Data privacy-based coordinated placement method of workflows and data

Tao Huang\(^1,2\) E-mail: ruishuangtao@163.com, Shengjun Xue\(^2,3\), Yumei Hu\(^4\), Yiran Shi\(^2\), Wei Jin\(^2\)

\(^1\)School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, People’s Republic of China
\(^2\)School of Computer Science and Technology, Silicon Lake College, Suzhou, People’s Republic of China
\(^3\)School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, People’s Republic of China
\(^4\)Shanghai HuanAn Environmental Management Limited Company, Shanghai, People’s Republic of China

Abstract: With the rapid development of data acquisition technology, many industries have already have the characteristics of big data and cloud technology has provided strong support for the storage and complex calculations of these massive data. The meteorological department established the cloud data centre based on the existing storage and computing resources and re-arranged the historical data to reduce the historical data access time of applications. However, the placement of each workflow and input data also affects the average data access time, which in turn affects the computing efficiency of the cloud data centre. At the same time, because of the collaborative processing of multiple nodes, the resource utilisation of cloud data centre has also paid more and more attention. In addition, with the increase of data security requirements, some privacy conflict data should avoid being placed on the same or neighbouring nodes. In response to this challenge, based on the fat-tree network topology, this study proposes a data privacy protection-based collaborative placement strategy of workflows and data to jointly optimise the average data access time, the average resource utilisation, and the data conflict degree. Finally, a large number of experimental evaluations and comparative analyses verify the efficiency of the proposed method.

Nomenclature

Symbols and meanings

\(TS\) tasks set
\(D_{inp}\) input data set
\(D_{his}\) historical data set
\(\gamma\) relationship between tasks and data
\(\beta\) conflicting relationship between data
\(\delta\) positional relationship between nodes
\(N_{col}\) number of compute nodes
\(N_{sto}\) number of storage nodes
\(VM_{MAX}\) resource of each compute node
\(VM_{sto}\) resource of each storage node
\(TVM\) amount of resources required for each task
\(DVM\) amount of resources required for each data
\(T_{Total}\) data access time of the task \(t_m\)
\(T_{AVG}\) average data access time for \(M\) tasks
\(U_{col}\) resource utilisation of the \(i\)th compute node
\(U_{sto}\) resource utilisation of the \(j\)th storage node
\(U_{AVG}\) average resource utilisation of all nodes
\(C\) data conflict degree for all conflicting data

1 Introduction

1.1 Background

With the advancement of data acquisition technology and the improvement of industry service requirements [1–3], the number and types of industry data continue to grow, and many industry data already have the characteristics of big data [4, 5], such as the traffic data, financial data, meteorological data and so on. Based on these massive industry data, the computational complexity of various industrial applications is also increasing [5]. So cloud technology provides strong support for the storage of these massive industry data and the complex calculations of various industry applications. Based on the existing storage and computing resources, many departments established the cloud data centre and offloaded a large number of industrial applications and data to cloud data centre for execution and storage [6, 7]. The powerful computing performance of cloud centre helps to improve the computing efficiency of various applications, but the data transmission between the nodes in the cloud centre also results in additional data access time, especially meteorological data. Whether from the time dimension or the spatial dimension, meteorological data is a kind of high-density big data. As the most important data source for all meteorological workflows, many meteorological workflows usually need access to large amounts of historical data from multiple storage nodes, which also leads to the dependence of meteorological workflows on massive historical big data is much greater than other industry workflows. Therefore, in order to reduce the average data access time of all meteorological applications, the meteorological department has analysed the characteristics of massive historical data and rationally distributed meteorological historical data to each storage node [8, 9]. However, the data access time of each application depends not only on the placement of historical data but also on the placement of each workflow and input data. Based on the overall placement of meteorological historical data, we continue to study how to collaboratively place the input data and tasks of workflows on the corresponding nodes in cloud centre [8, 10], thereby further reducing the average data access time for all tasks in workflows [11, 12]. As the number of meteorological workflows and data offloaded to cloud centre increases rapidly [13, 14], increasing the resource utilisation of active nodes in cloud centre is also being paid more and more attention [15, 16], and it has become an important indicator to measure the performance of placement method [17, 18]. In addition, with the improvement of the confidentiality of meteorological data, there are some privacy conflicts between meteorological data, so that these conflicting meteorological data should avoid being placed on the same or neighbouring storage nodes to ensure the security of these privacy data [19–21]. Therefore, while improving the resource utilisation of nodes, the placement of conflict data is also receiving increasing attention.

ISSN 2398-3396
Received on 11th February 2020
Revised 25th April 2020
Accepted on 19th June 2020
E-First on 21st October 2020
doi: 10.1049/iet-cps.2020.0007
www.ietdl.org
attention. All in all, the collaborative placement of workflows and data based on data privacy has become a challenge. In response to this challenge, this paper proposes a data privacy-based optimisation method for collaborative placement of workflows and data.

