

# Research on Optimal Scheduling in Cloud Computing

Wenxin Feng<sup>1\*</sup>, Man Zhao<sup>2</sup>

<sup>1</sup>Henan Geology Mineral College, Zhengzhou, Henan, China

<sup>2</sup>China University of Geosciences, Wuhan, Hubei, China

\*Corresponding Author.

## Abstract:

Aiming at the problem of unbalanced load and slow convergence speed of task scheduling based on ant colony algorithm, an improved task scheduling optimization algorithm is proposed. The pheromone update rules of ant colony algorithm are optimized by giving weight to speed up the solution speed. The comprehensive performance of the algorithm is optimized by dynamically updating the volatilization coefficient, and in the updating process of local pheromone. The load weight coefficient of virtual machine is introduced to ensure the load balance of virtual machine. The experimental results show that the task scheduling strategy of the improved algorithm can not only ensure the reasonable allocation of tasks, but also improve the convergence speed and shorten the total execution time.

**Keywords:** *Ant Colony Algorithm, Task Scheduling, Load Imbalance, Convergence Speed.*

---

## I. INTRODUCTION

With the rapid development of the information age, computer technology and network technology are becoming more and more mature, and the application of Internet technology is becoming more and more extensive [1-2]. Massive data and huge task processing capacity put forward quite high requirements for hardware infrastructure, and the procurement and maintenance of hardware and other infrastructure has brought huge costs to enterprise users and individual users [3]. It has been predicted that computing will one day become a public infrastructure. At the same time, a large number of computing resources, storage resources, data resources and application resources gathered in the Internet promote the transformation of the Internet from a traditional communication platform to an intelligent computing platform, meet the needs of the above-mentioned enterprise users and individual users, reduce a large amount of infrastructure investment and save hardware maintenance costs [4-6]. As a new computing mode, cloud computing is the result of the development of parallel computing, grid computing

and distributed computing technology. Cloud computing forms the idle resources in the Internet into a unified resource pool through virtualization technology, which can be dynamically deployed and configured on demand to provide services for users [7]. This service is distributed on demand and paid on time. With the increase of the number of users, considering the dynamics, heterogeneity and complexity of nodes in cloud computing network, task scheduling has become the focus and difficulty of cloud computing research. As a business service, task scheduling in cloud computing should not only consider efficient scheduling efficiency and high resource utilization, but also consider complex service quality and low economic cost. Traditional cloud computing task scheduling algorithms have insufficient performance and can not meet complex scheduling objectives [8-10]. At present, cloud computing task scheduling focuses on the problems and challenges, which is also the starting point and purpose of this paper.

## **II. BASIC RESEARCH ON CLOUD COMPUTING TASK SCHEDULING**

### **1. Overview of cloud computing**

Based on the gradual development of distributed computing, parallel computing and grid computing, cloud computing came into being. In August 2006, Google took the lead in putting forward the new term of cloud computing, pointing out that cloud computing can be transmitted through the Internet, accessed on demand like water, electricity and gas, and charged on time. This means that computing services have become public services, public services.

Based on the cloud computing service solutions of major companies, the whole cloud computing architecture can be divided into three layers: infrastructure layer, cloud platform layer and cloud application layer. The infrastructure layer mainly includes a resource pool composed of a large number of computing resources, storage resources, network resources, data resources and virtual resources. Cloud platform layer is the core layer of cloud computing system, mainly including user management, resource management, task management and security management of cloud system. The cloud application layer mainly includes cloud software application services and interactive interfaces. Figure 1 shows different levels of cloud computing architecture:

