meteorological Fat-tree network, it is assumed that the task t_m and its required data are stored in the compute node u_i and the storage node u_j, respectively, and the data amount is d_n. The positional relationship between u_i and u_j can be defined as $\delta_{i,j}$, then:

- If u_i and u_j are the same node, then $\delta_{i,j} = 0$;
- If u_i and u_j belong to the same switch, then $\delta_{i,j} = 1$;
- If u_i and u_j belong to the different switches of the same pod, then $\delta_{i,j} = 2$;
- If u_i and u_j belong to the different pods, then $\delta_{i,j} = 3$;

Therefore, according to the positional relationship $\delta_{i,j}$ between u_i and u_j, the access time T_m^{AC} of the task t_m for the data can be expressed as:

$$T_m^{AC} = \begin{cases} 0\,; & \delta_{i,j} = 0 \\ 2 * d_n / B_{se}\,; & \delta_{i,j} = 1 \\ 2 * (d_n / B_{se} + d_n / B_{ea})\,; & \delta_{i,j} = 2 \\ 2 * (d_n / B_{se} + d_n / B_{ea} + d_n / B_{ac})\,; & \delta_{i,j} = 3 \end{cases} \tag{1}$$

where B_{se}, B_{ea}, and B_{ac} represent the bandwidth between the server and the edge layer switch, the bandwidth between the edge layer switch and the aggregation layer switch, and the bandwidth between the aggregation layer switch and the core switch, respectively.

Therefore, the data access time of task t_m for its required data set γ_m can be calculated as:

$$T_m^{Total} = \sum_{d_n \in \gamma_m} {T_m}^{AC} \tag{2}$$

Then, the average data access time for M tasks can be calculated as:

$$T_{avg}^{AC} = \sum_{m=0}^{M-1} T_m^{Total}/M \tag{3}$$

3.3 Resource Utilization Model

Assume that the number of compute nodes and storage nodes are N^{col} and N^{sto}, respectively. And the resource of each compute node and storage node are VM_{MAX}^{col} and VM_{MAX}^{sto}, respectively. The amount of resources required for each task and each data are expressed as $TVM = \{tvm_0^{col}, tvm_1^{col}, ..., tvm_m^{col}, ..., tvm_{M-1}^{col}\}$ and $DVM = \{dvm_0^{sto}, dvm_1^{sto}, ..., dvm_n^{sto}, ..., dvm_{P+Q-1}^{sto}\}$, respectively. tvm_m^{col} and dvm_m^{col} represent the amount of resources required for the m-th task and the n-th data, respectively.

Therefore, the placement of M tasks on the compute nodes can be represented as the two-dimensional array $CT[N^{col}, M]$, and the placement of $P+Q$ data on the storage nodes can be represented as the two-dimensional array $SD[N^{sto}, P+Q]$, then:

$$CT[i,m] = \begin{cases} 1; t_m \ is \ placed \ on \ the \ i-th \ compute \ node \\ 0; Otherwise \end{cases} \tag{4}$$

$$SD[j,n] = \begin{cases} 1; d_n \ is \ placed \ on \ the \ j-th \ storage \ node \\ 0; Otherwise \end{cases} \tag{5}$$

Then, the resource utilization of the i-th compute node and the j-th storage node can be expressed as U_i^{col} and U_j^{sto}, respectively.

$$U_i^{col} = \sum_{m=0}^{M-1} tvm_m^{col} * CT[i,m]/VM_{MAX}^{col} \tag{6}$$

$$U_j^{sto} = \sum_{n=0}^{P+Q-1} dvm_n^{sto} * SD[j,n]/VM_{MAX}^{sto} \tag{7}$$

If the number of compute nodes and storage nodes that have been used is ${N_{use}}^{col}$ and ${N_{use}}^{sto}$, respectively. The average resource utilization of the currently used compute nodes and storage nodes can be expressed as $\overline{U^{col}}$ and $\overline{U^{sto}}$, respectively.

$$\overline{U^{col}} = \sum_{i=0}^{N^{col}} U_i^{col}/N_{use}^{col} \tag{8}$$

$$\overline{U^{sto}} = \sum_{j=0}^{N^{sto}} U_j^{sto} / N_{use}^{sto} \tag{9}$$

Finally, the average resource utilization of compute nodes and storage nodes is calculated as:

$$\overline{U} = \frac{N^{col}}{N^{col} + N^{sto}} * \overline{U^{col}} + \frac{N^{sto}}{N^{col} + N^{sto}} * \overline{U^{sto}} \tag{10}$$

3.4 Data Conflict Model

Because the closer the conflicting data is placed on network, the greater the possibility of privacy breaches. Therefore, in order to ensure data privacy, conflicting data should be prevented from being placed on the same node or the same pod.

