Big Data Cloud Storage Optimization Scheme based on PSO Algorithm

Jie Cheng*, Songyan Liu, Wei Sheng, Yi Sun
Electronic Engineering College, Heilongjiang University, 150080 Harbin, China
*Email: xiaotainancj@163.com

Abstract. PSO algorithm, as one of the best swarm intelligence algorithms, has a strong optimization ability. It has obvious advantages to apply PSO algorithm to the optimization of resource scheduling strategy of cloud computing platform. Through the research on the application of PSO algorithm in cloud computing, effective methods are provided for the improvement of efficiency of cloud computing platform. This paper analyzes and studies the application of PSO algorithm optimized by cloud model in business scheduling of big data cloud storage platform.

1. Introduction
With the development of Internet technology and the coming of cloud era, big data technology has attracted more and more attention. Big data is usually used to describe a large number of unstructured and semi-structured data created by a company. These data will spend too much time and money to download to relational database for analysis. At present, people generally believe that cloud platform is a more efficient storage method, but due to the large amount of data, bandwidth and hardware resources, improving the storage efficiency of cloud storage system has become the most effective method [1].

Particle swarm optimization (PSO) is a kind of probabilistic global optimization algorithm. Each particle can be regarded as a search individual in the n-dimensional search space. The current position of the particle is a candidate solution of the corresponding optimization problem, and the flight process of the particle is the search process of the individual. The flight speed of the particle can be dynamically adjusted according to the historical optimal position of the particle and the historical optimal position of the population. In this paper, PSO algorithm is applied to cloud computing resource scheduling. The research on the computing efficiency of cloud computing platform provides a new idea and method.

2. Technical Background
At present, the traditional big data cloud storage scheme mainly applies SSM architecture. The whole platform is composed of three functions in the Figure 1, namely client, transport and service.

The client provides the user of the platform with the registration function directly, including the registration login and the file filter driver two small functions [2]. Users can write user name, password, and file data encryption key to log in when registering, and use the file filter driver function to realize the encryption of the local file system to be encrypted.

The transmission end of big data cloud storage platform has strong security and reliability. In order to improve the security of file data transmission, most big data cloud storage platforms adopt HTTPS
network transmission protocol. Since the number of threads in the request processing system is limited, non-blocking IO transmission is generally used to transfer data with a single thread to improve transmission performance.

The server side is usually composed of the storage cluster service and the request processing service system. As the name implies, the big data cloud storage is responsible for the storage of massive data, while the request processing system is responsible for the responsibility from each client [3].

![Figure 1. Traditional BIG data cloud storage SSM architecture.](image)

3. PSO algorithm model
Particle Swarm Optimization (PSO) is inspired by the predation of birds, which simulates particles in a Particle Swarm into birds in a flock, and particles have two properties of speed and position. Each particle in the particle swarm is a possible solution to a problem [4]. When the particle finds the optimal solution, it marks its position and shares it with other particles in the particle swarm. At the same time, other particles adjust their position and speed to follow the optimal solution.

PSO is initialized to a group of random particles (random solutions). The optimal solution is then found by iteration. In each iteration, the particle updates itself by tracking two "extreme values" (pBest, GBest). After finding these two optimal values, the particle updates its velocity and position using the following formula.

\[
v_i = v_i + c_1 \times rand() \times (pbest_i - x_i) + c_2 \times rand() \times (gbest_i - x_i)
\]

\[
x_i = x_i + v_i
\]

In formula (1) and (2), \( I = 1,2...N \), \( N \) is the total number of particles in the particle swarm; \( v_i \) is the velocity of the particle, \( rand() \) is a random number between 0-1, \( x_i \) is the current position of the particle, \( c_1 \) & \( c_2 \) is the learning factor, in general, \( c_1 = c_2 = 2 \) is ordered.

