Mathematical Biosciences and Engineering

MBE, 19(12): 12212–12231.
DOI: 10.3934/mbe.2022568
Received: 09 April 2022
Revised: 19 July 2022
Accepted: 04 August 2022
Published: 22 August 2022

http://www.aimspress.com/journal/MBE

Research article

A method to calculate the number of dynamic HDFS copies based on file access popularity

Xi-yue Cao[1,2], Chao Wang[1,2,*], Biao Wang[1,2] and Zhen-xue He[1,3]

[1] School of Information Science and Technology, Hebei Agricultural University, Baoding 071001, China
[2] Hebei Urban Forest Health Technology Innovation Center, Baoding 071001, China
[3] Hebei Key Laboratory of Agricultural Big Data, Baoding 071001, China

* **Correspondence:** Email: bdking@sina.com; Tel: +8613831206517.

Abstract: HDFS heterogeneous clusters usually have multiple storage media at the same time. How to efficiently read and write file copies and reasonably use various storage media is a problem to be solved. Dynamically adjusting the number of copies is important in HDFS, which can solve the problem of accessing a large number of hot files at the same time and improve the efficiency of cluster services. A method is introduced to calculate the number of dynamic HDFS copies based on file access popularity in this paper. Firstly, an algorithm was proposed to predict file popularity based on the cuckoo search optimization Markov model. The unbiased grey model is used to predict the accessing file's popularity at the next moment according to the recent access of the file. The cuckoo search is used to optimize the Markov model, and the prediction error is corrected. Then, the calculation method of the number of copies is designed based on the prediction of the popularity of the file to be accessed and the availability of the node. The experiment shows that the proposed method has a high fitting degree with the actual value, and the MAPE is 3.08%, and it is the smallest, compared with several commonly used prediction models. In CloudSim4.0 simulation platform, multiple users write 10 files to the cluster at the same time, and the change number of copies is calculated according to the predicted value at the next moment, so as to improve the user access efficiency.

Keywords: popularity prediction; unbiased grey prediction; Markov model; cuckoo search; number of copies

1. Introduction

With the development of society and the improvement of computer storage and data processing ability, the explosive growth of data has become an important feature. Faced with the growing scale of massive data, the storage and management of massive data has been a concern. In order to improve the reliability and access efficiency of HDFS, replica technology is commonly used to copy multiple data files and store them on multiple nodes of the distributed file system. For files of different popularity in distributed system, how to dynamically adjust the number of copies so it can not only ensure the reliability and availability of data, but also improve the data access efficiency? The system should be able to adjust the number of copies when the predicted access popularity value is changed, with the number of copies increased for high popularity files and the number of copies reduced for low popularity files, so as to improve the access efficiency and save storage resources. The solution of the above problems is of great significance in improving performance of the cluster.

File popularity is used to describe the degree to which files are required by users in a time period, and it is defined by the frequency of file access in a certain period of time [1], i.e., the times the file is accessed by the user in unit time. Domestic and foreign scholars mainly use neural network models [2,3], grey models [4], deep learning models and so on in file popularity prediction. In References [5,6], aiming at the shortcomings of static copy strategy, a prediction model based on grey model is proposed to predict the future access popularity of data according to the latest access characteristics of files. The frequency of file access is regarded as the popularity of file access. According to temporal locality, the recently accessed files will be accessed again in a short time, and the algorithm calculated file's accessing temperature by analyzing file's accessing frequency within the specified time, and improved the performance of data service [7,8]. Literature [9,10] used combination forecasting model to deal with the shortcomings of single forecasting model. The neural network model in Reference [11] needs to set up accurate levels and a large number of sample space for training, which will affect the time of file popularity prediction. A method was proposed based on LSTM deep learning model to predict the popularity of files [12], which divides the dynamic time window to construct a time series of file access features, and predicts the access trends of different data. With a subjective and objective popularity calculation method, Reference [13] dynamically predicted the file access volume through LSTM neural network model, calculated the file popularity through the predicted file access volume, and adjusted the copy storage strategy.

In the aspect of replica dynamic management, Veeravalli et al. [14] proposed a CDRM dynamic copy management model, which adjusts the number and storage location of data copies through dynamic changes in node storage space capacity and load conditions. Aiming at the low efficiency of randomly selecting the source node and target node for replication in the default copy creation process of HDFS in Hadoop, Higai et al. [15] proposed a copy management strategy based on a single ring structure. The copy management strategy of this method is 25% more efficient than the default method.

In the current methods of file popularity prediction, the BP neural network prediction tends to get stuck on local optimum, and the accuracy of grey prediction model is not enough. The time feedback model and the prediction method with decay factor are suitable for short-term prediction, while the prediction accuracy is not high. The deep learning LSTM requires a long training time. In the light of the above problems, aiming at the static copy management strategy and the default copy placement strategy, the paper created the unbiased grey model to predict the future access popularity of the file, and proposed an optimization algorithm based on the cuckoo search which can optimize the Markov

model to correct the grey prediction value, then the number of copies is adjusted by the popularity prediction value. The main innovations of this paper are listed as follows:

(1) This paper builds a prediction model of file popularity based on the unbiased grey model. This model has the sequence and uncertain character, and has high prediction accuracy.

(2) This paper presents an optimized algorithm. The Markov model is optimized by cuckoo search algorithm to correct the predicted values and improve the accuracy of prediction.

(3) This paper presents a new method to compute the number of copies. This method dynamically adjusts the number of copies and improves the access efficiency.

The remainder of this paper is organized as follows. Section 2 introduces file access popularity prediction method. In Section 3, we introduce how to calculate the number of copies. Section 4 is about the determination of the number of file copies. Section 5 reports experimental results, and conclusions of this paper are made in the last section.

2. Preliminary preparations

In order to deal with the problem of low access efficiency caused by the sudden increase of hot file access, the method of predicting the popularity of the file at the next moment in advance is adopted to adjust the number of copies in time to improve user access efficiency [16,17]. In the past research, a single method model generally has limitations and doesn't work well in addressing certain problems. Many researchers use the combined model to cope with the limitations of the single model and achieve better results [18]. In our research, the grey model is adopted which can predict the data development trend through a small amount of simple calculation and incomplete information, and predicts the file popularity at the next moment through the latest access characteristics of the file [19]. In order to further improve the accuracy of the prediction data, the cuckoo search algorithm is used to optimize the Markov model and modify the prediction data to improve the prediction accuracy.

2.1. Grey-Markov model

The grey model is suitable for the prediction model of the system with uncertain factors. For the grey system with known and unknown information, the grey differential prediction model is established by using a small amount of incomplete information to predict the development and change of the system behavior eigenvalue and the grey process related to time series which changes in a certain range. Grey process finds the law of system change through correlation analysis, establishes the corresponding differential equation model, and predicts the future development trend. The grey prediction model uses equal time interval to obtain a series of quantitative values reflecting the characteristics of the prediction object to construct the grey model prediction data set. This paper takes the file access popularity as the original data, and predicts the future development trend of the file popularity through the grey model processing.

