

Research and Simulation of Task Scheduling Strategy in Cloud Computing

Tao Lin^{1,2}, Qianqian Xuan^{1,*}, Qingguo Xu¹ and Mengxian Wu¹

¹Institute of Control Science and Engineering, Hebei University of Technology, Tianjin 300130, China

²Institute of Computer Science and Software, Hebei University of Technology, Tianjin 300401, China

*Corresponding author

Abstract—How to make full use of resources in the cloud to efficiently schedule tasks is an important issue in the field of cloud computing. This paper proposes a cloud computing task scheduling algorithm based on optimized Differential Evolution Algorithm (DE), which introduces the initial population model based on the Weighted Round-Robin Scheduling Algorithm, and makes the two control parameters of the algorithm adaptively adjust with the iteration number increase. When the task scheduling model is established, considering the user needs and the interests of the cloud provider, the execution time, the execution cost, the resource utilization rate and the execution energy consumption are taken as the scheduling targets. The optimal differential evolution algorithm is used to solve the task model, and an optimal scheduling scheme is obtained. The simulation results show that the improved algorithm can improve the scheduling performance and the effect is better.

Keywords—component; cloud computing; task scheduling; differential evolution algorithm; weighted round-robin algorithm

I. INTRODUCTION

Cloud computing is a new computing model developed on the basis of distributed computing, parallel computing and grid computing. It has been paid more and more attention since its appearance^[1]. In the cloud computing system, the scale of the server is huge, the resources are heterogeneous and diverse, the user community is wide, the application task type is different, the quality of service (QoS) target constraint requirement is different. How to timely and efficiently carry out the task scheduling, rational use of resources, improve the efficiency of resource utilization and task execution, become one of the core issues of cloud computing researching. Therefore, task scheduling problem is a hot topic in the field of cloud computing.

In recent years, some artificial intelligence algorithms have been widely concerned in the application of scheduling problems, among them, Differential Evolution Algorithm is used to solve the problem of cloud computing task scheduling, which is based on the simple process and less controlled parameters. In [2], a hybrid scheduling algorithm based on genetic algorithm and ant colony algorithm is proposed to solve the multi-QoS constraints. Task scheduling model considers execution time, cost, security and reliability. The algorithm has good performance in resource utilization and QoS guarantee, but the time complexity of scheduling algorithm is too high. In [3], the cuckoo search algorithm and the Pareto optimal model

are used to complete the multi-objective optimization task scheduling and achieve the win-win of efficiency and energy consumption. The cuckoo search algorithm is a continuous algorithm, and the cuckoo search algorithm is only suitable for solving single-objective optimization problems, and the load balance is not considered in the paper.

In view of the above problems, a Self-Adaptive Differential Evolution Algorithm (IADE) scheduling algorithm is proposed based on the optimization of the adaptive differential evolution, and the model which contains the execution time, cost, resource utilization and energy consumption is established. The experimental results verify the effectiveness of the proposed scheduling strategy.

II. MODEL OF TASK SCHEDULING

A. Task Scheduling Problem Description

Task Scheduling is designed to establish a reasonable mapping relation between application task set and available computing resource set according to some constraint condition, and to allocate suitable computing resources for application tasks participating in the scheduling task to complete the application task of system user scheduling request.

The set $P = \{p_1, p_2, \dots, p_n\}$ is used to denote the set of compute resource nodes contained in the cloud computing system, where n is the number of computing resources nodes; The set $T = \{t_1, t_2, \dots, t_m\}$ represents the set of application tasks to be scheduled, where m represents the number of application tasks to be scheduled. In general, the number of application tasks involved in scheduling is much larger than the number of available computing resources, $m \gg n$. A five-element array $\sum: (T, C, X, TIME, E)$ is used to describe the task scheduling problem in the cloud environment, where T is the set of application tasks to be scheduled. C is the execute cost matrix of $m * n$, where the element $c_{i,j}$ indicates the execution cost required for the application task t_i to be assigned to the computing node p_j for scheduling execution; X is the task allocation matrix of $m * n$, where the element $x_{i,j} = 1$

represents the application task t_i being assigned to the compute node P_j for scheduling execution, otherwise $x_{i,j} = 0$; $TIME$ is the execute time matrix of $m * n$, where the element $time_{ij}$ indicates that the application task t_i is assigned to the computing node P_j for scheduling execution time; E is the executed energy consumption matrix of $m * n$, where the element e_{ij} indicates that the application task t_i is assigned to the computing node P_j for scheduling execution energy consumption.