In formula (2-1), the first plus sign is called the memory term, representing the magnitude and direction of the previous velocity. The second part after the plus sign is called the self recognition term, which is the vector that points to its best advantage from its last position; The last part is called the group cognition term, which is the amount of the most recent position pointing to the best advantage of the group, reflecting the process of sharing the optimal solution among particles [5]. Through the
above process, the particle combines its own experience with that of its partner to find the best way to proceed to the next step.

\[ v_i = \omega \times v_i + c_1 \times \text{rand}() \times (p\text{best}_i - x_i) + c_2 \times \text{rand}() \times (g\text{test}_i - x_i) \]  

(3)

In Equation (3), \( \omega \) is the inertia factor, which is a constant negative value. When the value of \( \omega \) is too large, the search ability in the whole range is strong, but the search ability in the small range is weak. When the value of \( \omega \) is small, the searching ability in the whole range is weak, but the searching ability in the small range is strong [6].

4. Cloud Storage Platform Scheduling Model Based on PSO Algorithm

Based on the above discussion on the big data cloud storage scheme and combined with the advantages of big data in large-scale data storage, a cloud storage platform scheduling model is established to better solve the problems of large-scale data storage and service sharing, taking into account the advantages of particle swarm optimization algorithm in independently searching for the optimal solution in the moving space.

4.1. Abstraction and Transformation of Cloud Storage Platforms

In this paper, we compare the cloud storage platform with the idea of PSO algorithm, and compare each client that needs to store data to a flock of birds, the server that needs to store data to a food, and the information transmission medium to a flight path. The stored procedure of information is like a bird looking for food, and the algorithm is used to find the best path for the client to store and read data, thus reducing the cost. All task-resource mappings constitute a task scheduling scheme, which can be regarded as a particle in the PSO algorithm, and the set of all possible task scheduling schemes can be regarded as a population in the algorithm. Finally, an optimal solution is selected for cloud storage task scheduling, which is equivalent to the global optimal solution found when PSO algorithm converges.

4.2. Cloud Storage Platforms Store Optimal Scheduling Policies

Cloud platform scheduling model based on PSO process, as shown in the Figure 2. assumes that the virtual resource pools have \( m \) used to store computing machines, respectively to \( R = \{ r_1, r_2, r_3...r_m \} \), the user request to the cloud environment, task parsing out \( n \) tasks can be performed at the same time \( S = \{ s_1, s_2, s_3...s_n \} \), using PSO algorithm to map tasks to virtual machine resources, finally, resource detection module to adjust the scheduling policy. In order to achieve the maximum efficiency of its computing storage speed, we will look for the most reasonable scheme of task set \( S \) on computing resources in resource pool \( R \), that is, to find its optimal scheduling policy. According to the PSO algorithm used in this paper, the following scheduling strategy is proposed. In a resource search, \( t_{ij} \) represents the time spent by task \( s_i \) executing on resource \( r_j \), and \( t_{ij} = 0 \) indicates that task \( s_i \) is not scheduled to execute on resource \( r_j \). To improve the efficiency of the cloud storage platform, it is necessary to make the task set time executed on the platform as short as possible. A particle fitness function \( F \) is constructed, as shown in formula (4).

\[
F(i) = \lambda_i \times (1/\max_{j=1}^{m} \sum_{i=1}^{n} t_{ij}) + \zeta_i, 1 \leq i \leq N
\]

(4)

When PSO algorithm searches for the optimal solution, it needs to adjust the value of sum according to different situations to change the size of \( F \), so as to deal with different problems.
4.3. Implementation of Cloud Storage System Scheduling Process

First, we need to set up ETC and RCU arrays, and set up equations with the time and cost required by the task as independent variables to determine the particle trajectory. Then determine the relevant parameters according to the actual production scale, such as the upper and lower limit of search space $x_{\text{max}}$ and $x_{\text{min}}$, particle swarm size $M$, threshold value $C$, etc. Combined with the above conditions, the particle swarm is initialized, and the position of each particle is coded. During the operation, the optimal position $P_i$ is recorded, all positions are calculated, and the minimum value of particle position in the population is set as $P^*_p$. Iterate according to the particle equation and update the new position. When the sample speed exceeds the established maximum value, make $V = V_{\text{max}}/2$. As the experiment goes on, update $P^*_p$ until the algorithm converges [7].

