

# A Cloud Computing Resource Scheduling Scheme Based on Estimation of Distribution Algorithm

Niansheng Chen

Electronic and Information School  
Shanghai Dianji University  
Shanghai, China

Xiaoping Fang

School of Computer Science  
Hubei Normal University  
Huangshi, China

Xin Wang

Electronic and Information School  
Shanghai Dianji University  
Shanghai, China

**Abstract**—Resource scheduling is one of the key problems of cloud computing, no wonder, the scheduling policy and algorithm affect the performance of the cloud system directly. In order to improve the utilization of cloud computing resources and keep load balancing, a cloud computing resource scheduling algorithm based on estimation of distribution algorithm is proposed. In this algorithm, the idea of population based incremental learning(PBIL) algorithm is fully used. In this paper, cloud computing resource scheduling algorithm model is established firstly, and then objective solution is made by using the PBIL algorithm. Finally, the simulation analysis of algorithm performance is conducted. The simulation results show that the PBIL algorithm can take shorter time to complete task and achieve resource load balancing, especially, for the resource scheduling with large-scale task, the advantages are more apparent.

**Keywords**-cloud computing; resource scheduling; estimation of distribution algorithm; PBIL algorithm

## I. INTRODUCTION

Cloud computing has become a current research focus since IBM announced the plans of cloud computing in 2007[1]. Cloud computing virtualizes all resources, puts them in a resource pool, and schedules these resources for the use of each task transparently, so the virtualization mapping between application layer and virtual resource layer is the key part of cloud computing. Resource allocation is to find an optimization scheme to implement a reasonable mapping between tasks and resources. How to find a reasonable assignment scheme is critical for the resource scheduling. In recent years, many researchers have also studied the problem and proposed several techniques. Existing works in this context include the adaptive MapReduce scheduling algorithm[2], the method of the Longest Approximate Time to End[3] and scheduling algorithm based on the trust mechanism[4]. The two methods in [2-3] are complementary to each other and have similarity with the method presented in [5].

In a similar vein, the classical scheduling algorithm[6-8] is used to solve the problem of resource allocation respectively, which is easy to come true, but for the large-scale data, its performance is poor. In [9], the problem of resource distribution is settled effectively by adopting the idea of economics, which is accurate, but the modeling process is too

complicated. In addition, many researchers adopted the heuristic algorithm to solve the problem of resource allocation[10-11], which is the most widely used at present.

With the combination of the characteristics of genetic algorithms and statistics, estimation of distribution algorithms (EDAs) is a new class of intelligent algorithm obtaining the next generation of group through the probability vector samples of the solution space. The algorithm provides a good global search framework and has the characteristics of parallel computing, so it has become an international research focus in the field of evolutionary computation in recent years. Population based incremental learning (PBIL) algorithm is a specific form of EDAs, which has been applied in a lot of problems. According to the problem of cloud computing resource allocation, this paper designed a novel resource scheduling algorithm based on PBIL algorithm. The simulation results show that the resource scheduling algorithm can effectively perform the problem of cloud computing resource allocation.

## II. THE BASIC PRINCIPLE OF PBIL ALGORITHM

### A. The basic idea of PBIL algorithm

The PBIL algorithm, established by Baluja at Carnegie Mellon university United States in 1994[12], can well solve the problem of the binary encoding optimization. The core idea of the algorithm is to use a probability vector  $\vec{p}(x) = (\vec{p}(x_1), \vec{p}(x_2), \dots, \vec{p}(x_n))$  to represent the distribution of solution from solution space, where  $\vec{p}(x_i)$  is the probability of the i-th gene bit value of 1. The specific implementation process is as follows: 1) Firstly, in each generation, the number of  $M$  individuals are sampled randomly from the solution space based on the probability vector. The number of  $N$  individuals are selected to form an advantage group from the set of the number of  $M$  individuals according to a certain rule; 2) Secondly, the probability vector is updated by using the set of the number of  $N$  individuals; 3) Finally, the optimal solution of the problem is achieved by iterating *Step1* and *Step2*. The rule of updating the probability vector in *Step2* is given by:

The work was supported by The key disciplines of Shanghai Dianji University(13XKJ01-08) and The Research Project of Shanghai Dianji University under Grant NO. 14QD08.

$$p_{l+1}(x) = (1 - \partial)p_l(x) + \partial \frac{1}{N} \sum_{k=1}^N x_l^k \quad (1)$$

where  $p_l(x)$  is the individual probability vector of the  $l$ -th generation,  $\partial$  denotes the learning rate ( $\partial \in (0,1]$ ),  $x_l^k$  refers to the  $k$ -th individual of the  $l$ -th generation.

#### B. The procedure of PBIL algorithm

Based on the basic principle of PBIL algorithm, the executing procedure and steps are as follows:

*Step1*: Generate initial population;

*Step2*: Choose advantage group from initial population;

*Step3*: Update probability model according to the advantage group;

*Step4*: Get a new generation of group according to the updated probability model sampling solution space randomly;

*Step5*: Determine whether conform to the termination conditions. Terminate if the conditions meet, which suggests that the optimal solution had been found; If not, go back to *Step2* and continue evolution process.