1.2 Paper contributions

In this paper, we have made the following contributions:

- Based on the fat-tree network topology, each meteorological application is modelled as a workflow which is a set of sequential tasks, and each task in workflows has a certain dependency on the corresponding historical data or input data.
- Based on the placement of historical data and the conflicted relationship of private data, the coordinated placement problem of meteorological workflows and data is modelled as a constrained multi-objective optimisation problem. In addition, as the three optimisation goals of this optimisation problem, the data access time indicator, the resource utilisation indicator and the data private conflict indicator are all modelled and formulated.
- Aiming at this multi-objective optimisation problem, we propose a coordinated placement optimisation method based on non-dominated sorting differential evolution (NSDE) algorithm and adopt the simple additive weighting (SAW) method and multiple criteria decision making (MCDM) method to determine the final optimal coordinated placement solution.
- The most commonly used coordinated placement optimisation method currently in the meteorological department and the same type of optimisation method as NSDE algorithm recently proposed by other researchers are selected for performance comparison.

The remainder of this paper is organised as follows. Section 2 mainly analyses the meteorological fat-tree network structure and a typical meteorological workflow. The coordinated placement problem is modelled as a constrained multi-objective optimisation problem in Section 3 and is solved based on the NSDE algorithm in Section 4. Section 5 compares and analyses the experimental results. Section 6 summarises the related work. Finally, we outline the conclusions and future work in Section 7.

2 Analysis of meteorological scenarios

2.1 Meteorological fat-tree network

The meteorological network usually adopts the tree structure, but the bandwidth in the traditional network is converged layer by layer, and the network congestion is likely to occur. Therefore, based on the traditional tree network structure, the fat-tree 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 the fat-tree topology network is not convergent so that it can provide high throughput transmission service for the meteorological data centre and ensure that the network is non-blocking. In addition, there are multiple parallel paths between any two nodes, so the fault tolerance of the network is good.

In actual meteorological applications, a public meteorological cloud data centre is constructed based on the virtualisation 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 centre contains \( N^{\text{pod}} \) pods, the number of edge switches and aggregation switches in each pod is \( N^{\text{pod}}/2 \), the number of servers that can be connected in each pod is \( (N^{\text{pod}}/2)^2 \), and the number of core switches is also \( (N^{\text{pod}}/2)^2 \). Fig. 1 shows a meteorological fat-tree network topology.

![Meteorological fat-tree network with four pods](image)

![Workflow of weather forecast production](image)

with four pods. In practical applications, the network size of the meteorological department is usually much larger than this.

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 analyses 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 applications' execution is also an important data source for each application. Therefore, based on the placed meteorological historical data, the coordinated placement of workflows and input data in the meteorological applications is completed, so that the average data access time and the data conflict degree are minimised, the average resource utilisation of all active nodes is maximised.

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

As the starting task, \( t_0 \) represents the task of data collection operation, including automatic station data, radar data, satellite nephogram and so on. Tasks \( t_1 \) and \( t_2 \) represent the historical weather summary operation and historical weather analysis operation, that summarise the historical weather phenomena and analyse the causes of historical weather formation in the past 48 h, respectively. Tasks \( t_3 \) and \( t_4 \) represent the real-time weather summary operation and real-time weather analysis operation, that summarise the current weather phenomena and analyse the causes of current weather formation, respectively. Task \( t_5 \) represents the forecast mode calculation operation, the future weather is calculated in real time based on the European Centre for Medium-Range Weather Forecasts (ECMWF) and global forecasting system. Tasks \( t_6 \) and \( t_7 \) represent the weather situation analysis operation and meteorological elements analysis operation, that analyse the future weather situation and the future meteorological elements based on the calculation results of the forecast model, respectively. Task \( t_8 \) represents the generation of forecast model conclusions operation, based on the analysis for weather situation and meteorological elements, the final conclusions of the forecast model is formed. As the termination task, task \( t_9 \) represents the
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 modelling and formulation

In this section, based on the fat-tree network topology, we mainly model this data privacy-based coordinated placement problem as a constrained multi-objective optimisation problem and formulate three optimisation goals: the average data access time, the average resource utilisation and the data private conflict degree, respectively.

3.1 Problem modelling

Assume that a meteorological workflow consists of M tasks, which can be defined as TS = {t0, t1, t2, ..., tM−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 Dinp = {dinp0, dinp1, dinp2, ..., dinpP−1} and Dinhs = {dinh0, dinh1, dinh2, ..., dinhQ−1}. Therefore, the relationship between these conflicting data can be expressed as D = {Dinp, Dinhs}. Then, the positional relationship between M tasks and data can be expressed as γ = {γ0, γ1, γ2, ..., γM−1}, where γm = {dinh0, dinh1, dinh2, ..., dinhQ−1} represents the data set required for the nth task tm. In addition, if there are K pairs of conflicting data, the conflicting relationship between these conflicting data can be expressed as \( \beta = [\beta_0, \beta_1, \beta_2, ..., \beta_K] \), where \( \beta_k = \{d_k, d_\beta, d_i, d_j \in \{D_{inp}, D_{inhs}\}\} \) represents the nth pair of conflicting data, and as the conflicting data, data d_i and d_j should be placed on two storage nodes that are far away in the fat-tree network to ensure the security of these conflicting data.