Assume that there are N^{SN} pairs of conflicting data placed on the same node. There are N^{SS} pairs of conflicting data placed on the different nodes of the same edge layer switch. There are N^{SP} pairs of conflicting data placed under different edge layer switches of the same pod. We set the corresponding weights w_0, w_1 and w_2 for these three placement of conflicting data.

Then, the data conflict degree for all conflicting data can be expressed as:

$$C = w_0 * N^{SN} + w_1 * N^{SS} + w_2 * N^{SP} \tag{11}$$

where $w_0 + w_1 + w_2 = 1$, the closer the conflicting data are placed, the larger the corresponding weight. Therefore, in this experiment, the three weights are set to 0.55, 0.3, and 0.15, respectively.

3.5 Objective Functions

In this paper, the coordinated placement of meteorological workflow and data with privacy conflict protection has been modeled as a multi-objective optimization problem. Average data access time, average resource utilization, and data conflict degree are used as the three objective functions of this optimization problem. Therefore, this optimization model can be expressed as:

$$Min(T_{avg}^{AC}, \overline{U}), Max(C) \tag{12}$$

In addition, this optimization problem also needs to meet certain constraints, that is, the used resources of each node cannot exceed the maximum resource amount of node, so the constraint can be expressed as:

$$s.t. \forall U_i^{col} \leq 1, \forall U_j^{sto} \leq 1 | 0 \leq i < N^{col}, 0 \leq j < N^{sto} \tag{13}$$

In addition, he symbols used in this work are summarized uniformly in the following table (Table 1).

Table 1. Symbols and meanings.

Symbols	Meanings
γ	Relationship between tasks and data
β	Conflicting relationship between data
δ	Positional relationship between nodes
T_m^{Total}	Data access time of the task t_m
T_{avg}^{AC}	Average data access time for M tasks
U_i^{col}	Resource utilization of the i-th compute node
U_j^{sto}	Resource utilization of the j-th storage node
$\overline{U}$	Average resource utilization of all nodes
C	Data conflict degree for all conflicting data

4 Problem Optimization

In Sect. 3, the coordinated placement of meteorological workflow and data has been modeled as a multi-objective optimization problem. In this section, based on NSDE algorithm, this multi-objective problem is optimized. Firstly, we encode the multi-objective optimization problem and generate the initial parental population. Secondly, based on the parental population, the mutation operation, crossover operation, and selection operation are continuously performed. In the selection phase, we adopt fast non-dominated sorting and crowding distance calculation to select individuals whose objective functions are relatively good to retain to the next generation. Finally, through comparing the utility values of multiple excellent individuals, the individual with the best utility value are output as the final result.

4.1 Encoding

According to the total number of compute nodes and storage nodes, the placement strategy of each task and data is encoded as a real number between $[0, N^{col} + N^{sto}]$. And each real number represents the location where the corresponding task or data is placed. After the encoding operation is completed, the placement strategies set $X = \{X^T, X^{OD}\}$ is generated, where $X^T = \{x_0^T, x_1^T, x_2^T, ..., x_m^T, ..., x_{M-1}^T\}$ represents the corresponding compute node locations of M tasks. $X^{OD} = \{x_0^{OD}, x_1^{OD}, ..., x_2^{OD}, ..., x_{P-1}^{OD}\}$ represents the corresponding storage node locations of P input data. In addition, the placement position X^{HD} of all historical data is fixed.

4.2 Objective Functions

As the three objective functions of this optimization problem: the average data access time, the average resource utilization, and the data conflict degree, we

need to find a suitable placement scheme so that all three objective functions are relatively good, not one or two of them are relatively good. The calculation of average data access time is illustrated in Algorithm 1. Then, the NSDE algorithm optimizes the population and finally obtain the best placement strategy.