Markov model is a stochastic process of things discovered by Russian mathematicians. The change process of this kind of things is only related to the recent state. Markov model can predict the future development trend according to the probability of transition between different states. In the process of state transition, the state after the nth transition only depends on the previous result state, which has nothing to do with the original state of the system and the Markov process before the transition. In this paper, unbiased grey model is used to predict the file popularity. Because the

randomness of user access and the sudden change of file popularity are two inevitable factors, the grey prediction model may produce large errors. Markov model is used to predict the state of file popularity in the future, and the error of file popularity prediction value is corrected.

2.2. Cuckoo search algorithm

Cuckoo search (CS) algorithm [20,21] is a natural heuristic algorithm proposed by Yang and Deb. It has the advantages of fast convergence, high efficiency and few adjustment parameters. This method is based on the cuckoo's parasitic brooding behavior and Lévy flight, which is more effective than genetic algorithm and particle swarm optimization. Lévy flight can expand the search range, increase the diversity of the population, and it is easier to jump out of the local point. The algorithm regards all host nests as a generation, and each nest carries bird eggs as a solution. The goal is to find the potential optimal solution through continuous iteration to replace the existing scheme.

This paper uses the cuckoo search algorithm for multi-objective optimization, based on the following three criteria:

(1) Each cuckoo lays K eggs at a time (representing the solution of K goals) and puts them in a randomly selected host nest;

(2) The best nest with high quality eggs (solutions) in each generation is brought to the next generation;

(3) The number of available host nests is fixed, and the probability of finding cuckoo eggs is $p_a \in (0,1)$. After discovery, the host can destroy the eggs or give up the old nest to build a new one.

Each egg in the nest represents a solution, and each cuckoo can lay only one egg. The purpose is to use a new or better solution to replace the bad one in the nest. The CS search algorithm is used to explore the relationship between the Markov state interval, the actual value and the predicted value to find the most suitable value of the interval and to modify the predicted value. The problem is a multi-objective optimization problem, and the optimal solution is used to divide the Markov model state interval.

3. File popularity prediction

The grey Markov model is widely used in the prediction of small sample and exponential distribution sample data. In a period of time, the access frequency of popularity files is like this. Therefore, the paper establishes the file popularity prediction model from grey model parameters, Markov correction value and prediction sequence update, and finally optimizes the model.

3.1. Establishment of unbiased grey prediction model

The unbiased grey model is used to process the file access sequence and predict the file popularity in the future. With the help of grey development coefficient α and grey action quantity U of grey model, the original data series fitting model of unbiased grey prediction model is determined, and the future popularity of documents is predicted. The original data sequence of unbiased grey model is the access of a file in the first m equal time periods.

Set the access volume of file I in the kth time period, and establish the original access volume sequence according to the access volume of file I in the first m equal time periods at the current time,

the original sequence data is represented by the upper corner mark "0".

$$f_i^{(0)} = \left\{ f_i^{(0)}(1), f_i^{(0)}(2), f_i^{(0)}(3), \ldots, f_i^{(0)}(m) \right\}$$

(1) Original sequence accumulation operation. Add the first and second data of the original sequence to get the second data of the accumulated sequence, add the third data and the second data to get the third data and so on. The calculation function is shown in (1).

$$f_i^{(1)}(k) = \sum_{m=1}^{k} f_i^{(1)}(m), (k = 1,2,3, \ldots, m) \tag{1}$$

(2) Smoothness test and quasi exponential test. Before establishing the model, we need to test the smoothness of the original sequence and the series test of the accumulated sequence. If it meets the requirements, we can further establish the model.

For smoothness test, time-series smoothness ratio is defined as follows:

$$\rho(k) = \frac{f^{(0)}(k)}{f^{(1)}(k-1)}, \ k = 3,4, \ldots, m \tag{2}$$

If $f^{(1)}(k)$ $meets$ $\frac{\rho(k+1)}{\rho(k)} < 1$, $\rho(k) \in [0, \ \varepsilon \ ]$, $\varepsilon < 0.5$, $k = 2,3, \ldots, m$, $f^{(1)}(\mathrm{k})$ is smooth data sequence.

For the quasi exponential test, the class ratio is defined as follows:

$$\sigma(k) = \frac{f^{(0)}(k)}{f^{(0)}(k-1)}, k = 3,4, \ldots, m \tag{3}$$

If a sequence ratio satisfies $\sigma(k) \in [1, 1.5], k = 3,4, \ldots, m$, then the sequence $f^{(1)}(k)$ is regarded as exponential distribution.

(3) The trend of accumulation sequence can be described by the whitening differential equation shown in Eq (4).

$$\frac{df_i^{(1)}(m)}{dt} + af_i^{(1)}(m) = u \tag{4}$$

Where parameters a and u are identified, and a is called development coefficient, u is the amount of grey action.

(4) Model solving. Parameters a and u are computed by the least square method. Let $F_m = \left[f_i^{(0)}(2), f_i^{(0)}(3), f_i^{(0)}(4), \ldots, f_i^{(0)}(m) \right]^T$, $\hat{\alpha} = \binom{a}{u}$, $\hat{\alpha}$ is estimator of parameter vector.

$$B = \begin{bmatrix} -\frac{1}{2}\left[f_i^{(1)}(1) + f_i^{(1)}(2) \right] & 1 \\ -\frac{1}{2}\left[f_i^{(1)}(2) + f_i^{(1)}(3) \right] & 1 \\ \cdots & \cdots \\ -\frac{1}{2}\left[f_i^{(1)}(\mathrm{n}-1) + f_i^{(1)}(n) \right] & 1 \end{bmatrix} \tag{5}$$

Substituting $\hat{\alpha}$ into the Eq (6), compute the value F_m.

$$F_m = B\hat{\alpha} \tag{6}$$

(5) The whitening differential equation is solved, and Eq (4) is solved to obtain the unbiased grey

prediction discrete-time response function, which is shown in formula (7).

$$\hat{f}_i^{(1)}(k+1) = \left[f_i^{(0)}(1) - \frac{u}{a}\right] e^{-at} + \frac{u}{a} \quad k = 0,1,2,\dots,m-1 \tag{7}$$

(6) The parameters b and A are calculated according to the estimated values a and u in grey model, and eliminate the inherent deviation. The calculation method is shown in Eq (8).

$$b = ln\frac{2-a}{2+a}, \ A = \frac{2u}{2+a} \tag{8}$$

(7) The unbiased grey prediction model of original data is shown in Eq (9).

$$\hat{f}_1^{(0)} = f_1^{(0)}, \hat{f}_{k+1}^{(0)} = Ae^{bk}, k = 1,2,3,\dots,m \tag{9}$$

By updating the parameters, the unbiased grey prediction model does not need to reduce and restore the prediction data, which simplifies the modeling steps and improves the operation efficiency of the system.