3.2 Data access time model

In the meteorological fat-tree network, it is assumed that task tm and its required data d are stored in the compute node u_i and the storage node v_j, respectively, and the amount of data is \( d_u \). The positional relationship between \( u_i \) and \( u_j \) can be defined as \( \delta_{ij} \), then:

- If \( u_i \) and \( u_j \) are the same nodes, then \( \delta_{ij} = 0 \).
- If \( u_i \) and \( u_j \) belong to the same switch of the same pod, then \( \delta_{ij} = 1 \).
- If \( u_i \) and \( u_j \) belong to the different switches of the same pod, then \( \delta_{ij} = 2 \).
- If \( u_i \) and \( u_j \) belong to the different pods, then \( \delta_{ij} = 3 \).

Therefore, according to the positional relationship \( \delta_{ij} \) between \( u_i \) and \( u_j \), the access time \( T^{AC} \) of task \( t_m \) for the data \( d_u \) can be expressed as

\[
T^{AC}_{m} = \begin{cases} 0, & \delta_{ij} = 0 \\ 2 \ast d_u / B_{se}, & \delta_{ij} = 1 \\ 2 \ast (d_u / B_{ac} + d_u / B_{ac}), & \delta_{ij} = 2 \\ 2 \ast (d_u / B_{ac} + d_u / B_{ac} + d_u / B_{ac}), & \delta_{ij} = 3 
\end{cases}
\]

(1)

where \( B_{se}, B_{ac} \) 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. Also, \( d_u \) represents the amount of data that need to be transferred.

Therefore, the total data access time of task \( t_m \) for its required data set \( y_m \) can be calculated as

\[
T^{Total}_{m} = \sum_{d_u \in y_m} T^{AC}_{m}
\]

(2)

Then, the average data access time for \( M \) tasks can be calculated as

\[
T^{Avg}_{m} = \frac{\sum_{m=0}^{M-1} T^{Total}_{m}}{M}
\]

(3)

3.3 Resource utilisation model

Assume that the number of compute nodes and storage nodes are \( N^{col} \) and \( N^{sto} \), respectively. In addition, the resource of each compute node and storage node is \( VM_{MAX} \) and \( VM_{MAX} \), respectively. The resources required for \( M \) tasks and \( P + Q \) data are expressed as \( TVM = \{tv_{m0}, tv_{m1}, ..., tv_{mP-1}, tv_{mQ-1}\} \) and \( DVM = \{dv_{m0}, dv_{m1}, ..., dv_{mP-1}, dv_{mQ-1}\} \), respectively. \( tv_{m0} \) and \( dv_{m0} \) represent the resources required for the nth task and the nth data, respectively.

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

\[
CT[i, m] = \begin{cases} 1, & t_m \text{ is placed on the }i\text{th compute node} \\ 0, & \text{otherwise} \end{cases}
\]

(4)

\[
SD[j, n] = \begin{cases} 1, & d_n \text{ is placed on the }j\text{th storage node} \\ 0, & \text{otherwise} \end{cases}
\]

(5)

Then, the resource utilisation of the ith compute node and the jth storage node can be expressed as \( U^c_{i} \) and \( U^s_{j} \), respectively:

\[
U^c_{i} = \sum_{m=0}^{M-1} tv_{i, m} \ast CT[i, m]/VM_{MAX} \quad \text{(6)}
\]

\[
U^s_{j} = \sum_{n=0}^{P+Q-1} dv_{j, n} \ast SD[j, n]/VM_{MAX} \quad \text{(7)}
\]

If the nodes with non-zero resource utilisation are defined as active nodes, assume that the number of active compute nodes and active storage nodes are \( N^{active} \) and \( N^{active} \), respectively. The average resource utilisation of the currently active compute nodes and storage nodes can be expressed as \( \overline{U}^c \) and \( \overline{U}^s \), respectively:

\[
\overline{U}^c = \frac{\sum_{i=0}^{K^{active}} U^c_{i}/N^{active}}{N^{active}} \quad \text{(8)}
\]

\[
\overline{U}^s = \frac{\sum_{j=0}^{K^{active}} U^s_{j}/N^{active}}{N^{active}} \quad \text{(9)}
\]

Finally, since the number of compute nodes and storage nodes may be different, based on the SAW method, the average resource utilisation \( \overline{U} \) of all nodes is calculated as

\[
\overline{U} = \overline{U}^c + \overline{U}^s \quad \text{(10)}
\]

where \( \overline{U}^c + \overline{U}^s = 1 \). The larger the proportion of compute nodes in all nodes, the larger the corresponding weight \( \overline{U}^c \) or vice versa. Therefore, in this experiment, the two weights \( \overline{U}^c \) and \( \overline{U}^s \) are set as follows:

\[
\overline{U}^c = \frac{N^{active}}{N^{active} + N^{active}} \quad \text{(11)}
\]

\[
\overline{U}^s = \frac{N^{active}}{N^{active} + N^{active}} \quad \text{(12)}
\]
3.4 Data conflict model

The closer the privacy conflict data is placed on the 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 or neighbouring nodes.

Based on current data placement, 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 situations of conflicting data.

Then, the data conflict degree \( C \) for all conflicting data can be expressed as

\[
C = w_0 * N^{SN} + w_1 * N^{SS} + w_2 * N^{SP}
\]

(13)

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 \( w_0, w_1 \) and \( w_2 \) are set to 0.55, 0.3 and 0.15, respectively.