5. Testing and Analysis

5.1. Implementation Method

This paper carries out experiments through ClouSim simulation platform, combines the physical model of the improved Particle Swarm optimization (PSO) with the resource model in CloudSim, expands the functions of CloudSim platform, and verifies the advantages and disadvantages of PSO algorithm in cloud task scheduling [8]. The simulation process of CloudSim is divided into the following steps:

- Create DataCenter DataCenter and DataCenter related resources, such as CPU, memory and bandwidth, etc.
- After resource creation, register with CIS.
- The process of information interaction managed by DataCenterBroker.

5.2. The Experimental Results

The experimental results show that, with a certain number of samples, PSO algorithm has significantly better convergence than PSO algorithm with the increase of iteration times. PSO algorithm starts to converge after about 200 iterations, while PSO algorithm has significantly faster convergence speed [9]. The number of tasks are shown in Figure 3.
Figure 3. PSO algorithm and GA algorithm convergence performance comparison.

Set the number of virtual machine resources unchanged, and the number of tasks is 20, 40, 60, 80, 100 respectively. Test and compare the effect of task number on scheduling ability under GA and PSO algorithms. The changes of total task completion time and total task cost with the number of tasks are shown in Figure 4 respectively [10].

Figure 4. PSO algorithm and GA algorithm convergence performance comparison.

Set the number of user tasks unchanged, and the number of virtual machine resources is 10, 20, 30, 40 and 50 respectively. Test and compare the influence of the number of resources on the scheduling ability under GA and PSO algorithms. According to the experiment, the changes of total task completion time and total task cost with the amount of resources are shown in Figure 5.
Figure 5. Changes of total task completion time and total task cost with the number of resources.

6. Conclusion
This paper mainly introduces the shortcomings of the traditional cloud storage scheduling algorithm, and makes improvements on this basis. Therefore, a cloud storage task scheduling method based on particle swarm optimization algorithm (PSO) is proposed. By updating the unique parameter inertia weight in iteration, the global search capability of GA algorithm is enhanced. At present, the node access cost and energy consumption of big data cloud storage technology are high, which greatly reduces the user's query and storage speed, and also increases the cost of the company. PSO algorithm can obviously improve this defect.

References
[1] Apple Inc. 2020 Patent Issued for Cloud Storage and Synchronization of Messages (USPTO 10,742,732) Computer Weekly News
[2] Pryon Inc. 2020 "Distributed Cloud Storage" in Patent Application Approval Process (USPTO 20200244735) Computer Technology Journal
[3] Han K, Yin J, Deng Z, Dong J, Tang S and Ma Z 2020 The base stations' networking scheme and spreading code optimization strategy of TC-OFDM China Satellite Navigation Conference (CSNC)
[4] Ji Y, Shao B, Chang J et al 2020 Privacy-preserving certificateless provable data possession scheme for big data storage on cloud Applied Mathematics & Computation doi: 10.1016/j.amc.2017.07.008
[5] Viswanath G and Krishna P V 2020 Hybrid encryption framework for securing big data storage in multi-cloud environment Evolutionary Intelligence doi: 10.1007/s12065-020-00404-w
[6] Saraswathi S S and Malarvizhi N 2020 Distributed deduplication with fingerprint index management model for big data storage in the cloud Evolutionary Intelligence 2020 1-8
[7] Mondal A S, Neogy S, Mukherjee N et al 2020 Performance analysis of an efficient object-based schema oriented data storage system handling health data Innovations in Systems and Software 16(2) 63-77
[8] Anand V and Pandey S 2020 New approach of GA–PSO-based clustering and routing in wireless sensor networks International Journal of Communication Systems 33(16) e4571
[9] Yu X, Bai C, Xu H et al 2020 A modified PSO assisted blind modulation format identification scheme for elastic optical networks Optics Communications 476 126280
[10] Zhang N, Pan R, Wang L et al 2020 Pattern retrieval of yarn-dyed plaid fabric based on modified interactive genetic algorithm Color Research & Application 45(6)1143-52