

Research on cloud computing task scheduling based on calculus mathematical equation

Xinying Zou

No.6 high school of Xiamen, Fujian, China

Zouxinying0121@163.com

Abstract. Due to the heterogeneity, distribution, autonomy and diversity of services in cloud computing environment, higher requirements are put forward for cloud platform scheduling mechanism, so the research on cloud architecture and its scheduling mechanism has attracted more and more attention from the industry. A cloud computing task scheduling algorithm based on calculus mathematical equation is proposed. Through the double boundary convergence control of the partial differential classification mathematical model, the partial differential classification data model is integrated into the data set, and the fuzzy control of the data is completed through the increment and decrement support vector. The membership function is used to transform the multi-QoS(quality of service) objective constraint problem into a single objective constraint solving problem. Compared with traditional methods, the method proposed in this paper can effectively reduce the deadline baseline violation rate of user task scheduling, and reduce its average task execution time and average task execution cost on the premise of meeting the user task multi-QoS target constraints.

Keywords: Calculus; Cloud computing; task scheduling.

1. Introduction

In the traditional parallel distributed computing, the computing resources are isomorphic, and the resources are usually concentrated geographically, so there is no need to consider resource attributes and communication overhead too much. In cloud computing, the computing resources are widely distributed, heterogeneous, and the load changes dynamically. Therefore, the task scheduling problem in cloud computing environment is much more complicated than that in traditional parallel distributed environment. The quality of service provided by cloud computing is mainly affected by task scheduling algorithm [1]. In the task scheduling of cloud computing, there are strong dependencies among many tasks, and users' demands for task scheduling time, cost and service quality are becoming more and more complicated. How to schedule tasks and allocate resources reasonably and effectively has become the bottleneck of the development of cloud computing. The quality of task scheduling algorithms directly affects the quality of services provided by cloud computing.

Due to the huge server scale, heterogeneous resources, extensive user groups, different types of application tasks and different requirements of QoS(quality of service) objectives and constraints, the cloud computing system always has to deal with a large number of user tasks and massive data [2]. In this context, how to allocate and manage the resources in the cloud system reasonably has become a research hotspot and technical difficulty in the field of cloud computing [3]. Therefore, under the service-based business computing mode of cloud computing, it is not only of high theoretical value, but also of good practical significance to study its task scheduling strategy in depth.

2. Task scheduling in cloud environment

Traditional task scheduling is mainly divided into independent task scheduling and workflow task scheduling. Independent task scheduling is relatively simple because the tasks are relatively independent and have no sequence or dependency, and the tasks can be executed in parallel without any order in the execution process. Task scheduling in cloud computing environment usually divides a large task set into several tasks, then assigns these tasks to appropriate computing resources for processing according to the specific requirements of users, and returns the processing results to users.

The process of task scheduling is complicated. At the beginning of scheduling, the task set is mainly split and decomposed, a task set is divided into several tasks, and computing resources are allocated to the tasks, which is the task scheduling stage. After calculating the tasks with assigned resources, execute the tasks, get the execution results of the tasks, and return the execution results to the users. This is the task execution stage. No matter what kind of tasks are executed, the main goal is to meet the needs of users, which is determined by the service-oriented business model of cloud computing. Cloud task scheduling is shown in Figure 1.