B. Task Scheduling Model

In this paper, cloud computing task scheduling satisfies the following conditions: a) tasks to be scheduled are independent tasks; b) the processor is heterogeneous, and have different processing power; c) task completion time includes task execution time.

The execution time of the scheduled task on the virtual resource node P_j see(1):

$$ETC_j = \sum_{i=1}^m x_{ij} * time_{ij} \quad (1)$$

$$time_{ij} = \frac{TaskLength}{EV_j} \quad (2)$$

$TaskLength$ is MIP, which is the Unit of measurement for task t_i , EV_j is the computing power of virtual machine, solution(3):

$$EV_j = cpunum_j \times cpuMIPS_j \quad (3)$$

$cpunum_j$ is the number of processors for the virtual machine node P_j , $cpuMIPS_j$ is virtual machine node P_j processing speed of each processor.

Since the tasks are running in parallel on the virtual machine, all tasks are completed:

$$TotalTime = \max_{j=1}^n ETC_j \quad (4)$$

In this chapter, only the execution energy consumption is considered, regardless of the energy consumption of the transmission. The total execution energy consumed by each task scheduling scheme for all tasks is:

$$EC = \sum_{i=1}^m \sum_{j=1}^n x_{ij} * e_{ij} \quad (5)$$

The higher the resource utilization, the energy consumption becomes lower. In addition, the higher the utilization of resources, cloud service providers will get more profits. The resource utilization is defined as follows:

$$RA = 1 - \sum_{j=1}^n (TotalTime - ETC_j) / (m * TotalTime) \quad (6)$$

The total execution cost required to complete all tasks:

$$COST = \sum_{i=1}^m \sum_{j=1}^n (c_{ij} \times x_{ij}) \quad (7)$$

Cloud independent task scheduling model can be abstracted into a multi-objective optimization problem, and the scheduling targets chosen in this paper are determined by analysis and research[4]. In other words, the scheduling goal is to minimize the total completion time, minimize execution cost, minimize energy consumption, and maximize resource utilization. From (6), the value of $TotalTime$ is inversely proportional to the value of RA , so the objective function is:

$$f(X) = \omega_1 * TotalTime(X) + \omega_2 * EC(X) + \omega_3 * COST(X) \quad (8)$$

$$\omega_1 + \omega_2 + \omega_3 = 1, \text{ all belongs to } [0,1].$$

III. DESIGN OF CLOUD COMPUTING TASK SCHEDULING ALGORITHM

Differential Evolution Algorithm is proposed by Store and Price in 1995[5]. It is a bionics intelligent method to simulate the natural evolution rule of survival of the fittest and survival of the fittest. The core idea of DE algorithm is to generate new intermediate individuals by using random bias disturbance, and then through iteration and selection, the iterative search leads to the global optimal solution. At present, DE is widely used in the field of continuous optimization, in the field of discrete optimization research and application of less[6]. In this paper, the DE algorithm is applied to the discrete task scheduling problem of cloud computing, and the algorithm is improved for the application in this field.

A. Coding Method

In this paper, the Task-Resource is used as an integer coding method, the length of the code is the number of tasks to be scheduled, and the value encoded for each bit is the corresponding resource number to which the task is assigned. Such as the total number of tasks $m = 10$, the total number of resources $n = 5$ a cloud computing task scheduling problem, the application of the above encoding method results are: $chromosome[10] = (3, 4, 2, 5, 1, 4, 3, 1, 2, 3)$. The decoding result

is: tasks t_1, t_7, t_{10} are scheduled on compute resources P_3 , tasks t_2, t_6 are scheduled on compute resources P_4 , task t_4 is scheduled on compute resources P_5 , tasks t_5, t_8 are scheduled on compute resources P_1 .