Algorithm 1. Calculate the Average Data Access Time

Require: X, D, TS, M, γ
Ensure: T_{avg}
1: **for** t_m in TS **do**
2: **for** d_k in γ_m **do**
3: $d = d_k, i = X_m^T, j$ *is position of* d_k
4: *calculate* T_m^{AC} *by* (1)
5: $T_m^{Total} += T_m^{AC}$
6: **end for**
7: **end for**
8: *calculate* T_{avg}^{AC} *by* (3)
9: **return** T_{avg}^{AC}

4.3 Optimizing Problem Using NSDE

As an efficient population-based global optimization algorithm, NSDE is adopted to optimize this multi-objective optimization problem. Firstly, we need to initialize an initial population as the first parental population.

Initialization. The size of population is NP, so this initial population can be expressed as $X = \{X_0, X_1, ..., X_i, ..., X_{NP-1}\}$, where X_i is the i-th individual of population, and represents a placement strategy for all tasks and data. If this optimization problem has M tasks and P user input data, then X_i can be expressed as $X_i = \{x_0, x_1, x_2, ..., x_M, x_{M+1}, ..., x_{M+P-1}\}$ that represents the placement strategies for M tasks and P user input data.

Evolution. Based on the parental population, the *mutation*, *crossover*, and *selection* operations are performed recurrently.

In the *mutation* phase, according to the mutation factor F and three randomly selected individuals X_a, X_b and X_c, the mutation individual H_i is calculated as follows:

$$H_i = X_a + F * (X_b - X_c) \tag{14}$$

Finally, the mutation population $H = \{H_0, H_1, ..., H_i, ..., H_{NP-1}\}$ whose size is also NP is generated.

In the *crossover* operation, according to the specified crossover probability CR, the corresponding genes from the parental individual R_i and the mutation individual H_i are selected to form the crossover individual R_i. The specific

calculation process is as follows:

$$R_{\mathrm{i,j}} = \begin{cases} H_{i,j}, rand(0,1) \leq CR || j = j_{rand} \\ X_{i,j}, \qquad\qquad\qquad Otherwise \end{cases} \tag{15}$$

Finally, the crossover population $R = \{R_0, R_1, ..., R_i, ..., R_{NP-1}\}$ is generated.

In the *selection* operation, based on the population $Y = \{Y_0, Y_1, ..., Y_i, ..., Y_{2NP-1}\}$ merged by the parental population X and the crossover population R, the *fast non-dominated sorting* method is performed for all individuals. Then, all individuals in population Y are divided into multiple non-dominated layers to achieve that all individuals in the lower non-dominated layer have better fitness values than individuals in the higher non-dominated layer. And for each individual in the same layer, we continue to calculate the crowding distance. Finally, the individuals in the lower non-dominated layer are preferentially retained into the next generation parental population X, and secondly the individuals with better crowding distances in the same layer are retained into the next generation parental population X until the size of X is NP.

Iteration. The *mutation*, *crossover*, and *selection* operations are continuously performed based on the parental population X to achieve population evolution, and multiple excellent individuals are obtained finally.

Utility Value Comparison. For the multiple excellent individuals obtained by NSDE, we also need to perform the *utility value comparison* to obtain the optimal individual as the final result. If T^i_{avg}, $\overline{U^i}$ and C^i represent the average data access time, the average resource utilization, and the data conflict degree of X_i, respectively, T^{min}_{avg}, T^{max}_{avg}, $\overline{U^{min}}$, $\overline{U^{max}}$, C^{min} and C^{max} represent the minimum and maximum of the corresponding fitness values, respectively. Therefore, the utility value v_i of X_i can be calculated as following:

$$v_i = \frac{1}{3} * (\frac{T^{\max}_{avg} - T^i_{avg}}{T^{\max}_{avg} - T^{\min}_{avg}} + \frac{\overline{U^i} - \overline{U^{min}}}{\overline{U^{max}} - \overline{U^{min}}} + \frac{C^{max} - C^i}{C^{max} - C^{max}}) \tag{16}$$

where the larger v_i, the better the individual X_i is.

5 Experiment and Analysis

Aiming at the three optimization goals of the coordinated placement problem, we designed a series of experiments and compared CPDE method with another common coordinated placement method in meteorological department. Firstly, we introduce the settings of parameters and another common coordinated placement method used in this experiment. Then, the performance of the two methods is compared and analyzed.

5.1 Parameters Setting and Comparison Method

In this experiment, we optimized three different scale workflows and compared the performance of several optimization methods. Assume that the sizes of workflows are set to 2, 4 and 6, respectively, and each workflow contains 20 tasks, but the data sets required for each task and the ordering of tasks execution are different. The setting of parameters used are as shown in the following table (Table 2).