3.5 Problem formulation and constraint

In this paper, the coordinated placement of meteorological workflows and data with privacy conflict protection has been modelled as a multi-objective optimisation problem. In addition, our optimisation goals are to minimise the average data access time calculated in (3) and the data conflict degree calculated in (13) and to maximise the average resource utilisation of active nodes calculated in (10). Therefore, this optimisation model can be expressed as

\[
\text{Min}(T_{avg}, \overline{U}), \text{Max}(C)
\]

(14)

In addition, since the resources of each computing node or storage node are limited, the used resources of each node cannot exceed its capacity. This coordinated placement optimisation problem is also a constrained multi-objective optimisation problem, and the constraint can be expressed as

\[
s.t. \ \forall U_i^{\text{cod}} \leq 1, \ \forall U_j^{\text{sto}} \leq 1 \quad 0 \leq i < N^{\text{cod}}, 0 \leq j < N^{\text{sto}}
\]

(15)

which represents that the resource utilisation rate of any computing node or storage node cannot exceed 100%.

Finally, the symbols used in this work are summarised uniformly in the Nomenclature.

4 Problem optimisation

In Section 3, the coordinated placement of meteorological workflows and data has been modelled as a constrained multi-objective optimisation problem. In this section, the coordinated placement method based on NSDE algorithm (CPDE) is proposed.

First, we encode the optimisation problem and generate an initial population as the first generation parental population. Second, based on the parental population, the mutation, crossover and selection operations are continuously performed. Through mutation and crossover operations, the better solutions are constantly looked for in the entire solution space. 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, the utility values of multiple excellent individuals are compared based on the SAW method and MCDM method, and the individual with the best utility value is output as the final optimal coordinated placement solution.

4.1 Encoding

In the encoding phase, the placement strategies for all tasks in workflows and input data are encoded as the real numbers, each real number can indirectly represent the location where the corresponding task or data is placed. Therefore, the placement strategies for \( M \) tasks and \( P \) data can be encoded as a placement strategies set

\[
X = \{X^{T}, X^{OD}\}
\]

where \( X^T = \{x_{t0}^*, x_{t1}^*, \ldots, x_{t_M}^*, \ldots, x_{T_P}^*\} \) represents the placement strategies of \( M \) tasks, and the placement positions of tasks must be the compute nodes, \( X^{OD} = \{x_{OD0}, x_{OD1}, \ldots, x_{ODM}, \ldots, x_{ODP}\} \) represents the placement strategies of \( P \) user input data, and the placement positions of these data must be those storage nodes. In addition, the placement position \( x_{OD}^{\text{ID}} \) of \( Q \) historical data is fixed.

Assume that the IDs of \( N^{\text{cod}} \) compute nodes and \( N^{\text{sto}} \) storage nodes are encoded as \([0, 1, 2, \ldots, A - 1]\) and \([A, A + 1, A + 2, \ldots, A + B - 1]\), respectively. So the range of all genes can be set to \([0, A + B]\). However, since all tasks and input data must be placed on the compute nodes and the storage nodes, respectively, these genes in the individual cannot fully represent the placement positions of the corresponding tasks and data. In order to facilitate the calculation of the NSDE algorithm, each gene in the individual needs to be converted into the ID of the compute node or the storage node. For example, the genes representing the placement positions of tasks need to be converted to IDs of compute nodes between \([0, A]\), and the genes representing the placement positions of input data need to be converted to IDs of storage nodes between \([A, A + B]\). Therefore, the conversion relationship between the gene \( x_{i,j} \) and the placement position \( p_{i,j} \) is as follows:

\[
p_{i,j} = \begin{cases} 
\text{floor}\left(\frac{x_{i,j} * N^{\text{cod}}}{N^{\text{cod}} + N^{\text{sto}}}\right), & x_{i,j} \text{ is a gene of a task} \\
\text{floor}\left(\frac{x_{i,j} * N^{\text{sto}}}{N^{\text{cod}} + N^{\text{sto}}}\right) + N^{\text{cod}}, & x_{i,j} \text{ is a gene of a data} 
\end{cases}
\]

(16)

where the function floor(\(X\)) represents \(X\) is rounded down.

Finally, the individual \( x_{i,j} = \{x_{i,j_0}, x_{i,j_1}, \ldots, x_{i,j_{P-1}}\} \) can be converted into a vector \( p_{i} = \{p_{i,0}, p_{i,1}, \ldots, p_{i,M \times P - 1}\} \) that can directly represent the placement node ID of each task and data. Table 1 illustrates the individual encoding and its conversion through a simple example, where \( x_{i,j} \) is the encoding of all genes and \( p_{i} \) is the placement node IDs of the corresponding tasks or data after conversion by (16).