3.2. The correction on Grey-Markov-model prediction value based on CS algorithm

In the Markov model, the error can be corrected by selecting the center value of the state interval where the grey prediction value is located. However, only considering the center value of the state interval may ignore the influence of other error distribution information, and the center value of the state interval cannot properly represent the error state. Therefore, with the help of cuckoo search algorithm, the grey prediction value of Markov model can be modified more accurately.

The partition of state interval is determined by relative precision, which is the difference between actual value and predicted value. Each data is distinguished according to its own relative precision. To adapt to the prediction of different data series, the boundary value of the interval between the first and the last states is selected as the maximum value. The target interval is divided equally into 3–5.

Suppose $[a_i, b_i]$ is the *ith* state interval, the objective function makes the error of prediction correction v_i as small as possible, and it is defined as follows (10).

$$v_i = \alpha_i a_i + (1-\alpha_i)b_i, \ i = 1,2,\dots,n \tag{10}$$

The better parameter α_i of each interval is found by Cuckoo search algorithm. When $\alpha = 0.5$, the correction value is the center value of the state interval.

The specific process of cuckoo search algorithm to optimize Grey-Markov model is as follows.

(1) Location update

In CS algorithm, cuckoos find the nest and lay eggs by the Lévy flight search. The invincible walk is a walk between long and short steps. Therefore, the nest updating position is calculated by Lévy flight as follows:

$$x_i^{t+1} = x_i^t + \beta \otimes \text{Lévy}(\lambda, s) \tag{11}$$

$$s = \frac{u}{|v|^{\frac{1}{\lambda}}}$$

$$\text{Lévy}(\lambda, s) = \frac{\lambda\Gamma(\lambda)\sin\left(\frac{\pi\lambda}{2}\right)}{\pi s^{1+\lambda}}, \lambda \in (1,2)$$

Where t is the number of iterations and x_i^t is the position of the *ith* point at time *t* and β is the step, $u \sim N(0, \sigma_u^2)$, $v \sim N(0, \sigma_v^2)$, $\sigma_u^2 = \left(\frac{\Gamma(1+\lambda)\sin\left(\frac{\pi\lambda}{2}\right)}{\Gamma\left[2^{\frac{(\lambda-1)}{2}} \cdot \lambda^{\frac{(1+\lambda)}{2}}\right]}\right), \sigma_v = 1$, $\Gamma(z) = \int_0^\infty \frac{t^{z-1}}{e^t} dt$, for $|s| \geq |s_0|$, *s* submits to Lévy distribution, s_0 is the min step, $s_0 \gg 0$, but in fact, if s_0 is reasonable, it is feasible, for example $s_0 \in (0.1,1)$.

The host bird rebuilds the nest after discovering the alien bird with a certain probability p_a, because the choice of direction is random, the probability of each direction is the same, obeying the uniform distribution. A relatively small value is selected by a large probability in the Lévy distribution, so this location can be approximately satisfied with the Mantegna algorithm. Therefore, the position of the new bird's nest is defined as follows:

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \epsilon) \otimes \left(x_j^t - x_k^t\right) \tag{12}$$

Among them, t is the number of iterations and x_i^t is the *ith* position at time *t*, α is the step, *s*, ϵ are random numbers that obey even distribution, *H* is the Heaviside function (13) derived from even distribution, ⊗ is the point multiplication, x_j^t and x_k^t is the location of two bird nests randomly selected at *t* time.

$$Heaviside(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{13}$$

(2) Definition of Cuckoo search objective function

The objective function is represented by the mean absolute value of relative percentage error (MAPE). The state interval is divided according to the residual between the grey prediction value and the actual value. Each generation is determined by the minimum MAPE value; the prediction value is modified according to the residual correction after each iteration. The objective function is defined as follows (14).

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|f_i^{(0)}(k) - \hat{F}_i^{(0)}(k)|}{f_i^{(0)}(k)}\right) \tag{14}$$

Where $\hat{F}_i^{(0)}(k)$ is the prediction correction value.

(3) Calculation of correction value

The correction value of each state interval is to calculate by the optimal decision interval coefficient $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ in the first step. According to formula (15), the correction values $v_1, v_2, \ldots, v_n$ are obtained.

$$\begin{cases} v_1 = \alpha_1 a_1 + (1 - \alpha_1) b_1 \\ v_2 = \alpha_2 a_2 + (1 - \alpha_2) b_2 \\ v_3 = \alpha_3 a_3 + (1 - \alpha_3) b_3 \end{cases} \tag{15}$$

(4) Determination of predicted value

Through the k-step state transition probability matrix and the state E_i of the current data and its initial vector $X_{(0)}$, the state of the predicted value at the next moment can be predicted. Assuming $max(P_{ij}) = P_{ik}$, the predicted data is most likely to change from state E_i to E_k at the next moment.

Each state interval is divided according to the relative accuracy, and the unbiased grey forecast is corrected according to the relative accuracy to reduce the error and increase the accuracy of the forecast. The formula is as follows (16).

$$\hat{F}^{(0)}(m+1) = \hat{f}^{(0)}(m+1) + v_i \tag{16}$$

Among them, v_i is the correction value of state i interval, $\hat{F}^{(0)}(m+1)$ represents the correction value of the final predicted value, which is the final predicted value of file access popularity.

Algorithm 1. Cuckoo search algorithm based on Lévy flight.

Input: Number of host nests n; number of iterations m; probability of detection p_a
Output: Optimal solution X_i

```
1.begin
2.  Initial population n host nests X_i (i = 1,2, ... , n);
3.  Calculate fitness F_i (i=1,2,…,n);
4.  While (not true)
5.      Using Lévy flight to generate a new solution X_i;
6.      Calculate the fitness value F_i of the new solution X_i;
7.      Select candidate solution X_j,
8.          if(F_i > F_j)
9.              F_j replaces F_i;
10.     end
11.     Discard the bad solution according to the detection probability p_a;
12.     Generate new nest (solution) by local random search;
13.     Preserve the best solution (Nest of high quality solutions);
14.     Find the current best X_i;
15.  end While
16. end
```

3.3. Data update

This paper predicts the data file access popularity based on the grey prediction model, and the data file is a small sample. Data access has the time locality characteristic, and the current frequently accessed files will be accessed again in a certain period of time in the future, so the older data can't reflect the current trend, and the data needs to be updated.