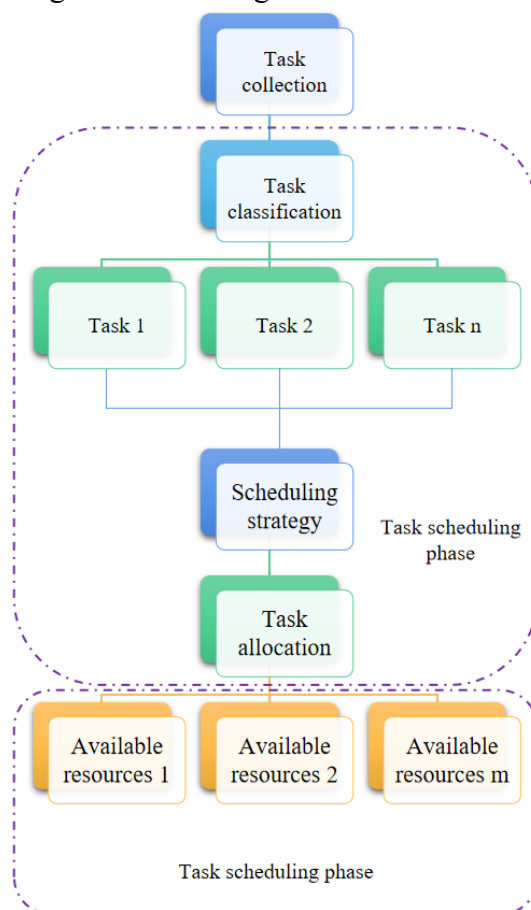


Figure 1. Schematic diagram of task scheduling

As can be seen from the schematic diagram of task scheduling in Figure 1 above, the task set will be divided into different blocks in the cloud computing environment, which will be allocated to the resource processing in the cloud environment to achieve the effect of distributed computing. MapReduce mode is to split the tasks to be processed and executed into Map and Reduce for distributed processing. The task model in cloud computing environment is similar to MapReduce mode, which consists of several Map tasks and several Reduce tasks.

3. Partial calculus classification mathematical model

At present, the principle of partial calculus is widely used in the field of associated data mining, which can improve the high frequency area of associated data, dynamically store the low frequency area of data, and increase the interference factors of data. However, when the principle of partial calculus promotes the low frequency region of data, the lowest frequency region of data is required to select the order [4]. In this paper, the principle of partial calculus is adopted in the process of association data mining, and the model of association data mining is shaped, so as to realize the differential data fusion based on the principle of partial calculus and improve the efficiency of association data mining.

Partial calculus equation is produced by the transformation of integer order partial differential equation, and partial derivative is obtained by replacing the partial derivative term of function influencing factor in integer order differential equation [5]. Partial calculus equation is:

$$\frac{\partial^\alpha S(x, y, t)}{\partial^\alpha t} = \frac{\partial^\alpha S(x, y)}{\partial^\beta x} + \frac{\partial^\beta S(x, y)}{\partial^\beta y} \quad (1)$$

In which: partial differential order of α, β table; x, y in the function $S(x, y, t)$ is the influence factor.

Described by Q' , and the group S_k in Q' is not an empty set, we can obtain the double-boundary stability of partial calculus equation with time delay under the double-boundary environment, which satisfies the following conditions:

$$\alpha^T Q \alpha = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j Q_{ij} \geq 0 \quad (2)$$

Under Lipschitz's convex condition, in order to keep the value of formula (2) stable at the initial stage of calculation, the edge integral term of the differential equation updated in the spatial matrix Q' is reversible. Assuming that one set of data sample α_c is not an empty set, a semi-positive definite eigendecomposition matrix is obtained:

$$Q' \times R' = Q' R_{II} - \frac{Q'}{R_{II}} (R_{*I} \times R_{*I})_{II} = Q' \times R_{II} + (R_{*I} \times R_{*I})_{II} = I \quad (3)$$

The following is a proof of the stability of the double boundary convergence control of the partial calculus classification mathematical model.

Because of the convergence of random universal function, there is an n -order square matrix in the finite field $Y^H(q)A(0)$, which conforms to the convergence $\dot{V}(t) \leq \xi^T(t) \Psi(d_1(t), d_2(t)) \xi(t) < 0$ condition of Bochner- Riese, and $\Psi(d_1(t), d_2(t)) < 0$ appears, and it is obtained that:

All matrices with dimensions in the normal range have boundary equilibrium points of partial calculus classification mathematical model. If one of λ_1, λ_2 is greater than 0, $\sum_1 + \sum_3^T \sum_2^{-1} \sum_3 < 0$ holds, and only if:

$$\begin{bmatrix} \sum_1 & \sum_3^T \\ \sum_3 & -\sum_2 \end{bmatrix} < 0 \text{ or } \begin{bmatrix} -\sum_2 & \sum_3 \\ \sum_3 & \sum_1 \end{bmatrix} < 0 \quad (4)$$

When it appears, there are two continuous equilibrium points in the mathematical model of partial differential classification, and it is concluded that the double boundaries of the association mining process have convergence, and the convergence is relatively stable.

4. Cloud computing task scheduling

4.1 Mathematical model of grid task scheduling problem

Grid task scheduling is a NP-hard problem, and the heuristic search algorithm can get the global optimal solution in a given problem domain. To simplify the problem space, only two key factors, resource processing speed and task length, are considered. The scope of this paper belongs to static task scheduling, that is, computing resources, task information and dependencies among tasks in grid are known in advance, and applications submitted to grid have been decomposed into multiple tasks, and tasks are defined as jobs submitted to grid.