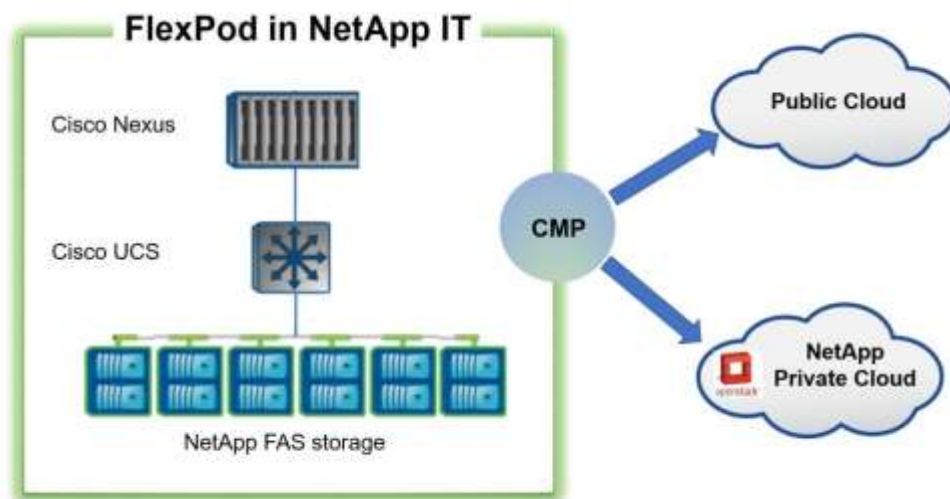


Fig 1: Cloud computing architecture diagram

As a new computing method, cloud computing involves many technologies, including data storage technology, virtualization technology, data security technology and so on.

(1) Data management technology. Cloud computing is characterized by massive data storage and massive analysis after reading. How to efficiently manage large data sets and how to find specific data in large data sets are the problems solved by cloud computing data management technology.

(2) Data storage technology. Cloud computing uses distributed storage to store data and redundant storage to ensure data reliability, that is, the same data has multiple copies. Cloud computing data storage technology also has the characteristics of high throughput and high transmission rate to meet the needs of a large number of users and provide services to users.

(3) Programming model and task scheduling model. In order to make it easier for users to use cloud computing services, the programming model of cloud computing must ensure that users can use the model to write simple programs to realize cloud computing services, and can intuitively understand the complex execution content and task scheduling running in the background.

(4) Data center energy saving. Due to the large scale of cloud computing system, it consumes a lot of energy. In order to respond to green environmental protection and reduce operation energy consumption, many scholars have done this research. Research results such as adaptive energy consumption management system, control CPU utilization, refrigeration system energy consumption optimization, multi-level data center cooling equipment and other energy-saving and emission reduction strategies have been proposed one after another.

## 2. Overview of cloud computing task scheduling

For the massive data and task processing capacity in cloud computing, data segmentation is generally used. Firstly, the huge task is divided into many smaller subtasks, and then allocated to idle resource nodes to execute in parallel. Finally, the running results are returned. The essence of task scheduling is to allocate tasks to heterogeneous available virtual resources to minimize the total task completion time, so as to achieve the task scheduling goal and make full use of resources. The essence is to find an appropriate scheduling strategy. Based on this strategy, a reasonable mapping relationship between tasks and resources is established to realize the reasonable allocation and efficient execution of tasks among resources. The purpose of task scheduling is to ensure the full utilization of resources and the efficient completion of tasks, so as to achieve a win-win situation for cloud computing providers and users.

Traditional task scheduling algorithms:

(1) Min-Min algorithm. The idea is to allocate tasks to the earliest available and fastest executing computing resources as much as possible through the strategy of easy first and then difficult, and complete the selection by calculating the minimum value of the earliest execution start time and the fastest execution speed. It belongs to a greedy algorithm. The disadvantage is that tasks are distributed to nodes with strong computing power, and nodes with low performance are not fully utilized, which is easy to cause uneven load.

(2) Max-Min algorithm. The idea is to schedule the tasks with the largest and earliest completion time to the resources with the earliest execution time according to the difficulty of task execution. The advantage is that the algorithm has certain advantages when the number of short tasks is much more than that of long tasks. The disadvantage is that it is easy to cause uneven load.