Based on (4) and (5), the vector \( p_{i} \) can also be expressed as the two-dimensional matrices \( \text{CT}[8, 10] \) and \( \text{SD}[4, 8] \) as follows:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
\end{bmatrix}
\]

4.2 Objective functions

In this constrained multi-objective optimisation problem, there are three objective functions: average data access time, average resource utilisation and data conflict degree, which are represented by 3, 10 and 13, respectively. However, we need to find a suitable
solution that achieves a balance between the three objective functions, so that these optimisation goals are relatively good, not one or two of them are relatively good. The calculation of average data access time of all tasks. First, for each task calculated (Line 5). Finally, the average data access time of all nodes and the storage nodes. First, the resource utilisation each storage node is calculated based on the placement location of all data (Lines 4–6). Then, the average resource utilisation of all compute nodes is calculated based on the placement location (Lines 2–4). Then, the total data access time of all tasks (Lines 1–3). In addition, the resource utilisation of all data (Lines 4–6). Then, the average resource utilisation of all compute nodes and storage nodes (Line 8).

4.3 Optimising problem using NSDE

As an efficient population-based multi-objective optimisation algorithm, NSDE algorithm is adopted to optimise this multi-objective optimisation problem. First, we need to initialise an initial population as the first parental population.

4.3.1 Initialisation: Assume the size of the population is NP, so this initial population can be expressed as $X = \{X_0, X_1, \ldots, X_{N_{\text{Pop}}-1}\}$, where $X_i$ is the $i$th individual of the population and represents a placement strategy for all tasks and data, so the size of the individual depends on the total number of tasks and input data. If this optimisation problem has $M$ tasks and $P$ user input data, then $X_i$ can be expressed as $X_i = \{x_{i0}, x_{i1}, \ldots, x_{iM}, x_{iM+1}, \ldots, x_{iM+P-1}\}$ that represents the placement strategies for $M$ tasks and $P$ user input data, and the placement strategy of each task or data is a gene in the individual $X_i$.

Based on the initial population as the first generation parental population, NSDE algorithm begins to perform mutation, crossover and selection operations recurrently.

4.3.2 Mutation: In the mutation phase, first, three different individuals $X_a$, $X_b$ and $X_c$ are randomly selected from the population $X$. Then, based on the mutation factor $F$, the mutation individual $H_a$ is generated by combining the difference vector of the two individuals with the third individual. Therefore, the mutation operation can be expressed as

$$H_i = X_a + F \times (X_b - X_c)$$

(17)

Finally, based on (17), we can generate the mutation population $H = \{H_0, H_1, \ldots, H_{N_{\text{Pop}}-1}\}$ whose size is also NP.

4.3.3 Crossover: In the crossover phase, according to the crossover factor CR, we select several genes from the mutation population $H$ and the parental population $X$ to form the crossover population $R = \{R_0, R_1, \ldots, R_{N_{\text{Pop}}-1}\}$ whose size is also NP.

First, in order to ensure that at least one gene in the crossover individual $R_j$ is from the mutation individual $H_a$, we need to randomly select a gene $H_{i,j}$ from $H_a$ and keep it to $R_{j}$. Then, for other genes in the crossover individual $R_j$, based on the crossover factor CR, we choose genes from the parental individual $X_i$ and the mutation individual $H_a$, respectively. Therefore, the crossover operation can be expressed as

$$R_{i,j} = \begin{cases} H_{i,j}, & \text{rand}(0, 1) \leq CR \| j = j_{\text{rand}} \\ X_{i,j}, & \text{otherwise} \end{cases}$$

(18)

Finally, based on (18), the crossover population $R = \{R_0, R_1, \ldots, R_{N_{\text{Pop}}-1}\}$ is generated.

4.3.4 Selection: In the selection phase, the parental population $X$ and the crossover population $R$ are merged into a new population $Y = \{Y_0, Y_1, \ldots, Y_{N_{\text{Pop}}-1}\}$ whose size is 2NP. According to

Table 1 Encoding and conversion

| Parameter            | Value |
|----------------------|-------|
| number of tasks      | 10    |
| number of data       | 8     |
| number of compute nodes | 8    |
| number of storage nodes | 4    |
| range of genes       | [0, 12) |
| IDs of compute nodes | {0, 1, 2, 3, 4, 5, 6, 7} |
| IDs of compute nodes | {8, 9, 10, 11} |
| $x_i$                | {1.4, 3.5, 8.7, 5.6, 10.2, 11.9, 2.8, 6.9, 0.3, 1.8, 11.9, 7.9} |
| $p_i$                | {0, 2, 5, 3, 6, 7, 1, 4, 0, 1, 11, 10, 8, 6, 9, 10, 11} |
4.3.5 Iteration: generation parental population

Based on the fast non-dominated sorting, each individual in the population \( Y \) are divided into multiple dominant layers \( L_i \) (\( i = 0, 1, 2, \ldots \)). In addition, all individuals in the \( L_i \) layer can completely dominate all individuals in the \( L_{i+1} \) layer. This also means that all individuals in \( L_i \) are better than all individuals in \( L_{i+1} \), considering any one performance indicator. Therefore, all individuals in the \( L_i \) layer are more likely to be retained in the next generation population than individuals in the \( L_{i+1} \) layer. However, in the same dominating layer \( L_i \), we perform the crowding distance calculation for each individual, the individuals with better crowding distance can be preferentially retained to the next generation population.

Therefore, individuals in the better dominating layer and individuals with better crowding distances in the same dominating layer are preferentially retained to the next generation parental population \( X \) until the size of \( X \) is also NP.