Suppose there is a grid system G , G is defined by triple like $G = \langle T, R, ETC \rangle$, which is expressed as follows:

T represents the user's application in grid system G , which contains n independent subtasks, and there is no data communication among each subtask, as shown in formula (5).

$$T = \{t_1, t_2, \dots, t_n\}, t_i \cap t_j = \emptyset \quad (5)$$

R represents computing resources in grid system G , and R contains m computing nodes, as shown in formula (6).

$$R = \{r_1, r_2, \dots, r_m\} \quad (6)$$

The expected execution time of the task $t_i (1 \leq i \leq n)$ assigned to the available node $r_j (1 \leq j \leq m)$ is $ETC(t_i, r_j)$, and the matrix formed by all assignments of T to R is called ETC matrix, as shown in formula (7).

$$ETC_{n \times m} = ETC(t_i, r_j) 1 \leq i \leq n, 1 \leq j \leq m \quad (7)$$

Due to the heterogeneity of tasks and computing resources, n tasks are scheduled on m hosts, and the execution time of a task on different machines is different, so n tasks and m hosts form a new execution time matrix ETC . With $start(r_j)$ representing the start time when task t_i assigns computing resource r_j to execute, and $end(t_i, r_j)$ representing the completion time of task t_i on computing resource r_j , formula (8) is given.

$$end(t_i, r_j) = start(r_j) + ETC(t_i, r_j) \quad (8)$$

Assuming that there is no data communication between tasks, the tasks will not be interrupted when they begin to execute, and the tasks have no priority and are static, that is, users will not interact after submitting the tasks to the grid system. Let $total(r_j)$ be the total time of computing resource M_j after all computing scheduling is completed, and the goal of grid task scheduling is shown in Formula (9).

$$\min \sum_{j=1}^m total(r_j) \quad (9)$$

4.2 Collaborative scheduling of cloud computing tasks with multiple QoS constraints

When scheduling tasks in the cloud computing environment, the application tasks participating in the scheduling usually have various QoS target constraints, and they hope to obtain better QoS guarantee from the remote resources of the cloud computing system to ensure that their QoS target constraints are met [6]. At the same time, cloud computing systems can often provide multi-level quality of service guarantee for application tasks. Therefore, in the cloud computing environment, the task scheduling strategy should generally consider the QoS target constraint requirements of scheduling tasks and the performance parameters of various available resources, so as to achieve a reasonable match between a large number of application tasks and available computing resources [7].

Generally, the relevant attribute parameter information of available resources and application tasks and QoS target constraints can be used to assist in making and optimizing scheduling decisions. Moreover, when a certain task scheduling strategy is applied for resource reservation, the allocation and deployment of computing resources can be planned in advance based on the above parameter information and constraints [8-9]. At present, many task scheduling strategies can predict the running time of application tasks to be scheduled by modeling, and then realize the reservation and planning of available computing resources in advance.

The QoS target constraint on the deadline baseline can be described by formula (10) [10]:

$$\text{Min}(O_{\text{deadline}}) = \sum_{i=1}^T \sum_{j=1}^N M_{ij} v_{ij} \text{ s.t. } \sum_{i=1}^T \sum_{j=1}^N M_{ij} R(t_i, R_k) \leq \sum_{j=1}^N \sum_{k=1}^K S(CN_j, R_k) \quad (10)$$

While getting satisfactory service, users always want to save their scheduling expenses as much as possible. Therefore, the cloud system should complete the scheduling and execution of all tasks as far as possible under the constraints of the given scheduling budget target.

According to the above analysis, the above QoS target constraint condition on the application task scheduling budget can be expressed in formula (11):

$$\text{Min}(O_{\text{budget}}) = \sum_{i=1}^T \sum_{j=1}^N M_{ij} c_{ij} \text{ s.t. } \sum_{j=1}^N M_{ij} c_{ij} \leq B_i, \forall i \in T \quad (11)$$

Fig. 2 shows the architecture of the M-QoS-OCCSM collaborative scheduling model proposed in this chapter.