Files in the distributed system are being accessed all the time, and the latest file access data will be generated at any time. In order to be able to use historical popularity to predict future file popularity more accurately, we deal with the original data sequence by metabolism according to the idea of high weight of new data in Reference [17]. When updating the data sequence, add the latest file popularity to the original sequence and eliminate the old popularity data. Repeat the above operations according to the file statistical cycle to update the file popularity forecast sequence. It can better reflect the recent trend of the file's popularity, and more accurately predict the file's future popularity.

4. Determination of the number of file copies

In general, the number of replicas is 3 in HDFS replica management policy, which is inefficient when a large number of files are accessed. Dynamic adjustment of replica number based on file popularity and file system availability can provide reliable and fast data access environment for users and improve efficiency in distributed file system.

In the static replica management strategy, the replica number is 3 and it is fixed value, which will not change in the whole life cycle from replica creation to destruction. However, in fact, the fixed number of copies does not meet the requirement, and it should be dynamically adjusted according to the user's access characteristics and demand changes [19]. Appropriate number of copies can not only improve the readability and availability of data, but also improve the task execution ability of MapReduce. With the dynamic change of file popularity, the copy number of high popularity files should be increased dynamically to improve the transaction processing ability of the system under concurrent access. When the file popularity is reduced, the copy number should be reduced to save storage space and maintenance cost.

The replica adjustment strategy includes replication, keeping and deletion of copies. Replication is to increase the number of copies and copy the copies of files to new nodes, in order to respond to a large number of access requests of files in time; copy keeping means that the number of copies of the document remains unchanged; copy deletion is to delete some copies of files on the premise of ensuring storage reliability when the popularity of file access decreases [22].

4.1. Calculation of the number of copies based on file popularity

Because the current prediction number of replicas is lagged, we should adjust the number of replicas with the prediction of the number at the next moment, so that we can solve the cyber source competition and congestion caused by the sudden access, and reduce the task execution delay caused by the request source competition. The strategy predicts the file access heat of the next time according to the previous n access heat values in the distributed file system before the current time, so as to adjust the number of copies.

The number of copies is determined by the relationship between the predicted access heat of files and the average access heat of all files in the system. In order to ensure the data reliability of cloud storage system, each file has the minimum number of copies C, $C = 2$. The number of replicas for each file is defined by the formula (17).

$$N_{adjust}(i) = \left\lceil \frac{C \times H_{(i)}}{H_{avg}} \right\rceil - N_{exist}(i) \tag{17}$$

Where, $N_{adjust}(i)$ is the number of copies of file i, $H_{(i)}$ is the file popularity of file i predicted by the prediction model at the next moment, H_{avg} is the average access popularity of all files in the distributed file system, $N_{exist}(i)$ is the number of existing copies of file i.

4.2. Replica factor adjustment based on availability

Most researchers consider the file popularity rather than the availability of cloud computing system in the fixed number of copies in the static copy strategy. Through the quantitative analysis of

system availability parameters and the availability of data blocks, the relationship function between the availability of file system and the number of copies is obtained.

(1) In the distributed file system, suppose that the file F is divided into n data blocks $\{b_1, b_2, \ldots, b_n\}$, which are stored on different nodes. Assuming that the data block b_k has r_j copies, and the r_j copies are placed on m nodes ($j \geqslant m$), only when all the copy storage nodes of the data block are unavailable, the data block is bad. Because each node in the distributed file system is independent each other, $P(B_k)$ is the availability probability of data block k, and $P(N_k)$ is the availability probability of data node k. Therefore, the calculation for the unavailability probability of the data block b_k is shown in (18).

$$P(\overline{B_k}) = P(\overline{N_1}) \times P(\overline{N_2}) \times \cdots \times P(\overline{N_m}) \tag{18}$$

In the cloud storage system, the reliability of nodes changes exponentially with time. In a distributed file system, it is assumed that the reliability of data nodes is an exponential distribution (19) in the life cycle T of the file.

$$f(\mathrm{T}) = e^{-\alpha T} \tag{19}$$

Where α is the failure rate of the node in the file system, and $f(T)$ is the effective expected value of the node in the life cycle of the file.

(2) The availability of a file is that every data block of the file is available. The failure of any data block will lead to the unavailability of the file. The file availability calculation formula is shown in (20).

$$P(\bar{F}) = P(\overline{B_1} \cup \overline{B_2} \cup \cdots \cup \overline{B_n}) \tag{20}$$

Each data block is independent of each other, and the formula (21) is obtained.

$$P(\bar{F}) = \sum_{i=1}^{n} (-1)^{i+1} C_n^i \left(\prod_{l=1}^{r_j} \left(1 - e^{-\alpha_l T_{r_j}} \right) \right)^i \tag{21}$$

(3) The minimum number of copies can be obtained according to the file availability. The formula is shown in (22).

$$P(F) = 1 - \ P(\bar{F}) \geq P_{standard} \tag{22}$$

Where $P_{standard}$ is the standard value of file availability.

4.3. Algorithm process

The final number of copy factors is determined by integrating the copy adjustment model based on file popularity. The 4.2 can ensure the minimum value of the copy numbers under the condition of file reliability, combined with the copy numbers obtained by the file popularity decision, and under the condition of ensuring file reliability, adjust the file copy numbers to improve the system response efficiency and system storage space utilization while ensuring the reliability of the distributed file system. The specific process of the dynamic adjustment of the copy numbers is shown in Figure 1. The minimum number of copies n_1 is calculated under the condition of ensuring the availability of the file, and the copy number n_2 is obtained by combining the relationship between the predicted value of the file's future popularity and the average value of all files in the cluster. When the number of

copies needs to be increased, the larger value is selected between n_1 and n_2. When we reduce the number of copies, the reduction n_3 is equal to the original number of copies minus the forecast numbers based on file popularity prediction. When n_3 is less than 2, the minimum number of copies is 2 by the multiple copy mechanism of the distributed file system.