4.3.6 Utility value comparison: For the multiple Pareto solutions generated after the optimisation of the NSDE algorithm, we also need to perform the utility value comparison for choosing an optimal solution as the final solution of this coordinated placement problem. However, these Pareto solutions are not dominated by each other, so it is difficult to accurately select an individual as the final solution. Therefore, the SAW method and MCDM method are employed to evaluate these individuals and help us select the final optimal solution.

If \( T_{avg} \), \( U \) and \( C \) represent the three objective function values of the individual \( X \), respectively, \( T_{avg} \), \( U_{avg} \), \( C_{avg} \), \( T_{max} \), \( U_{max} \), \( C_{max} \), \( C_{min} \) and \( C_{emin} \) represent the minimum and maximum of the three objective function values of all individuals, respectively. Therefore, the utility value of three objective functions can be expressed as the following:

\[
u_i = \frac{T_{max} - T_{avg}}{T_{avg} - T_{min}}
\]

According to the importance of these three utility values, the corresponding weights \( w_t \), \( w_u \) and \( w_c \) are set, respectively. In addition, it is also regarded as the first generation parental population (Line 2). Based on the parental population \( X^{g} \) where \( g \) is the evolution time, the mutation operation is performed and the mutation population \( H^{g} \) is generated (Line 4). We continue to perform the crossover operation and generate the crossover population \( R^{g} \) based on the parental population \( X^{g} \) and the mutation population \( H^{g} \) (Line 5). Then, the parental population \( X^{g} \) and the crossover population \( R^{g} \) are merged into a new population \( Y^{g} \) (Line 6), the fast non-dominated sorting is performed based on this population \( Y^{g} \) and all individuals in \( Y^{g} \) are divided into multiple dominant layers \( L_i \) (Line 7). The individuals in better dominant layers can be preferentially retained to the next generation population parental populations \( X^{g+1} \). Finally, we calculate the crowding distance of each individual in the same dominated layer \( L_i \) and keep better individuals in the next generation parental population \( X^{g+1} \) until the size of \( X^{g+1} \) is also NP (Lines 9–13). Constant iteration until the termination condition is satisfied, and the excellent individuals set \( S \) is output (Line 16).

5 Comparison and analysis of experimental results

In this section, in order to evaluate the performance of our proposed CPDE method, we perform a series of experiments and compared the CPDE method with two other common coordinated placement methods. First, we introduce the settings of parameters and these three common coordinated placement methods used in this experiment. Then, the performance of the three methods is compared and analysed.

5.1 Parameters setting and comparison methods

In this experiment, the compute virtual machine uses a custom second-generation Intel Xeon scalable processor (Cascade Lake) with a sustained full-core Turbo frequency of 3.6 GHz, while the storage virtual machine uses high-frequency Intel Xeon E5-2686 v4 (Broadwell) processor with a base frequency of 2.3 GHz. Among them, each compute node and storage node has two CPUs, and the memory size is set to 4 and 16 GB, respectively. By means of the CloudSim simulation tool, the collaborative placement strategies are simulated on a desktop PC with Intel Core i7-8700 3.20 GHz processors and 16 GB RAM.

In order to compare and evaluate the performance of different methods, we optimise workflows of different sizes, so the sizes of the workflows are set as 5, 10, 15 and 20, respectively. Also, for data sets of different sizes, the task distribution in the workflows conforms to a normal distribution between 5 and 20, the total number of simultaneously placed tasks reaches between 50 and
250. Finally, the setting of parameters used is as shown in Table 2, and data set settings used are as shown in Table 3, where $N_{WS}$, $TN_{w}$ and $TN_{total}$ represent the number of workflows, number of tasks in each workflow and the total number of tasks, respectively.

Besides our proposed CPDE method, we also compare performance with two other coordinated placement methods: the coordinated placement method based on the greedy algorithm (CPG) which is commonly used by the meteorological department and the coordinated placement method based on NSGA-II algorithm (CPGA) which is recently proposed by other researchers, and it is also an optimisation method of the same type as our proposed CPDE method. These three coordinated placement optimisation methods are briefly described as follows:

- **CPDE**: Our proposed CPDE method estimates the average data access time, the average resource utilisation and the data conflict degree comprehensively, and optimises the placement strategy using NSDE algorithm.
- **CPG**: Compared with data conflict degree, CPG method is more concerned with the average data access time and the average resource utilisation. 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. Second, based on the placement of each task, the input data are preferentially placed on the storage node closest to the task set to which belongs. In addition, for the storage nodes having the same distance, the input data are preferentially placed on the storage node with the highest resource utilisation.
- **CPGA**: CPGA method is close in function to our CPDE method, this comparison method is mainly proposed by Xu et al. [22]. Therefore, as an optimisation algorithm of the same type as NSDE, it is used to compare the performance of our CPDE method.

### 5.2 Comparison and analysis of method performance

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

Fig. 6 shows the performance comparison of three placement methods on the average data access time indicator based on four scale data sets. Overall, the difference between the three methods in this performance indicator is not very obvious. First, CPDE and CPGA are two optimisation methods of the same type, although their slightly better than the CPGA method in the average data access time indicator. Second, based on the Greedy algorithm, CPG method prioritises the historical data access time and then completes the placement of input data based on tasks that have been placed. Therefore, CPG method lacks the consideration of the joint placement of tasks and input data. This method can maintain good performance on the data access time indicator in small-scale data sets, such as the number of workflows is 5. However, with the gradual expansion of workflow scale, CPG method gradually loses its advantage in the data access time indicator, and CPDE method has gradually become the optimal method among the three placement methods.