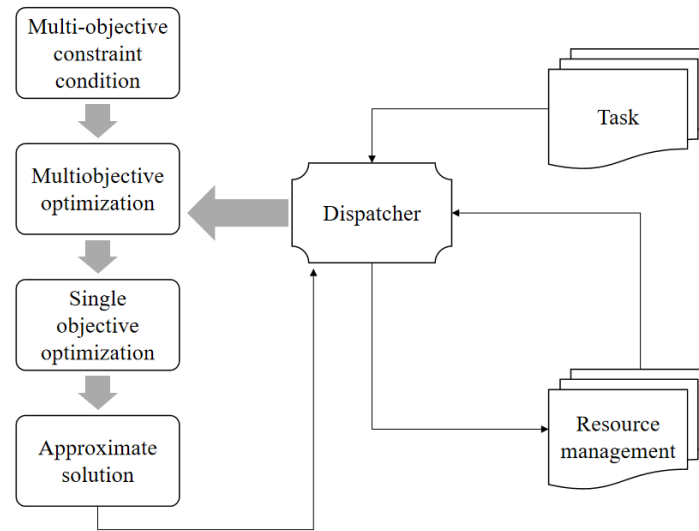


Figure 2. Cloud task collaborative scheduling architecture with QoS target constraints

As shown in fig. 2, the execution flow of cloud task collaborative scheduling with QoS target constraints is as follows: firstly, the user submits the application task with deadline baseline and scheduling budget target constraints to the task scheduler; Then, the task scheduler sends the QoS target constraint condition and other related parameter information of the application task to the extension module of the cloud task collaborative scheduling model with QoS target constraint. The obtained result is the finally obtained scheduling decision scheme, which is returned to the scheduler, and the scheduler selects the appropriate computing resources for the application tasks to be scheduled based on the optimization result, thus completing the scheduling decision.

The optimization problem with multi-objective QoS constraints is transformed into an optimization problem with single-objective constraints. The reconstructed genetic algorithm is applied to solve the above single objective constrained optimization problem. The solution process is stated as follows.

First, the values of crossover probability p_c , mutation probability p_m and chromosome population scale r need to be determined, and then the initial population $P(0)$ of chromosomes is randomly generated. The value range of chromosome loci is an integer in the interval $[1, m]$, the population algebra is expressed by, and the initial population algebra is 0.

Two chromosome individuals (P_i, P_j) are randomly selected from the current population $P(g)$, and then based on the crossover probability p_c , modular operation is applied to perform crossover

operation on the selected two chromosome individuals. The new chromosome set generated by crossover operation is denoted as O_c .

With the mutation probability p_m , a unified mutation operation is performed on the new chromosome set O_c generated by the crossover operation. The new chromosome set generated by mutation operation is denoted as O_m .

Merging the current population and the new chromosome set generated by crossover and mutation operation to obtain a set $P(g) \cup O_c \cup O_m$, then applying the elite retention strategy, selecting R best chromosome individuals from the set $P(g) \cup O_c \cup O_m$ as the next generation population, and keeping them as $P(g+1)$, and making $g=g+1$.

If the termination conditions of genetic algorithm are met, such as the number of iterations exceeds the preset maximum number of iterations or the optimization results meet the user's requirements, the iterative operation is terminated; Otherwise, if the iteration times of the algorithm have not reached the preset maximum value, and the current optimization result of the algorithm has evolved obviously.

5. Experimental analysis

5.1 Scalability test

To evaluate the expansion performance of the experimental method, a large number of nodes are needed to participate in the calculation to judge whether the method can accurately process huge data. The experiment enlarges the data at the same scale, and the number of nodes ranges from 1 to 8, and tests the extension performance of this method to Trajectory, Habermans and Iris data sets. The results are described in Figure 3.