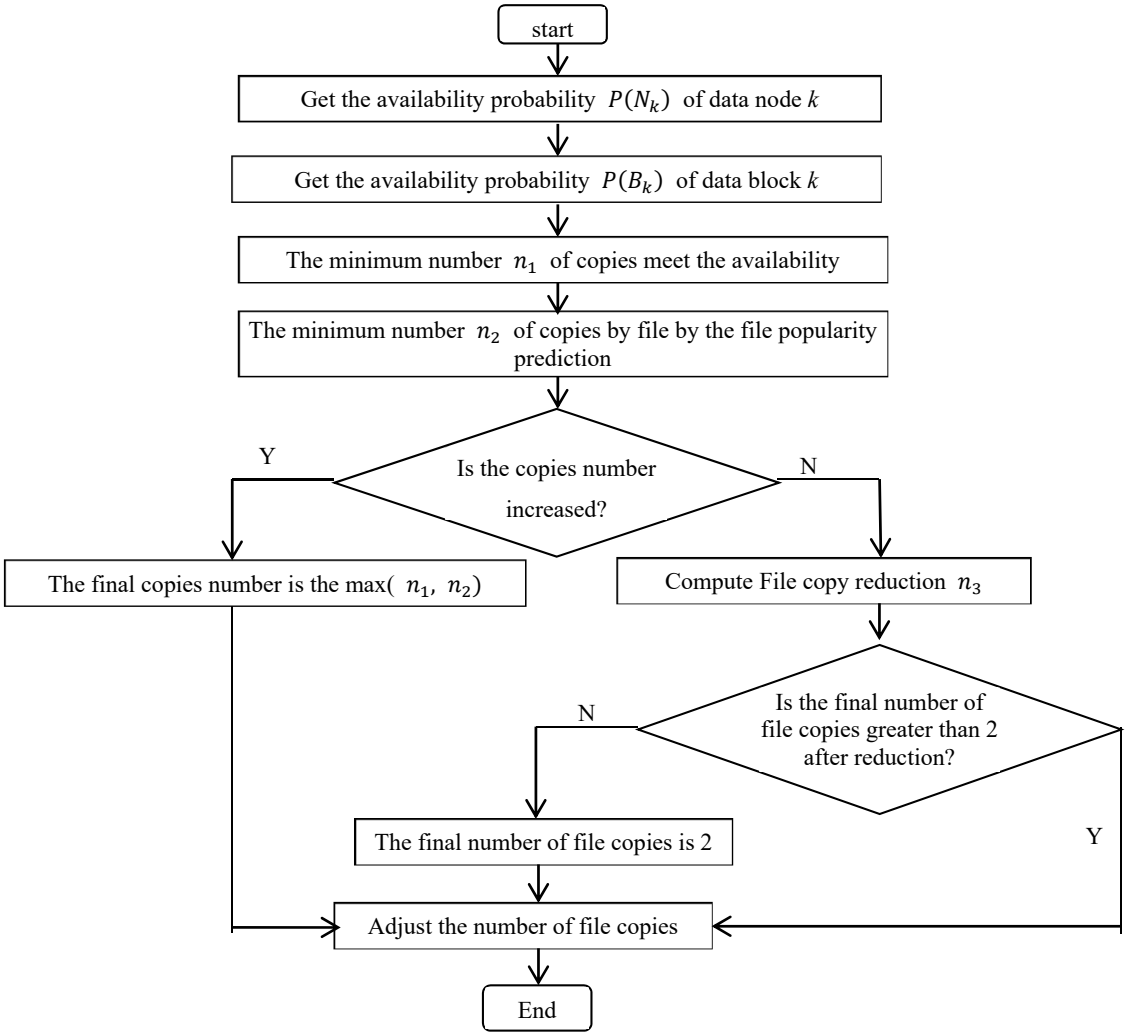

Figure 1. Flow chart of the dynamic adjustment of the copies number.

5. Experiment and result analysis

In this section, we present a series of experiments to verify the algorithm innovation points in this paper. In order to achieve a real simulation of the changes in user access to files, we collect the data from real-time data of a social communication software, as shown in Table 1. The data to the user's access to a certain file (file popularity) is collected every ten minutes, which is used as the original data sequence before processing. First, accumulate the original sequence to get a new sequence, and perform smooth and exponential detection on the new sequence. Through formulae (2) and (3) and test standards, it is verified that the cumulative sequence is a smooth sequence and has an exponential distribution.

Table 1. Real-time visits to hot news by users of a social communication platform.

Serial number/period	1	2	3	4	5	6	7	8
Pageview/ten thousand times	324.01	366.7	447.19	370.52	289.23	382.11	388.01	381.45

CloudSim4.0 is used as the platform to verify the effect of the dynamic HDFS copies placement strategy based on file access popularity. The operating system is Windows 10. The development tool is Eclipse. The version of JDK is 1.8.0. In the experiment, ten simulated heterogeneous data nodes are distributed on three racks, with nodes marked as DN1 to DN10 and racks marked as Rack0, Rack1 and Rack2. One rack contains 4 nodes and the other two racks contain 3 nodes respectively. The detailed information of the nodes and racks is shown in Table 2.

Table 2. Node configuration information.

DataNode	CPU		RAM/MB	Disk Space /GB	Rack
	Basic Frequency/GHz	core			
1	2.0	8	6144	1024	0
2	2.5	6	4096	512	0
3	2.0	6	8192	1024	0
4	2.5	4	4096	512	0
5	2.5	4	4096	1024	1
6	2.0	8	8192	512	1
7	2.0	6	6144	1024	1
8	2.5	4	4096	512	2
9	2.5	8	8192	1024	2
10	2.0	6	4096	512	2

The experiments are listed as follows:

- Experiment 1: State interval division verification;
- Experiment 2: Unbiased grey prediction model optimized by CS verification;
- Experiment 3: File copies number verification;
- Experiment 4: File copies adjustment efficiency verification;
- Experiment 5: Performance test and result analysis.

5.1. Experiment 1: State interval division verification

According to the grey model and Markov model, we have realized the prediction of future file visits to the original data sequence. Among them, $a = 0.0061$, $u = 385.1279$. Unbiased grey prediction model parameters $b = -0.0061, A = 383.9498$. According to the prediction and fitting results to the original data sequence, we select the two maxima of the residuals as the boundary of the state interval, and they are equally divided into three intervals, corresponding to the three states of E1, E2 and E3. The number of states is the same as the nest number of the CS algorithm. The division of the state interval is shown in Table 3 and the residual of the unbiased grey Markov model are given

respectively in Table 4.

Table 3. Markov modified state interval division.

State	Interval
E1	[−85.41, −34.30)
E2	[−34.30, 16.81)
E3	[16.81, 67.92)

Table 4. Residual error and state of unbiased grey model predicted value and actual value.

Serial number	1	2	3	4	5	6	7	8
residual	0	−14.90	67.92	−6.43	−85.41	9.76	17.94	13.64
status	E2	E2	E3	E2	E1	E2	E3	E2

5.2. Experiment 2: unbiased grey prediction model optimized by CS verification

(1) The parameters of CS algorithm

Considering the efficiency and accuracy of the algorithm, the parameters of CS algorithm are obtained after many experiments. The parameter settings are shown in Table 5.

Table 5. CS algorithm parameter settings.

Parameter	Nest capacity	Nest number	Step	Random step	Detection probability	Iterations
value	3	25	0.01	1.5	0.25	200

(2) Evaluation of experimental results

The prediction model is evaluated by MAPE (mean absolute percentage error). MAPE can not only determine whether the prediction effect of the model is good, but also can be used as the basis for the comparison between the models.