B. Population Initialization

At present, most of the DE algorithm is applied to solve the problem of cloud computing task scheduling, often using random initial population approach, but the initial population quality is not excellent. Therefore, this chapter will adopt the Weighted Round-Robin Scheduling Algorithm to initialize the population. According to the computing power of available resources (also known as weight), the resource is allocated a reasonable task to be scheduled. The greater the weight, the more computing power, these resources can perform more application tasks, otherwise the contrary, some computing power is relatively weak resources can also perform application tasks. Using this initial population method, it ensured that the resulting population was of good quality.

C. Fitness Function

The fitness function is used to evaluate population

individual, in other words, to assess the quality of the task scheduling scheme. population individual. The greater the fitness value of the population, the stronger the survivability of the individual, and the easier it is to keep the evolutionary iteration process. The corresponding task scheduling scheme is better. The fitness function of IADE algorithm proposed in this paper is:

$$Fitness = \frac{1}{f(x)} \quad (9)$$

$f(x)$ is the objective function of the independent task scheduling model, as in (8).

D. Mutation Operator

DE algorithm uses the difference strategy to realize the individual variation, the mutation strategy adopted in this paper is as (10). The free exploration and inheritance of this strategy are relatively balanced, and have good adaptability.

$$V_i = X_i + F \times (X_{best} - X_i) + F \times (X_{p1} - X_{p2}) \quad (10)$$

X_i is the i th individual in the parent population, X_{best} is the best individual in the parent population, X_{p1}, X_{p2} is two randomly selected individuals from the parent population, and they are also individuals different from X_i , F is the zoom factor.

In the evolution process, in order to ensure the validity of the solution, it is necessary to judge whether the individual

value satisfies the boundary condition. If the boundary condition is not satisfied, the individual value needs to be rationalized as (11).

$$\begin{cases} V_i = X_i + F \times (X_{best} - X_i) + F \times (X_{p1} - X_{p2}), & X_{min} \leq V_i \leq X_{max} \\ V_i = X_{min} + (X_{max} - X_{min}) \times 0.1, & otherwise \end{cases} \quad (11)$$

X_{min} and X_{max} are respectively the minimum and maximum values of the boundary.

In order to avoid prematurity, we add an operator with adaptive mutation. According to the progress of the algorithm, the adaptive mutation algorithm is designed as follows:

$$\lambda = e^{\frac{1 - G_m}{G_m + 1 - G}} \quad (12)$$

$$F = 2^{\lambda - 1} \quad (13)$$

G_m Is the maximum number of iterations, G is the current number of iterations.

(12) and (13), the range of the adaptive zoom factor is $[0.5, 1]$. At the beginning of the algorithm, the zoom factor is larger, which is helpful to keep the individual diversity and avoid premature. With the increase of the algebra of the algorithm, the later zoom factor is close to 0.5, and the excellent information is preserved to avoid the destruction of the optimal solution and increase the probability of finding the global optimal solution.

E. Greedy Crossover

The crossover operation is to improve the diversity of individual population by random recombination of the variation vector V_i and the target vector x_i , The new cross vector can be generated from (14) by interpolating the $x_i(g)$ of the g th population $x_i(g)$ and the intermediate $v_i(g+1)$.

$$u_{j,i}(g+1) = \begin{cases} v_{j,i}(g+1), & rand(0,1) \leq CR \text{ or } j = j_{rand} \\ x_{j,i}(g), & otherwise \end{cases} \quad (14)$$

j_{rand} is a random integer $[1, 2, \dots, D]$, CR is the crossover probability, and the formula is:

$$CR = \lambda \quad (15)$$

It can be seen from the above formula, CR value range is approximately $[0, 1]$. At the beginning of the algorithm, a larger CR is conducive to open up new space, thus speeding up the

convergence rate. With the increase of the algebra of the algorithm, the CR is close to 0 in the later period, and the smaller CR can keep the population diversity and improve the accuracy of the algorithm.