Table 2. Parameters setting.

Parameter	Value
Number of compute nodes	19
Number of storage nodes	17
Bandwidth of the edge layer	200 MB/s
Bandwidth of the aggregation layer	500 MB/s
Bandwidth of the core layer	1 GB/s
Forwarding power of the switch	5 W

Besides our proposed CPDE method, we also compare performance with another coordinated placement method commonly used by meteorological department, the Coordinated Placement method based on Greedy algorithm (CPG), which is briefly described as follows:

Compared with data conflict degree, CPG is more concerned with average data access time and average resource utilization. Based on historical data that has been placed, tasks are preferentially placed at the computing node closest to their required historical data to ensure that each task has the shortest average data access time for historical data. Secondly, based on the placement of each task, the input data is preferentially placed on the storage node closest to the task set to which is belongs. And for the storage nodes having the same distance, the input data is preferentially placed on the storage node with highest resource utilization. However, our proposed CPDE method estimates the average data access time, the average resource utilization and the data conflict degree comprehensively, and optimizes the placement strategy using NSDE algorithm.

5.2 Comparison and Analysis of Method Performance

In this section, we will compare and analyze the performance of two methods on the three objective functions to demonstrate the superiority of our proposed CPDE method in terms of overall performance.

Figure 3 and Fig. 4 shows the performance comparison of CPDE method and CPG method on the average data access time indicator and the average resource utilization indicator based on three scale data sets. Overall, the difference between two methods in these two performance indicators is not very

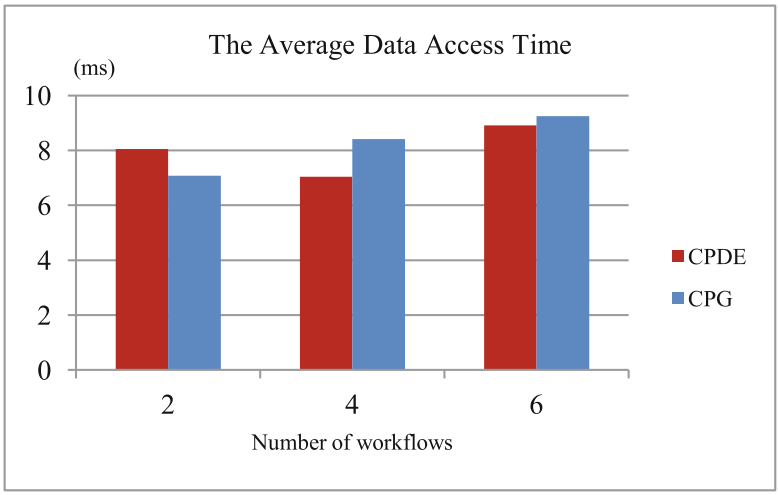

Fig. 3. Comparison analysis of the average data access time.

obvious. Only when the data set is small, for example, the number of workflows is 2, the performance of CPG method on the average data access time indicator is better than CPDE method, but the performance of CPDE method on the average resource utilization indicator is better than CPG method. However, with the size of data set expands, the performance of CPDE method on the average data access time indicator gradually begins to outperform CPG method, but the performance of CPG method on the average resource utilization indicator gradually also begins to outperform CPDE method.

Figure 5 shows the performance comparison of CPDE method and CPG method on the data conflict degree indicator based on three scale data sets. It can be clearly seen that the two methods have a large gap in this performance, and the performance of CPDE method is always significantly better than CPG method.

CPG method prioritizes the average data access time indicator and the average resource utilization indicator, and both of these indicators tend to place all tasks and data centrally to ensure the less data access time and the higher resource utilization. But CPG method does not consider the data conflict degree indicator, because in order to ensure the smaller data conflict degree, it is necessary to disperse the conflicting data, which contradicts the placement principle of CPG method. However, our proposed CPDE method can optimize these three indicators at the same time, so that CPDE method has better comprehensive performance than CPG method.

Finally, it can be determined that our proposed CPDE method is definitely better than CPG method, which has been verified in this experiment.