(3) The model correction value

Through the CS algorithm iterative optimization, when the number of iterations is 100, it can show the expected effect. Considering the accuracy and the efficiency of the experiment, the number of iterations is 200, and $\alpha = (\alpha_1, \alpha_2, \alpha_3) = (0.9883, 0.4183, 0.4889)$, and the optimized Markov correction value $v = (-84.81,\ -4.56,\ 42.93)$.

Note: The Markov residual correction value obtained from the experiment is only suitable for the current prediction sequence. When the prediction sequence changes, it is necessary to recalculate α to achieve the optimal effect.

(4) Comparison of experimental data

Table 6 shows the comparison of experimental data based on Grey Markov model, unbiased grey Markov model and CS unbiased grey Markov model.

Let x_i is the ith element of the original sequence, y_i is the predicted value of x_i, e_i is the error, r_i is the relative error. The error and the relative error can be defined as follows:

$$e_i = y_i - x_i \quad i = 1,2,\dots n$$

$$r_i = \frac{|e_i|}{x_i} \qquad i = 1,2,\dots\dots n \tag{23}$$

Finally, MAPE is used to evaluate the fitting degree of the three models.

Table 6. Comparison of experimental data based on the three grey prediction models.

Sequence number	x_i	Grey Markov model			Unbiased Grey Markov model			CS Unbiased Grey Markov model		
		y_i	e_i	r_i	y_i	e_i	r_i	y_i	e_i	r_i
1	324.01	324.01	0.00	0.0000	324.01	0.00	0.0000	324.01	0.00	0.0000
2	366.70	381.97	— 15.27	0.0168	372.45	−5.75	0.0157	376.94	— 10.24	0.0279
3	447.19	379.63	67.56	0.0572	421.64	25.55	0.0571	422.20	24.99	0.0559
4	370.52	377.31	−6.79	0.0000	367.80	2.72	0.0073	372.29	−1.77	0.0048
5	289.23	375.00	— 85.77	0.0883	314.78	— 25.55	0.0883	289.83	−0.60	0.0021
6	382.11	372.70	9.41	0.0485	363.24	18.87	0.0494	367.69	14.42	0.0377
7	388.01	370.42	17.59	0.0629	412.07	— 24.06	0.0620	413.00	— 24.99	0.0644
8	381.45	368.16	13.29	0.0587	358.66	22.79	0.0597	363.15	18.30	0.0480
Prediction value	362.89	356.80	6.09	0.0168	356.41	6.48	0.0179	361.00	1.89	0.0052
MAPE		4.36%			4.47%			3.08%		

Note: The x_i and y_i unit is ten thousand times in Table 6.

From Table 6, we can see that the optimization effect of the CS Unbiased Grey Markov model is superior to the Grey Markov model and the unbiased grey Markov model. the MAPE of the Grey Markov model and the Unbiased Grey Markov model are both about 4.4%, indicating that the Markov correction model has a good effect on reducing the error of the grey model. The Markov model optimized by the CS algorithm improves the error correction, and the MAPE is only 3.08%.

In order to further verify that CS unbiased grey Markov model can effectively predict the next moment data in the case of small samples. Using the same group of data, we do the experiment based on the grey model, unbiased grey model, grey Markov model, unbiased grey Markov model, BP neural network and grey neural network. The experimental results are shown in Figure 2 and Table 7.

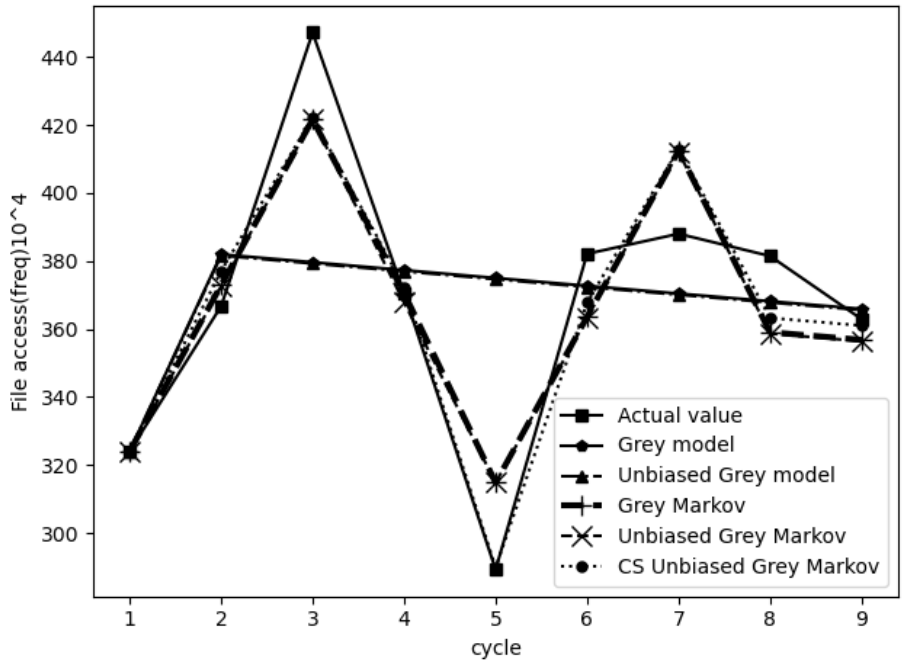

Figure 2. Comparison of fitting effects of five grey models.

Table 7. Comparison of MAPE values of grey relevant models.

Model	Grey model	Unbiased grey model	Grey Markov model	Unbiased grey Markov model	Grey neural network	CS Unbiased Grey Markov model
MAPE	7.76%	7.75%	4.36%	4.47%	3.35%	3.08%

From Table 7, it is concluded that the prediction accuracy of grey model and unbiased grey model is not much different, but the unbiased grey prediction model doesn't need to carry out incremental operation for the prediction of file popularity, so the calculation efficiency will be higher. In the case of small sample, the MAPE of CS unbiased grey Markov model is the smallest compared with other models, that is, the accuracy of data prediction is higher. It shows that the model can better predict the short-term data, and provide a good method for coping with future environmental changes and taking corresponding measures in advance.

Table 8. Comparison of MAPE values of other algorithms.

Model	Model with decay factor	Time feedback model	BP neural network	Our method
MAPE	8.46%	3.48%	4.33%	3.08%

Table 8 shows the comparison of MAPE values between our method and other algorithms. In the time feedback model, the prediction is more accurate in a short period of time, but the effect becomes worse in a long period of time. The model with the decay factor has a higher prediction accuracy when the change is relatively uniform, and the prediction accuracy decreases when there is a sharp change. The MAPE value of the BP neural network model is between the time feedback model and the CS

unbiased grey Markov model. Our method achieves a relatively good effect.