(3) Sufferage algorithm. The idea is that when a task competes, the difference between the next smallest completion time and the smallest completion time of a task is defined as the execution loss of the task, and then the task is assigned to the resource node with the largest estimated execution loss.

(4) First in first out scheduling algorithm. The idea is that tasks are allocated to resources according to the time sequence submitted by users. The disadvantage is that the system resources may be heavily occupied by the tasks submitted first, the processing time span is large, and the subsequent tasks occupy less resources, but they need to wait in line, which affects the execution efficiency and user experience of the system.

### **III. HEURISTIC ALGORITHM**

In this paper, the basic cuckoo algorithm is improved as follows, and an adaptive Cauchy cuckoo search (ACCS) algorithm based on adaptive Cauchy mutation is proposed. The basic idea of ACCs is to make use of the global variation and discrete distribution characteristics of Cauchy distribution to make Cauchy variation on a single nest to increase the diversity of nests,

which is conducive to jump out of the local extreme value for global search, and improve the search speed and quality.

When the average fitness difference of the same nest remains within the variation range in successive  $t$  generations, it indicates that the nest has fallen into a local optimal state, can not make independent progress, and needs external intervention. Because Cauchy mutation has better performance in global search, Cauchy perturbation operator is introduced to intervene the bird's nest in local optimal state.

Disturbance timing:

$$\frac{1}{T_{pre}} \sum_{t=T_{cur}-(T_{pre}-1)}^{T_{cur}} |F_t - F_{t-1}| \leq \delta \quad (1)$$

Where,  $T_{pre}$  represents the algebra of forward comparison;  $t_{cur}$  represents the current algebra;  $F_t$  represents the fitness value of the current nest of generation  $t$ ; Where  $T_{pre} < t < = T_{cur}$ ;  $\delta$  is the disturbance factor.

Variation method:

$$StepSize_d = \begin{cases} (MaxVal_d - MinVal_d)C(0,1), if rand \geq CR \\ 0, if rand < CR \end{cases} \quad (2)$$

$$X_i^{new} = X_i + StepSize \quad (3)$$

Wherein,  $StepSize_d$  represents the  $StepSize_d$  of the  $d$ -dimensional update;  $MaxVal_d$  represents the upper bound of the  $d$ -dimensional value;  $MinVal_d$  represents the lower bound of the  $d$ -dimensional value;  $x_i$  represents the  $i$ th solution,  $x_{new}$  represents the generated new solution;  $CR$  represents variation probability;  $C(0, 1)$  represents Cauchy variation formula.

Randomly generate  $n$  frogs to initialize the population  $P = \{X_1, X_2, \dots, X_n\}$ , and the  $i$ th frog in the  $S$ -dimensional solution space is expressed as  $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ . Then the frogs in the population are arranged in descending order of fitness value, and  $x_g$  represents the frog with the best fitness value in the whole population. The whole frog population is divided into  $m$  sub-populations according to certain rules, each sub-population contains a knife frog, and the condition  $N=m*n$  is satisfied. The grouping rules of frog groups are as follows: the first frog is assigned to the first sub-population, the second frog to the second sub-population, the  $m$  frog to the  $m$  sub-population, the  $m+1$  frog to the first sub-population, and so on.

$X_b$  and  $X_w$  represent frogs with the best or worst fitness in each sub population, respectively. Then, the  $X_w$  cycle in each sub population is searched locally. According to the original leapfrog rules, the update method is:

$$D_i = rand() (X_b - X_w), i = 1, 2, \dots, N \quad (4)$$

$$X_w^{new} = X_w + D_i, D_{max} \leq D_i \leq D_{max}; i = 1, 2, \dots, N \quad (5)$$

Where  $rand$  represents the random number of  $[0,1]$ , and  $D_{max}$  represents the movable maximum value of frog position. After the update, if the new frog  $x_{new}$  is better than the original

frog  $X_w$ , it will replace the original frog; If there is no improvement, replace  $X_b$  with  $X_g$  and perform leapfrog update according to formula (4) (5); If there is still no improvement, a new frog will be generated to directly replace the original  $X_w$ . Judge whether the stop conditions are met. If not, continue the local search.