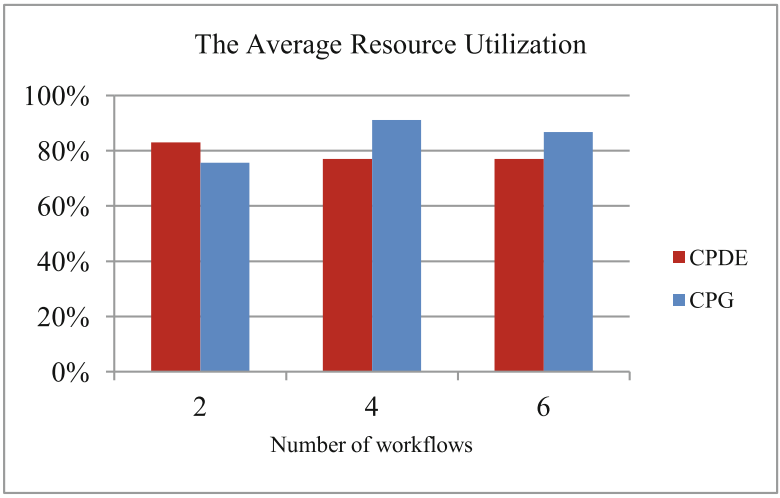

Fig. 4. Comparison analysis of the average resource utilization.

6 Related Work

In order to improve the execution efficiency of applications in the cluster, optimizing the data placement strategy helps to reduce the data access time to the application.

Li et al. proposed a two-stage data placement strategy and adopt the discrete PSO algorithm to optimize the placement of data for reducing data transfer cost [4]. In [7], aiming at the efficient data-intensive applications, an adaptive data placement strategy considering dynamic resource change is proposed, based on the resource availability, this placement strategy can reduce the data movement cost effectively. Ebrahimi et al. proposed a BDAP data placement strategy, which is a population-based distributed data placement optimization strategy [8]. These data placement strategies have a good effect. However, with the rapid increase of applications and data in the cluster, the resource utilization of equipment is also receiving more and more attention. In [10], based on the limited resources, Whaiduzzaman et al. proposed a PEFC method to improve the performance of cloudlet. In [12], Chen et al. proposed a correlation-aware virtual machine placement scheme to enhance resource utilization. In addition, ensuring the stability and security of data in cluster is also receiving increasing attention. In [14], Kang et al. formulated the data placement problem as a linear programming model and developed a heuristic algorithm named SEDuLOUS for solving the Security-aware data placement problem. At the same time, some scholars have conducted comprehensive research on these indicators. In [16], proposes a BPRS big data copy placement strategy, which can reduce the data movement of each data center and improve the load balancing problem.

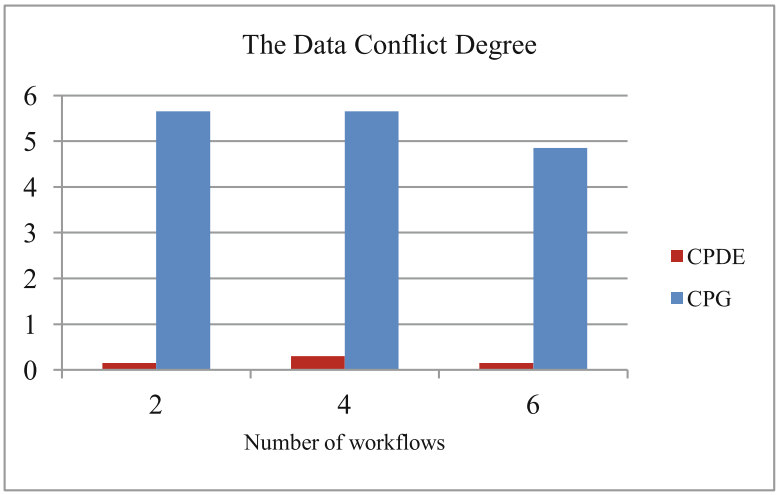

Fig. 5. Comparison analysis of the data conflict degree.

However, to the best of our knowledge, there are still few placement strategies that consider the three important factors of average data access time, average resource utilization and data conflict degree. Therefore, based on these three objectives, this paper proposes to optimize the placement of tasks and data using NSDE algorithm, and achieved remarkable results.

7 Conclusion and Future Work

The meteorological department mainly optimizes the placement of massive meteorological historical data for reducing the average data access time of applications. But it lacks the optimization of the coordinated placement of tasks and input data in each application. Therefore, firstly, the coordinated placement problem of meteorological workflows and data is modeled as a multi-objective optimization problem. And minimizing the average data access time and the data conflict degree, maximizing the resource utilization are used as the three optimization objectives. Secondly, we analyze and construct the models of these three objective functions, respectively. Then, based on NSDE algorithm, we propose a coordinated placement optimization method named CPDE to optimize the multi-objective problem. Finally, by comparing with the commonly used coordinated placement methods of meteorological departments, the availability and superiority of our proposed CPDE method is demonstrated.