F. Selection Operation

DE algorithm uses the greedy algorithm to select individuals into the next generation of population as (16).

$$x_i(g+1) = \begin{cases} u_i(g+1), & \text{if } f(u_i(g+1)) \leq f(x_i(g)) \\ x_i(g), & \text{otherwise} \end{cases} \quad (16)$$

If and only if the fitness value of the new vector $u_i(g+1)$ is better than the fitness of the individual $x_i(g)$, $u_i(g+1)$ individuals will accept, otherwise, $x_i(g)$ individuals will remain in the next generation of population, the father of the individual and the new generation of candidates between the one-to-one survival of the fittest, making the individual better than the parent individual, so that the population evolutionary convergence to the optimal solution.

G. IADE Algorithm Flow

IADE Algorithm Flow is:

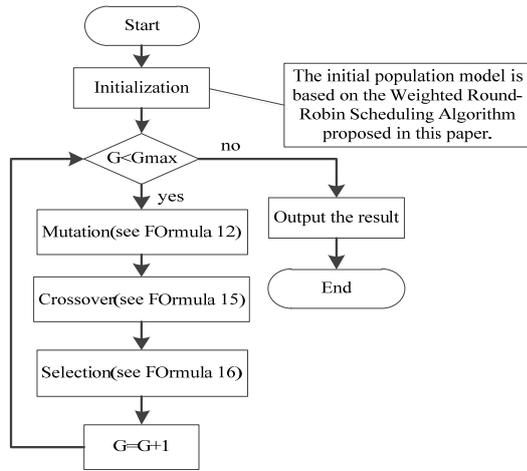


FIGURE I. FLOW OF IADE

IV. SIMULATION EXPERIMENT AND RESEARCH

A. Experiment Parameter Setting

In order to verify the feasibility of the proposed algorithm in cloud computing task scheduling, this paper extends cloud simulation platform CloudSim3.0.2[7-8]. The hardware environment of this experiment is: memory 4G, hard disk 500GB; software environment: Windows 7 operating system, MyEclipse development tools. The performance of genetic algorithm, standard differential evolution algorithm and the algorithm in this paper are compared and analyzed under the same experimental conditions.

Experimental data is divided into two parts: available resource and task attribute parameter settings, and simulation algorithm parameter settings.

TABLE I. VIRTUAL MACHINES AND TASK EXPERIMENT PARAMETER SETTINGS

Cloud Environment Entity Type	Parameter	Value
VM	datacenter	1-5
	total number of virtual machines	30
	number of PEs	1-2
	processing rate(Mips)	200-1000
	virtual machine memory(Mb)	256-2048
	virtual machine energy	1-3
	virtual machine cost	3-10
	bandwidth(Mb)	200-500
	resource management type	Time-shared
cloudlet	task length	1000-20000
	total number of tasks	30-300

Scheduling algorithm traditional Genetic Algorithm, standard Differential Evolution Algorithm and the proposed IADE scheduling algorithm need to use the experimental parameters as follows.

TABLE II. SCHEDULING ALGORITHM PARAMETER SETTINGS

Algorithm	Parameter	Value
Genetic Algorithm	population size	50
	crossover probability	0.15
	mutation probability	0.02
	evolutional generation	100
	terminate threshold	0.3%
Differential Evolution Algorithm	population size	50
	evolutional generation	100
	zoom factor	0.5
	crossover probability	0.9

B. Identify the Headings

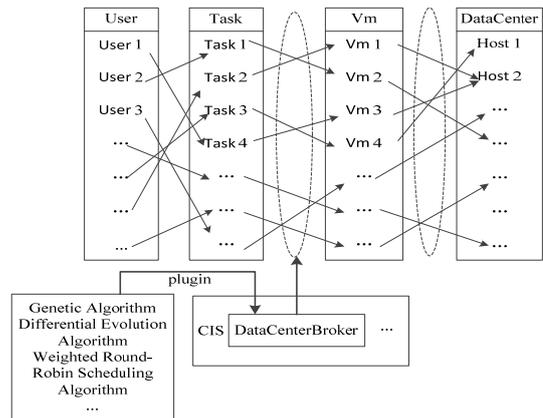


FIGURE II. WORKFLOW OF CLOUDSIM

Cloud computing task scheduling model under CloudSim simulation platform is shown in Figure 2.