After the local search is completed, the global information exchange is carried out, that is, all frogs are remixed, sorted and allocated sub populations, and then the local search is carried out repeatedly until the global convergence condition is reached.

The balance strategy between global information exchange and local depth search makes the hybrid leapfrog algorithm jump out of local extreme points and find the global optimum. The algorithm includes the idea of individual meme evolution and the idea of meme global information exchange. Hybrid leapfrog algorithm has the characteristics of simple concept, few parameters, fast calculation speed, strong global optimization ability and easy implementation. It has been successfully applied to the problems of mounting sequence optimization, traveling salesman problem, function optimization, power grid planning and so on. Although the basic leapfrog algorithm has advantages, it also has its disadvantages. The algorithm is a linear search in the local search step of the sub population, the search range is small, and the step size of each search is the same; In the leapfrog update formula, the moving step size is a random number between [0,1], which may lead to the random size of the moving step size and affect the convergence of the algorithm; when the population tends to be stable, although it has global communication, it has no self-learning ability and is easy to fall into local optimization.

#### IV. HEURISTIC TASK SCHEDULING ALGORITHM IN CLOUD COMPUTING ENVIRONMENT

Resources in cloud computing are provided to users through virtualization. Generally, any resource VM in cloud computing can be described as:

$$VM = \{vmid, mips, size, ram, be, pesNumber\} \quad (6)$$

VMID represents the resource ID number, MIPS represents the number of CPU running instructions, size represents the resource size, RAM represents memory, BW represents bandwidth, and pesnumber represents the number of CPUs owned.

Any task CLET submitted by the user can be described as:

$$Clet = \{id, length, fileSize, outputSize\} \quad (7)$$

Where id represents the task ID, length represents the task length, and filesize and outputsize represent the input and output file sizes of the task CLET respectively. Filesize and outputsize respectively represent the input and output file sizes of the task CLET.

The cloud computing task scheduling problem in this paper can be described as: in the cloud computing system, n independent tasks are allocated to M virtual resources for execution

according to a certain scheduling strategy,  $n > m$ , and the allocation relationship can be expressed as:

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{1n} \\ x_{21} & x_{22} & x_{2n} \\ x_{m1} & x_{m2} & x_{mn} \end{pmatrix} \quad (8)$$

Resource scheduling scheme belongs to combinatorial optimization problem. To solve the scheduling problem by algorithm, we first need to construct a reasonable coding method to represent the solution of the scheduling problem. In cloud task scheduling, if there are several tasks that need to be executed on M virtual machines, it represents a unary array solved as knife dimension, and the virtual machine number corresponding to each task is the value of the current dimension. When encoding the virtual machine number, the integer encoding from 1 to m is adopted. Since the real number encoding method is used in the iteration of ACCs and dlsfla algorithms, formula (9) is used for encoding and formula (10) is used for decoding:

$$X = [x_1, x_2, \dots, x_n], x_i \in [1, m + 1] \quad (9)$$

$$X' = [\lfloor x_1 \rfloor, \lfloor x_2 \rfloor, \dots, \lfloor x_m \rfloor] \quad (10)$$

$x_i$  is a real number and  $x_i \in [1, m + 1]$ ,  $\lfloor x_i \rfloor$  means rounding down  $x_i$ . After decoding, the value on each dimension represents the execution resource serial number allocated by the corresponding task, that is, the virtual machine number.