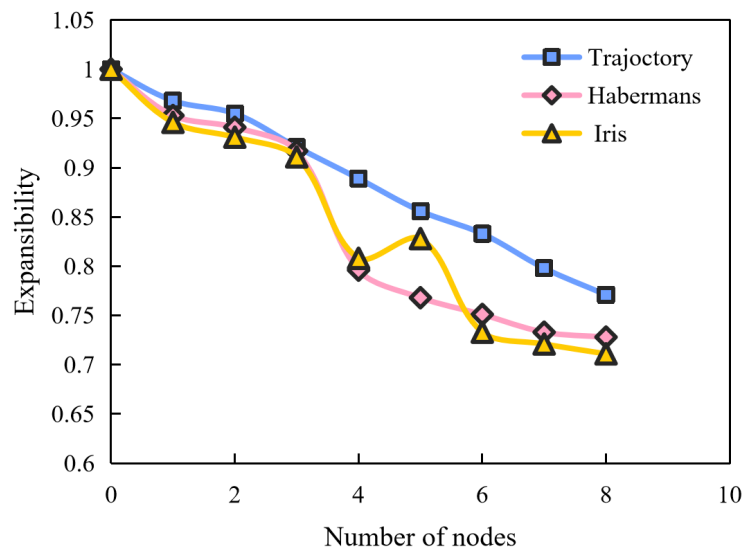


Figure 3. Extension effect of this method in different data sets

By analyzing Figure 3, it can be concluded that the scalability value of this method is lower than 1, and its scalability value is always greater than 0.06, because the number of nodes and the size of data sets are increasing in the same proportion. Compared with other two methods, this method has better scalability, which is 0.71 and 0.66 higher than Par2PK - means method and traditional undersampling method, respectively, which shows that the improved association mining method of partial calculus classification mathematical model designed in this paper has higher scalability and adaptability in the process of mining association data of big data.

5.2 Comparison of algorithm convergence

In order to verify the performance of this algorithm, two groups of comparative experiments were carried out with reference [8-10]. Using simulation software to create 1 024 grid computing resources and 600 tasks. Compare the convergence rates of the four algorithms under the same amount of tasks, as shown in Figure 4.

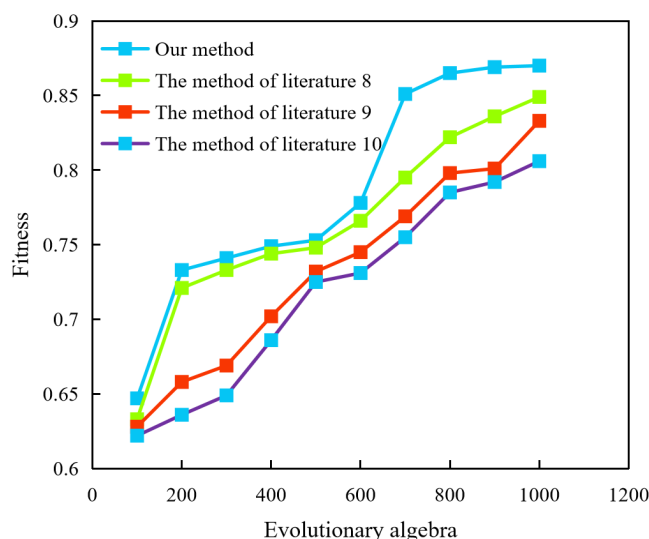


Figure 4. Comparison of algorithm convergence

It can be seen from fig. 4 that the algorithm in this paper starts to converge after 700 generations of evolution, while the algorithms in literature [8-10] all start to converge after 850 generations of evolution, which shows that the convergence speed of the algorithm in this paper is better than that in literature. In terms of fitness, the average fitness of the proposed algorithm is consistently better than that of the literature algorithms in the evolution process, which indicates that the proposed algorithm effectively simulates the diversity maintenance and antibody response mechanism of the immune system, and has high local optimization ability. In a word, the grid task scheduling result of this algorithm is better than that of the algorithm in literature [8-10].

5.3 Average scheduling overhead cost test results

Figure 5 shows the comparison of the average task scheduling overhead cost of each of the three algorithms when different numbers of application tasks participate in the scheduling. Scheduling overhead cost refers to the actual scheduling overhead cost paid by users to complete the scheduling execution of a task in the target computing environment.

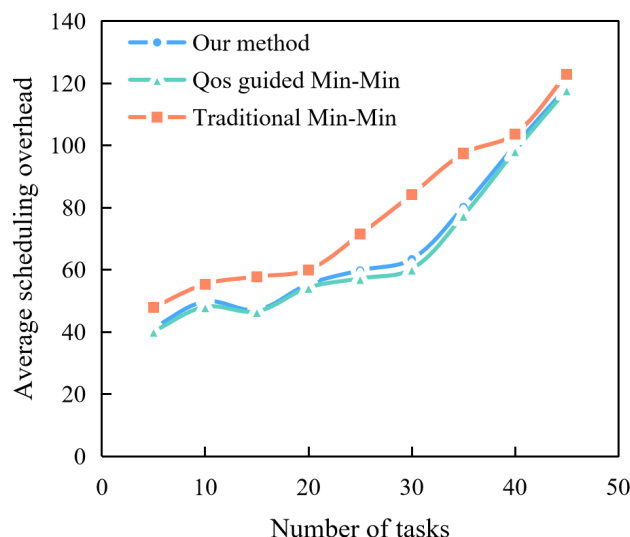


Figure 5. Comparison of average scheduling costs of three algorithms under different task quantities