However, in the future work, we also need to further consider the energy consumption of the data center and the execution time of each application from the perspective of resource providers and users, respectively. In addition, We consider to appropriately improve our proposed CPDE method to improve the performance of method, such as the optimization speed.

Acknowledgment. This research is supported by the Scientific Research Project of Silicon Lake College under Grant No. 2018KY23.

References

1. Li, X., Li, D., Wan, J., Vasilakos, A.V., Lai, C.-F., Wang, S.: A review of industrial wireless networks in the context of industry 4.0. Wireless Netw. **23**(1), 23–41 (2015). https://doi.org/10.1007/s11276-015-1133-7
2. Lin, B., et al.: A time-driven data placement strategy for a scientific workflow combining edge computing and cloud computing. IEEE Trans. Ind. Inf. **15**(7), 4254–4265 (2019)
3. Tang, J., Tang, X., Yuan, J.: Traffic-optimized data placement for social media. IEEE Trans. Multimedia **20**, 1008–1023 (2017)
4. Li, X., et al.: A novel workflow-level data placement strategy for data-sharing scientific cloud workflows. IEEE Trans. Serv. Comput. **12**, 370–383 (2019)
5. Dong, Y., Yang, Y., Liu, X., Chen, J.: A data placement strategy in scientific cloud workflows. Future Gener. Comput. Syst. **26**(8), 1200–1214 (2010)
6. Deng, K., Kong, L., Song, J., Ren, K., Dong, Y.: A weighted k-means clustering based co-scheduling strategy towards efficient execution of scientific workflows in collaborative cloud environments. In: IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, DASC 2011, 12–14 December 2011, Sydney, Australia, pp. 547–554 (2011)
7. Kim, H., Kim, Y.: An adaptive data placement strategy in scientific workflows over cloud computing environments. In: NOMS 2018–2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1–5 (2018)
8. Kashlev, A., Lu, S., Ebrahimi, M., Mohan, A.: BDAP: a big data placement strategy for cloud-based scientific workflows. In: 2015 IEEE First International Conference on Big Data Computing Service and Applications, pp. 813–820 (2015)
9. Liao, Z., Yu, B., Liu, K., Wang, J.: Learning-based adaptive data placement for low latency in data center networks. In: IEEE 43rd Conference on Local Computer Networks (2018)
10. Whaiduzzaman, M., Gani, A., Naveed, A. PEFC: performance enhancement framework for cloudlet in mobile cloud computing. In: IEEE-ROMA-2014, pp. 224–229 (2014)
11. Xu, X. et al.: A multi-objective data placement method for IoT applications over big data using NSGA-II. In: IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 503–509 (2018)
12. Chen, T., Zhu, Y., Gao, X., Kong, L., Chen, G., Wang, Y.: Improving resource utilization via virtual machine placement in data center networks. Mobile Netw. Appl. **23**(2), 227–238 (2017). https://doi.org/10.1007/s11036-017-0925-7
13. Cui, L., Zhang, J., Yue, L., Shi, Y., Li, H., Yuan, D.: A genetic algorithm based data replica placement strategy for scientific applications in clouds. IEEE Trans. Serv. Comput. **11**(4), 727–739 (2015)
14. Kang, S., Veeravalli, B., Aung, K.M.M.: A security-aware data placement mechanism for big data cloud storage systems. In: IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), pp. 327–332 (2016)

15. Wang, R., Yiwen, L., Zhu, K., Hao, J., Wang, P., Cao, Y.: An optimal task placement strategy in geo-distributed data centers involving renewable energy. IEEE Access **6**, 61948–61958 (2018)
16. Liu, L., Song, J., Wang, H.: BRPS: a big data placement strategy for data intensive applications. In: IEEE 16th International Conference on Data Mining Workshops (ICDMW), pp. 813–820 (2016)
17. Shu, J., Liu, X., Jia, X., Yang, K., Deng, R.H.: Anonymous privacy-preserving task matching in crowdsourcing. IEEE Internet Things J. **5**(4), 3068–3078 (2018)
18. Chi, Z., Wang, Y., Huang, Y., Tong, X.: The novel location privacy-preserving CKD for mobile systems. IEEE Access **6**, 5678–5687 (2018)