In CloudSim simulation platform, tasks and virtual machine parameters initialization, task assignment and some other core operations are carried out in DataCenterBroker, the task scheduling model shown in Figure 1. During the task

scheduling process, the methods in the DataCenterBroker class are dispatched in turn. This paper expands the scheduling strategy of tasks in DataCenterBroker in the form of plug-ins. The scheduling algorithm proposed in this paper is embedded in DataCenterBroker in the form of a plug-in to realize part of the code of the scheduling simulation, as shown in Figure 3.

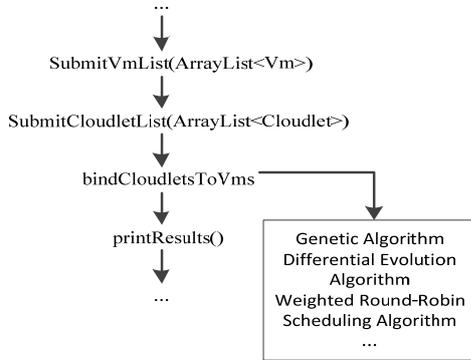


FIGURE III. EXTENDED FORMS OF DATACENTERBROKER

C. Experimental Results and Analysis

The average execution time, the average execution cost, the average resource utilization rate and the average execution energy consumption of the three scheduling strategies under different number of tasks are obtained by simulation. The data in the graph are the average of 10 sets of data, and specific comparison is as follows.

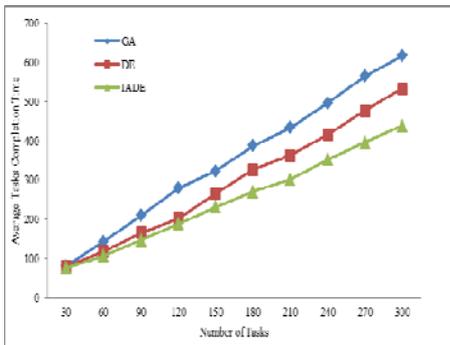


FIGURE IV. AVERAGE TASKS COMPLETION TIME

It can be seen from Figure 4 that IADE scheduling algorithm has no obvious advantages in reducing tasks completion time compared with the other two scheduling algorithms when the number of tasks to be scheduled is small. When the number of tasks is small, the available resources are still sufficient and the competition between the tasks is not intense. Therefore, the advantage of the proposed scheduling strategy is not obvious. With the increase of tasks, since the computing power of available resources is fixed, the competition between tasks is increasing so that the advantage of IADE scheduling strategy is more and more obvious.

With the increase of the number of tasks, the competition of resources becomes more intense, and the waiting time of application task is longer. Therefore, all the three scheduling

strategies will increase the task completion time with the increase of task number.

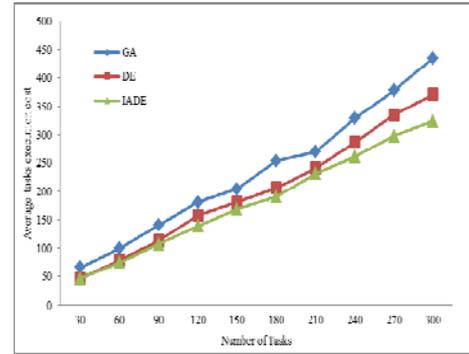


FIGURE V. AVERAGE TASKS EXECUTION COST

It can be seen from Figure 5 that the advantage of IADE scheduling algorithm in reducing the tasks execution cost is not obvious when the number of tasks is the same as the average task completion time, but its advantages are more and more obvious when the task amount is large. With the increase in the number of tasks, more intense competition for resources, some tasks will inevitably be assigned to the higher cost of resources to perform, so the three scheduling strategies will increase as the number of tasks, the average task execution costs also increase.