Looking at fig. 5, it can be seen that the average task scheduling cost of the proposed method and the improved QoS-oriented Min-Min algorithm is obviously superior to the traditional Min-Min scheduling algorithm. Due to the limited computing power of available resources, the resource competition is intensified. Compared with the QoS-oriented Min-Min algorithm, the QoS-constrained cloud task collaborative scheduling strategy needs to consider meeting the deadline baseline constraints of scheduling tasks when making scheduling decisions. Therefore, it is inevitable that a small number of key tasks will be assigned to computing nodes with relatively high running costs for scheduling and execution, which makes the average scheduling overhead cost of the algorithm increase to some extent.

In addition, with the increase of the number of application tasks participating in scheduling, some application tasks will inevitably be assigned to computing nodes with high running cost for scheduling and execution, so the average task scheduling overhead cost of the three strategies will increase with the increase of the number of tasks participating in scheduling.

6. Conclusion

Task scheduling strategy is an important part of cloud computing technology. Because different users' application task scheduling requests have different requirements for QoS, and cloud resources are dynamic and changeable, it has become a research hot spot in the field of cloud computing to schedule application task scheduling requests reasonably and efficiently. In this paper, a cloud task collaborative scheduling model with multiple QoS constraints is proposed based on calculus mathematical equations. Test results show that, in the same cloud computing simulation environment, the QoS goal-constrained cloud task collaborative scheduling strategy is superior to the Min-Min algorithm and the improved QoS-oriented Min-Min algorithm in terms of average task completion time, task deadline baseline violation rate and average task scheduling overhead cost.

References

- [1] Ma T, Pang S, Zhang W, et al. Virtual Machine Based on Genetic Algorithm Used in Time and Power Oriented Cloud Computing Task Scheduling [J]. *Intelligent automation and soft computing*, 2019, 25(3):603-611.
- [2] Pang S, Li W, He H, et al. An EDA-GA Hybrid Algorithm for Multi-objective Task Scheduling in Cloud Computing [J]. *IEEE Access*, 2019, PP(99):1-1.
- [3] Mukherjee P, Pattnaik P K, Swain T, et al. Task scheduling algorithm based on multi criteria decision making method for cloud computing environment: TSABMCDMCCE [J]. *Open Computer Science*, 2019, 9(1):279-291.
- [4] Fan L, Dong M, Jing C. An Efficient Task Scheduling Algorithm Based on Particle Swarm Optimization with Self-Learning Strategy and Neighbor Heuristic Mechanism on the Cloud [J]. *Computer journal*, 2020, 31(2):180-196.
- [5] Hung P P, Alam G, Hai N, et al. A Dynamic Scheduling Method for Collaborated Cloud with Thick Clients [J]. *The international arab journal of information technology*, 2019, 16(4):633-643.
- [6] Arunarani A, Manjula D, Sugumaran V. Task scheduling techniques in cloud computing: A literature survey[J]. *Future Generation Computer Systems*, 2019, 91:407-415.
- [7] Bezdan T, Zivkovic M, Bacanin N, et al. Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm[J]. *Journal of Intelligent and Fuzzy Systems*, 2021(3):1-13.
- [8] Guo X. Multi-objective task scheduling optimization in cloud computing based on fuzzy self-defense algorithm [J]. *AEJ - Alexandria Engineering Journal*, 2021, 60(6):5603-5609.
- [9] Huang X, Li C, Chen H, et al. Task scheduling in cloud computing using particle swarm optimization with time varying inertia weight strategies [J]. *Cluster Computing*, 2020, 23(2):1137-1147.

- [10] Agarwal M, Srivastava G. Opposition-based learning inspired particle swarm optimization (OPSO) scheme for task scheduling problem in cloud computing [J]. Journal of Ambient Intelligence and Humanized Computing, 2021(12):1-21.