ACCs based task scheduling steps:

- (1) Initialize n bird nests, i.e. n scheduling schemes;
- (2) The solution code of each bird's nest is used as the solution in the scheduling scheme to judge whether variation is required. If necessary, Cauchy variation is carried out on the solution, and the value of the objective function is calculated as the fitness of the solution;
- (3) According to Levi's flight update formula, the solution of the scheduling scheme is updated to generate n new solutions, judge whether the newly generated solution meets the constraints, and retain the original optimal solution;
- (4) Generating a new solution from the discovered solution according to the discovered probability  $P_a$ ;
- (5) Calculate and evaluate the fitness corresponding to each solution (the fitness corresponding to each bird's Nest), that is, calculate and evaluate the fitness of each scheduling scheme to find out the solution corresponding to the minimum fitness. Update the optimal solution set;
- (6) Judge whether the termination condition is satisfied, that is, the number of iterations. If it is satisfied, the optimal solution is output. If not, return to step (2) and continue to execute until the termination condition is satisfied.

## V. CONCLUSION

Based on the analysis of the research status of cloud computing and cloud computing task scheduling, this paper introduces the concept, characteristics and objectives of cloud computing task scheduling. The advantages and disadvantages of traditional task scheduling algorithm and heuristic task scheduling algorithm are analyzed. The basic cuckoo search algorithm and hybrid frog leaping algorithm, which are less studied, are selected from the heuristic algorithms. The basic ideas and algorithm processes of the two algorithms are analyzed. They are improved according to their respective shortcomings. Five standard test function experiments are carried out on the improved algorithm. In short, the two task scheduling algorithms proposed in this paper need to be further improved.

## REFERENCES

1. R Yu, I Cheng, B Zhu, S Bedmutha, A Basu. Adaptive Resolution Optimization and Tracklet Reliability Assessment for Efficient Multi-Object Tracking. *IEEE Transactions on Circuits & Systems for Video Technology*. 2017;PP(99):1-1.
2. F Meyer, P Braca, P Willett, F Hlawatsch. A Scalable Algorithm for Tracking an Unknown Number of Targets Using Multiple Sensors. *IEEE Transactions on Signal Processing*.2017; 65(13):3478-3493.
3. S Subedi, YD Zhang, MG Amin, B Himed. Group Sparsity Based Multi-Target Tracking in Passive Multi-Static Radar Systems Using Doppler-Only Measurements. *IEEE Transactions on Signal Processing*.2016; 64(14):3619-3634.
4. JQ Li, RH Zhao, JL Chen, CY Zhao, YP Zhu. Target tracking algorithm based on adaptive strong tracking particle filter. *Iet Science Measurement & Technology*.2016; 10(7):704-710.
5. Y Chen, Q Zhao, Z An, P Lv, L Zhao. Distributed Multi-Target Tracking Based on the K-MTSCF Algorithm in Camera Networks. *IEEE Sensors Journal*.2016; 16(13):5481-5490.
6. H Zhang, H Ge, J Yang, Y Yuan. A GM-PHD algorithm for multiple target tracking based on false alarm detection with irregular window. *Signal Processing*.2016;120(C):537-552.
7. Sun B, Jiang C, Li M. Fuzzy Neural Network-Based Interacting Multiple Model for Multi-Node Target Tracking Algorithm. *Sensors*.2016; 16(11):1823.
8. Bhuiyan M Z A, Wang G, Vasilakos A V. Local Area Prediction-Based Mobile Target Tracking in Wireless Sensor Networks. *Computers IEEE Transactions on*.2015; 64(7):1968-1982.
9. M Beard, S Reuter, K Granström, BT Vo, BN Vo, A Scheel. Multiple Extended Target Tracking With Labeled Random Finite Sets. *IEEE Transactions on Signal Processing*.2016; 64(7):1638-1653.
10. L Wang, T Liu, G Wang, KL Chan, Q Yang. Video Tracking Using Learned Hierarchical Features. *IEEE Transactions on Image Processing*.2015; 24(4):1424-1435.