5.2. Experiment 3: File copy number verification

CloudSim simulation platform is used for simulation experiment, and HDFS simulation is realized by extending the distributed file system. The prediction of file popularity uses multiple clients to write 10 files into the cluster at the same time. The number of copies obtained based on file popularity is shown in Table 9. Firstly, the CS unbiased grey Markov model is used to predict the file popularity in the future, and according to the future file access heat and the availability of the file, the change of the copy is obtained. When the file access volume increases, the number of copies needs to be increased, instead, the number of copies reduced. However, to ensure the availability of files, when the number of file copies is less than the default value, select the default value directly.

Table 9. Change in number of copies.

File	Initial file popularity	Predict file popularity at t	Initial number of copies	Number of copies at t	Copies change
1	444.01	690.09	2	3	1
2	551.56	499.9	3	2	−1
3	869.73	489.99	4	2	−2
4	467.32	1279.66	2	5	3
5	365.01	657.36	2	3	1
6	152.33	344.98	2	2	0
7	240.87	689.76	2	3	1
8	268.56	578.66	2	3	1
9	446.34	267.99	2	2	0
10	604.56	200.98	3	2	−1

The experiment shows that when the file popularity increases, such as file 4, the number of copies increases; when the access popularity decreases, the number of files decreases, such as file 3. In order to ensure the availability of file copies, when the number of file copies are less than the default value, the number of copies does not change, such as file 9.

5.3. Experiment 4: File copies adjustment efficiency verification

We verify the performance of the system after replica factor adjustment by setting different task quantities, executing tasks for 10 times and recording the average completion time with different task quantities. As is shown in Figure 3.

Through the analysis of the experimental results in Figure 3, the adjustment of the hot file replica factor has improved the system computing efficiency. The multi-factor node evaluation algorithm (MFNEA) is used to optimize the dynamic placement strategy of the new replica factor, which can promote the overall computing efficiency of the system, make full use of the computing power of the entire cluster node, and then effectively improve the work execution efficiency of MapReduce.

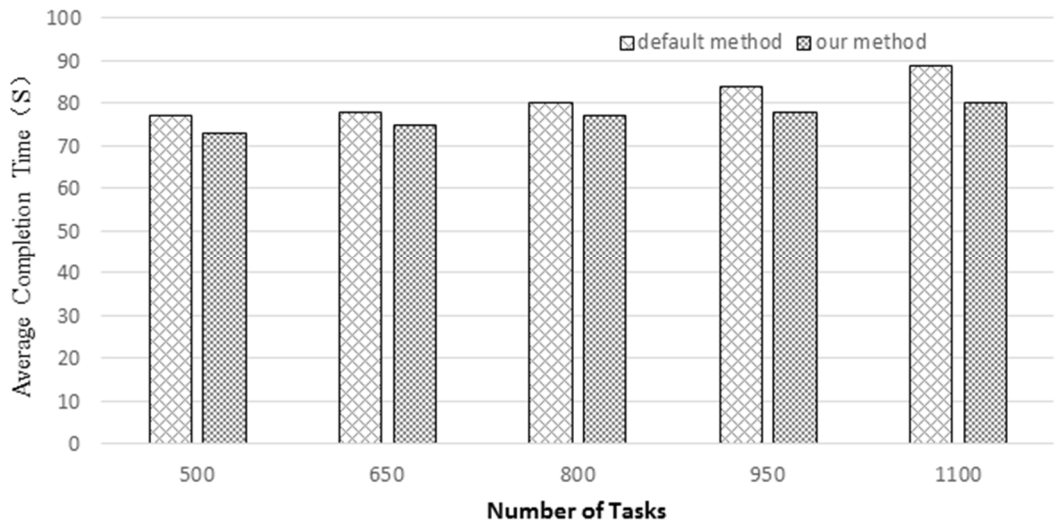

Figure 3. Comparison of average completion time under different tasks.

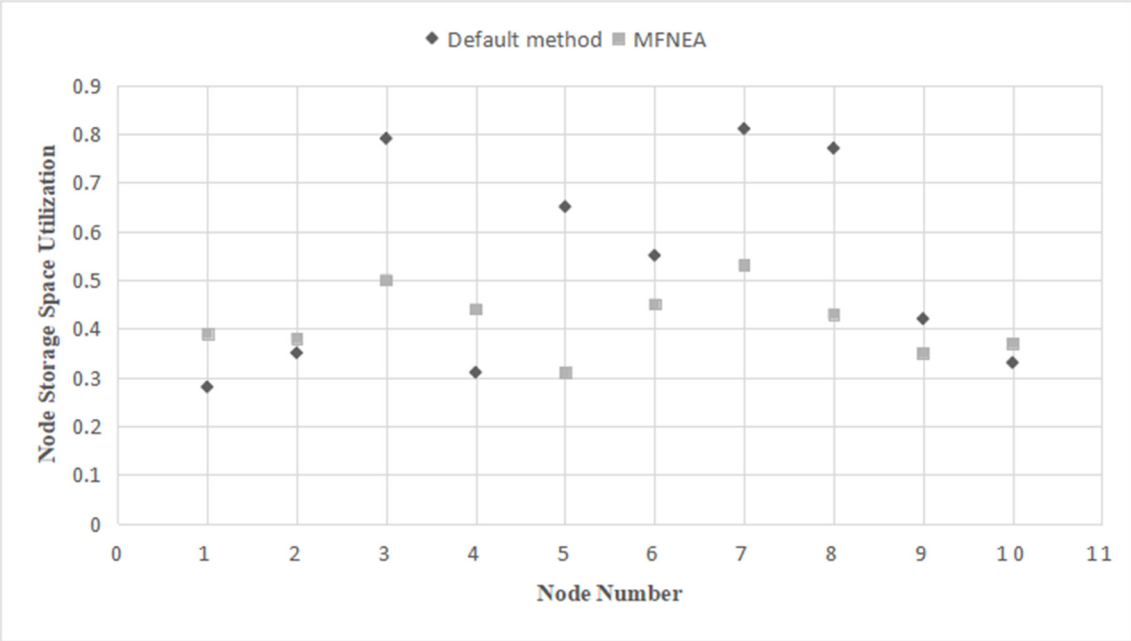

Figure 4. Storage space usage of each node.

5.4. Experiment 5: Performance test and result analysis

In the simulation experiment, 600 files were simultaneously written to the cluster by multiple clients, and the default copy placement strategy was adopted for initial writing of files. When the system runs for a period of time, the access popularity of files changes. The default copy placement strategy is compared with the copy placement strategy with MFNEA. The storage resource usage of each node in the cluster is shown in Figure 4.

The "default method" represents the default copy placement strategy in the distributed file system, and Balancer is used to adjust the copy distribution. Based on the CPU load, memory load and disk load, MFNEA determines the node performance with the method of analytic hierarchy process.