Fig. 7 shows the performance comparison of three placement methods in the average resource utilisation indicator based on four scale data sets. First, CPDE method is always slightly better than CPGA method in this indicator. Second, based on the Greedy algorithm, CPG method prioritises the historical data access time, the decentralised placement of historical data leads to the decentralisation of tasks placement, which in turn leads to the decentralisation of input data placement. Therefore, the resource utilisation of each node is low in small-scale data sets, for example, when the number of workflows is 5. However, as the size of data set gradually increases, the resource utilisation of each node also increases significantly, and the advantages of Greedy algorithm are gradually revealed, especially when the number of workflows is 15, CPG method has the best performance. However, in other cases, for example, when the number of workflows is 5, 10 and 20, respectively, in these three placement methods, the CPDE method always maintains optimal performance.

Fig. 8 shows the performance comparison of three methods on the data conflict degree indicator based on four scale data sets. It can be clearly seen that the three methods have a large gap in this performance indicator. First, it is obvious that the CPG method has
the worst performance in this indicator, CPG method prioritises the average data access time indicator and the average resource utilisation indicator, these two indicators tend to place all tasks and data centrally to ensure the less data access time and the higher resource utilisation. However, it 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. Second, CPDE and CPGA are two optimisation methods of the same type. Although their performance is still relatively close on this indicator, the CPGA method is slightly better than the CPDE method, for example, the number of workflows is 5, 15 and 20. CPDE method only shows the best performance when the number of workflows is 10.

In summary, it can be clearly seen that although the performance of three methods differs in the average data access time and average resource utilisation indicators, the gap is not very large. However, in the data conflict degree indicator, the performance of CPG method is far inferior to the other two methods. In addition, CPGA method and CPDE method are based on two optimisation algorithms of the same type. Although their performance on these three indicators has always been similar, more often, CPDE method is superior to the CPGA method. Therefore, based on data sets of different sizes, we make statistics on the optimal placement methods of the three performance indicators, as shown in Table 4. It can be clearly seen that the CPDE method exhibits the best performance most times, followed by CPGA method and CPG method.

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

6 Related work

In order to improve the execution efficiency of applications in cluster optimising the data placement strategy helps to reduce the data access time to the application. Li et al. [4] proposed a two-stage data placement strategy and adopt the discrete particle swarm optimisation algorithm to optimise the placement of data for reducing data transfer cost. In [7], an adaptive data placement strategy considering dynamic resource change for efficient data-intensive applications was proposed, based on the resource availability, this method can reduce the data movement cost-effectively. Ebrahimi et al. [8] proposed a BDAP data placement strategy, which is a population-based distributed data placement optimisation strategy. These data placement strategies have a good effect. However, with the rapid increase of applications and data in the cluster, the resource utilisation of equipment is also receiving more and more attention.

In [11], based on the limited resources, Whaiduzzaman et al. proposed a PEFC method to improve the performance of cloudlet. In [13], Chen et al. proposed a correlation-aware virtual machine placement scheme to enhance resource utilisation. In addition, ensuring the stability and security of data in the cluster is also receiving increasing attention. In [17], Kang et al. formulated the data placement problem as a linear programming model and developed a heuristic algorithm named SEDAUS for solving the security-aware data placement problem. At the same time, some scholars have conducted comprehensive research on these indicators. In [19], the authors proposed a BPRS big data copy placement strategy, which can reduce the data movement of each data centre and improve the load-balancing problem.

average data access time, average resource utilisation and data conflict degree. Therefore, based on these three objectives, this paper proposes to optimise the placement of tasks and data using the NSDE algorithm, and achieved remarkable results.

7 Conclusion and future work

In order to reduce the average data access time of meteorological applications, the meteorological department mainly places the massive meteorological historical data, but it lacks the optimisation of tasks and data placement for each application. Therefore, first, we model the placement of tasks and data in meteorological applications as a constrained multi-objective optimisation problem. Second, we analysed and constructed the data access time model, resource utilisation model and data conflict degree model and optimised the constrained multi-objective problem using the NSDE algorithm. Finally, the effectiveness of our proposed CPDE method is verified by comparison and analysis of multiple sets of experiments.

Based on the work done in this paper, we also need to continue to optimise the placement of tasks and data to reduce the average data access time and the data conflict degree, while improving the average resource utilisation. In addition, we also need to constantly update our method based on actual performance.