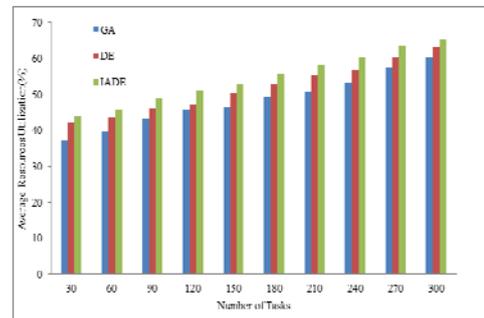


FIGURE VI. AVERAGE RESOURCES UTILIZATION

It can be seen from Figure 6 that, as in the above case, the advantage of the IADE scheduling algorithm in increasing the resource utilization rate is not obvious when the number of tasks is small, but the advantages are more obvious when the number of tasks is large. With the increase of the number of tasks, the resources utilization rate of the three scheduling strategies becomes higher and higher due to the fixed resource computing capacity and the increased resource execution time.

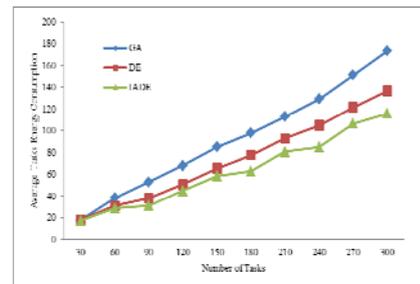


FIGURE VII. AVERAGE TASKS ENERGY CONSUMPTION

It can be seen from Figure 7, the three scheduling strategies in the small amount of tasks, their implementation of energy consumption difference is not large, and the resource competition is not intense, so the execution time is not much difference. However, with the increase in the number of tasks, more intense competition for resources, the virtual machine to perform more tasks, so the implementation of energy consumption also increased, So the implementation of energy consumption also increases. From the figure is not difficult to find, the more tasks, IADE scheduling algorithm advantage is more obvious.

V. SIMULATION EXPERIMENT AND RESEARCH

In this paper, we propose a cloud computing task scheduling strategy based on optimized Differential Evolution Algorithm, a task scheduling model with multiple parameters is established, and we verify the scheduling policy and scheduling model on the CloudSim emulator. The simulation results show that the scheduling strategy proposed in this paper can shorten the task completion time, improve the efficiency of resource utilization, meet the needs of users and increase the benefits of cloud providers compared with the traditional genetic algorithm and standard differential evolution algorithm.

REFERENCES

- [1] Armbrust M., Fox A., Griffith R., et al. A view of cloud computing[J]. *Communications of the ACM*, 2010, 53(4): 50-58.
- [2] Dai Y, Lou Y, Lu X. A Task Scheduling Algorithm Based on Genetic Algorithm and Ant Colony Optimization Algorithm with Multi-QoS Constraints in Cloud Computing[C]// *International Conference on Intelligent Human-Machine Systems and Cybernetics*. IEEE, 2015.
- [3] WU Guo-fang. Multi-objective Task Scheduling Scheme Based on Cuckoo Search Algorithm in Cloud Environment [J]. *Applications Research of Computers*, 2015(9):2674-2677.
- [4] Mathew T, Sekaran K C, Jose J. Study and analysis of various task scheduling algorithms in the cloud computing environment[C]// *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on*. IEEE, 2014: 658-664.
- [5] Storn R, Price K. Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces[M]. Berkeley: ICSI, 1995.
- [6] Pintea C M. Bio-inspired Computing[M]// *Advances in Bio-inspired Computing for Combinatorial Optimization Problems*. Springer Berlin Heidelberg, 2014: 3-19.
- [7] Calheiros R N, Ranjan R, De Rose C A F, et al. Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services[J]. *arXiv preprint arXiv:0903.2525*, 2009.
- [8] Calheiros R N, Ranjan R, Beloglazov A, et al. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms[J]. *Software: Practice and Experience*, 2011, 41(1): 23-50.