MFNEA evaluates the node in a linear weighted way, with the copy placement node selected according to node evaluation value. It can be seen from the figure that the disk space usage of each node in the default copy placement strategy varies greatly. The highest percent of node usage is 81%, while some nodes are only about 30%, which is relatively idle, causing serious imbalanced copy distribution. The MFNEA balances the storage space utilization of each node in the cluster, with the node utilization within 20%.

6. Conclusions

The number of copies will directly affect whether it can provide users with a reliable and fast data access environment in the copy management strategy. Because the file popularity changes dynamically in the system, the static copy decision cannot adapt to the changing demand of file access. The paper designs a file copy dynamic adjustment strategy. Firstly, the recent access characteristics of the file are obtained, the unbiased grey model is used to predict the file access popularity at the next moment, and the Markov model optimized by the cuckoo search algorithm is used to correct the predicted value to increase the accuracy of the predicted value. The number of copies is calculated based on file popularity and file availability, with the number of copies reduced for low-popularity files, which improves the storage space utilization in the distributed file system, and accordingly the number of copies increased for high-popularity files to improve system performance and system reliability.

Acknowledgments

This work was supported by Hebei Urban Forest Health Technology Innovation Center, Hebei Key Laboratory of Agricultural Big Data, National Natural Science Foundation of China (No. 62102130), Central Government Guides Local Science and Technology Development Fund Project (No. 226Z0201G), Precision Animal Husbandry Discipline Group Construction Project of Hebei Agricultural University (No. 1090064), Natural Science Foundation of Hebei Province (No. F2020204003), Science and Technology Research Foundation of Hebei Province (No. ZD2016158), Key Research and Development Program of Hebei Province (No. 19220119D), Hebei Youth Talents Support Project (No. BJ2019008), Research Project of Hebei Provincial Department of Human Resources and Social Security (No. JRS-2021-5024), Introducing Talent Research Project of Hebei Agricultural University (No. YJ201829).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. C. L. Abad, Y. Lu, R. H. Campbell, DARE: Adaptive data replication for efficient cluster schedule, in *2011 IEEE International Conference on Cluster Computing*, (2011), 159–168. https://doi.org/10.1109/CLUSTER.2011.26

2. M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, H. Mathkour, Least fresh first cache replacement policy for NDN-based IoT networks, *Pervasive Mobile Comput.*, **52** (2019), 60–70. https://doi.org/10.1016/j.pmcj.2018.12.002
3. H. Wu, H. Lu, F. Wu, C. W. Chen, Energy and delay optimization for cache-enabled dense small cell networks, *IEEE Trans. Veh. Technol.*, **69** (2020), 7663–7678. https://doi.org/10.1109/TVT.2020.2989033
4. Q. Li, X. Zheng, Research survey of cloud computing, *Comput. Sci.*, **38** (2011), 32–37.
5. S. P. Menon, N. P. Hegde, A survey of tools and applications in big data, in *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*, (2015), 1–7. https://doi.org/10.1109/ISCO.2015.7282364
6. Y. Taleb, S. Ibrahim, G. Antoniu, T. Cortes, Characterizing performance and energy-efficiency of the ramcloud storage system, in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 1488–1498. https://doi.org/10.1109/ICDCS.2017.51
7. L. Rao, F. Yang, X. M. Li, Dynamic replica creation algorithm based on temperature's analysis, *J. Comput. Appl.*, **34** (2014), 130–134.
8. H. Wu, H. Lu, F. Wu, C. W. Chen, Energy and delay optimization for cache-enabled dense small cell networks, *IEEE Trans. Veh. Technol.*, **69** (2020), 7663–7678. https://doi.org/10.1109/TVT.2020.2989033
9. A. M. Daniel, W. Yu, Optimization of heterogeneous coded caching, *IEEE Trans. Inf. Theory*, **66** (2020), 1893–1919. https://doi.org/10.1109/TIT.2019.2962495
10. L. Shi, M. Y. Guo, L. Liu, Y. L. Shen, L. Xu, Feedback mechanism based prediction method of dynamic replicas number, *J. Syst. Simul.*, **23** (2011), 193–199.
11. X. Xu, C. Yang, J. Shao, Data replica placement mechanism for open heterogeneous storage systems, *Procedia Comput. Sci.*, **109** (2017), 18–25. https://doi.org/10.1016/j.procs.2017.05.290
12. Z. J. Cheng, L. Wang, Y. D. Cheng, G. Chen, Q. B. Hu, H. B. Li, File access heat prediction for high energy physical hierarchical storage, *Comput. Eng.*, **47** (2021), 7.
13. Y. Qin, *Reserch on HDFS Replica Placement Management Policy and Retrieval Algorithm in Heterogeneous Storage Environment*, Ph.D thesis, University of Electronic Science and Technology of China, 2020.
14. Q. Wei, B. Veeravalli, B. Gong, L. Zeng, D. Feng, CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster, in *2010 IEEE International Conference on Cluster Computing*, (2010), 188–196. https://doi.org/10.1109/CLUSTER.2010.24
15. A. Higai, A. Takefusa, H. Nakada, M. Oguchi, A study of effective replica reconstruction schemes for the hadoop distributed file system, *IEICE Trans. Inf. Syst.*, **98** (2015), 872–882. https://doi.org/10.1587/transinf.2014EDP7242
16. L. F. Chen, D. B. Hoang, Adaptive data replicas management based on active data-centric framework in Cloud Environment, in *2013 IEEE 10th International Conference on Highperformance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, (2013), 101–108. https://doi.org/10.1109/HPCC.and.EUC.2013.24
17. S. Zhang，D. U. Qing-Wei, J. Sun, Z. Sun, Dynamic replicas strategy based on predicted popularity, *Comput. Modernization*, **2015** (2015).
18. L. L. Xu, H. Li, J. Li, Research on population prediction based on grey prediction and radial basis function network, *Comput. Sci.*, **46** (2019), 431–435.

19. Y. S. Lee, L. I. Tong, Forecasting energy consumption using a grey model improved by incorporating genetic programming, *Energy Convers. Manage.*, **52** (2011), 147–152. https://doi.org/10.1016/j.enconman.2010.06.053
20. X. S. Yang, S. Deb, Cuckoo search via Lévy flights, in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, (2009), 210–214. https://doi.org/10.1109/NABIC.2009.5393690
21. D. Chitara, K. R. Niazi, A. Swarnkar, N. Gupta, Cuckoo search optimization algorithm for designing of a multimachine power system stabilizer, *IEEE Trans. Ind. Appl.*, **54** (2018) 3056–3065. https://doi.org/10.1109/TIA.2018.2811725
22. X. C. Huang, J. W. Yin, HDFS load balancer for video cloud storage service, *J. Chin. Comput. Syst.*, **38** (2017), 293–298.

AIMS Press