8 References

[1] Li, X., Li, D., Wan, J., et al.: ‘A review of industrial wireless networks in the context of industry 4.0’, Wirel. Netw., 2017, 23 (1), pp. 23–41
[2] Lin, B., Zhu, F., Zhang, J., et al.: ‘A time-driven data placement strategy for a scientific workflow combining edge computing and cloud computing’, IEEE Trans. Ind. Inf., 2019, 15 (7), pp. 4254–4267
[3] Tang, J., Tang, X., Yuan, J.: ‘Traffic-optimized data placement for social Media’, IEEE Trans. Multimed., 2018, 20 (4), pp. 1008–1023
[4] Li, X., Zhang, Z., Wu, Y., et al.: ‘A novel workflow-level data placement strategy for data-sharing scientific cloud workflows’, IEEE Trans. Serv. Comput., 2019, 12 (3), pp. 370–383
[5] Yuan, D., Yang, Y., Liu, X., et al.: ‘A data placement strategy in scientific cloud workflows’, Future Gener. Comput. Syst., 2020, 100 (120), pp. 1202–1206
[6] Deng, K., Kong, L., Song, J., et al.: ‘A weighted K-means clustering based co-scheduling strategy towards efficient execution of scientific workflows in collaborative cloud environments’, 2011 IEEE Ninth Int. Conf. on Dependable, Autonomic and Secure Computing, Sydney, Australia, 2011, pp. 547–554
[7] Kim, H., Kim, Y.: ‘An adaptive data placement strategy in scientific workflows over cloud computing environments’, NOMS 2018 – 2018 IEEE/IFIP Network Operations and Management Symp., Taipei, Taiwan, 2018, pp. 1–5
[8] Ebrahimi, M., Mohan, A., Kashlev, A., et al.: ‘BDAP: a big data placement strategy for cloud-based scientific workflows’. 2015 IEEE Int. Conf. on Big Data Computing Service and Applications, Redwood City, CA, USA, 2015, pp. 105–114
[9] Liu, K., PENG, J., WANG, J., et al.: ‘A learning-based data placement framework for low latency in data center networks’, IEEE Trans. Cloud Comput., 2019, pp. 1–14
[10] Xu, X., Fu, S., Qi, L., et al.: ‘An IoT-oriented data placement method with privacy preservation in cloud environment’, J. Netw. Comput. Appl., 2018, 124, pp. 148–157
[11] Whaiduzzaman, M., Gani, A., Naveed, A.: ‘PEFC: performance enhancement framework for cloudlet in mobile cloud computing’. 2014 IEEE Int. Symp. on Robotics and Manufacturing Automation (ROMA), Kuala Lumpur, Malaysia, 2014, pp. 224–229
[12] Xu, X., Gu, R., Yuan, Y., et al.: ‘A multi-objective data placement method for IoT applications over big data using NSGA-II’. 2018 IEEE Int. Conf. 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), Halifax, NS, Canada, 2018, pp. 503–509
[13] Chen, T., ZHU, Y., GAO, X., et al.: ‘Improving resource utilization via virtual machine placement in data center networks’, Mobile Netw. Appl., 2018, 23 (2), pp. 227–238
[14] Gong, W., Qi, L., Xu, Y.: ‘Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment’, Wirel. Commun. Mobile Comput., 2018, 18 (4), pp. 1–8
[15] Cui, L., Zhang, J., Yue, L., et al.: ‘A genetic algorithm based data replica placement strategy for multi-cloud applications in clouds’, IEEE Trans. Serv. Comput., 2018, 11 (4), pp. 727–739
[16] He, D., Kumar, N., Khan, M., et al.: ‘Efficient privacy-aware authentication scheme for Mobile cloud computing services’, IEEE Syst. J., 2018, 12 (2), pp. 1621–1631
[17] Kang, S., Veeravalli, B., Aung, K.M.M.: ‘A security-aware data placement mechanism for big data cloud storage systems’. 2016 IEEE Second Int. Conf. on Big Data Security on Cloud (BigDataSecurity), New York, NY, USA, 2016, pp. 327–332

Table 4 Parameter settings

| Indicators          | The number of workflows |
|---------------------|-------------------------|
| average data access | CPG CPDE CPDE CPDE     |
| average resource    | CPDE CPDE CPG CPG       |
| utilisation         | CPDE CPDE CPGA CPGA     |
| data conflict       | CPDE CPDE CPGA CPGA     |

This is an open access article published by the IET under the Creative Commons Attribution License
[18] Wang, R., Lu, Y., Zhu, K., et al.: ‘An optimal task placement strategy in geodistributed data centers involving renewable energy’, *IEEE Access*, 2018, 6, pp. 61948–61958

[19] Liu, L., Song, J., Wang, H., et al.: ‘BRPS: a big data placement strategy for data intensive applications’. 2016 IEEE 16th Int. Conf. on Data Mining Workshops (ICDMW), Barcelona, Spain, 2016, pp. 813–820

[20] Shu, J., Liu, X., Jia, X., et al.: ‘Anonymous privacy-preserving task matching in crowdsourcing’, *IEEE Internet Things J.*, 2018, 5 (4), pp. 3068–3078

[21] Chi, Z., Wang, Y., Huang, Y., et al.: ‘The novel location privacy-preserving ckd for mobile crowdsourcing systems’, *IEEE Access*, 2018, 6, pp. 5678–5687

[22] Xu, X., Fu, S., Li, W., et al.: ‘Multi-objective data placement for workflow management in cloud infrastructure using NSGA-II’, *IEEE Trans. Emerg. Top. Comput. Intell.*, 2020, PP (99), pp. 1–11