In *Step2*, the estimation of distribution algorithm varies if the method to choose advantage group from the initial population differs. In *Step5*, there are many typically termination condition, such as the number of the iteration, the gene of a certain individual with the probability of all 1 or 0, the given minimum deviation and the variation tendency of the fitness.

### III. CLOUD COMPUTING RESOURCE SCHEDULING SCHEME BASED ON PBIL ALGORITHM

#### A. Cloud computing resource allocation model

Assume there are  $m$  tasks and  $n$  virtual resources in cloud computing resource scheduling.  $\{t_1, t_2, \dots, t_m\}$  and  $\{r_1, r_2, \dots, r_n\}$  are denoted as tasks and resources respectively. The definitions are as follows:

**Definition 1** Resource allocation matrix  $N$ : according to the number of tasks and resources of cloud computing resources scheduling, a  $m \times n$  matrix  $N$  is initialized. If the task  $t_i$  calls the resource  $r_j$ , the entry is labeled as  $N[i][j] = 1$ , else  $N[i][j] = 0$ .

**Definition 2** Resources call time matrix  $T$ :  $T$  is a  $m \times n$  matrix.  $T[i][j]$  denotes the time that each task  $t_i$  calls the resource  $r_j$  independently.

**Definition 3** Resource allocation model: Assume there are  $m$  tasks and  $n$  virtual resources in cloud computing resource scheduling, the resource allocation model is defined as follows:

$$TotalTime = \max_{j=1}^n h_j \quad (2)$$

where  $TotalTime$  is the total time that the  $m$  tasks are completed,  $h_j$  denotes the sum of the time that all the tasks are completed on the resource  $r_j$ . The value of  $h_j$  is defined as:

$$h_j = \sum_{i=1}^m N[i][j] \times T[i][j] \quad (3)$$

The goal of cloud computing resource scheduling is to make the  $TotalTime$  as small as possible when there are  $m$  tasks and  $n$  virtual resources in cloud computing resource allocation.

#### B. The design of resource scheduling algorithm based on PBIL

##### 1) encoding

Coding is the foundation that the algorithm begins to generate the initial population. The encoding scheme used in this paper is that the resource allocation matrix is converted into a dimension table which is namely represented as a chromosome.

For example, given four tasks calling four resources, so it

$$\text{forms a resource allocation matrix } N = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Further, a chromosome  $\{1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1\}$  is obtained. It denotes that the task  $t_1$  is allocated to resource  $r_1$  and  $r_4$ , the task  $t_2$  is assigned to resource  $r_1$  and  $r_2$ , the task  $t_3$  is allocated resource  $r_1$ ,  $r_2$  and  $r_3$ , the task  $t_4$  is assigned to resource  $r_3$  and  $r_4$ .

##### 2) Determine the fitness function

The goal of cloud computing resource scheduling is to make the total time  $TotalTime$  as small as possible when  $m$  tasks call  $n$  virtual resources. So the fitness function can be computed as follows:

$$fit = \frac{1}{Totaltime} \quad (4)$$

##### 3) The choice of advantage group

When the advantage group is chosen, fitness values are sorted firstly. Then the front half of the group is selected as an advantage group which can be as a part of new group for the following iterative operation.

##### 4) The adaptive changes of learning divisor

The relationship between learning divisor and iteration turns from close to distant gradually, and the speed of reduction becomes fast in the later stage. We use the following condition:

$$r_k = r_1 - (r_1 - r) * (\frac{k}{T})^2 \quad (5)$$

where  $k$  is the number of iterations;  $r_k$  denotes learning divisor and is the function of  $k$  changing with the different  $k$ ;  $r_1$  is the initial learning divisor ;  $r$  is the final learning divisor and is a limited value;  $T$  is the biggest number of iterations.

The adaptive change of the learning divisor is to maintain the global optimal in the early stage in order to reduce the number of running into the local optimum in the later stage.

#### IV. EXPERIMENT SIMULATION

The algorithm is simulated on the ClousSim platform to test the application effect of PBIL algorithm in cloud computing resource allocation.

To compare the completion time of the less task scheduling and multiple-task scheduling, we selected RR algorithm, SFLA algorithm and PBIL algorithm contained in ClousSim for our experiments to simulate the resource scheduling.

##### A. Simulation parameter setting

Simulation scenario 1: Given 500 chromosomes, 5 user tasks  $t_0, t_1, t_2, t_3, t_4$ , 3 virtual resource nodes  $r_0, r_1, r_2$ . The detailed parameters in scenario 1 are shown in Table 1.

Simulation scenario 2: Assume that there are 50, 150, 450, 150 tasks. It respectively calls the number of virtual resources from 20 to 40 to complete the task. The specific parameters in scenario 2 are shown in Table 2.

##### B. The analysis of simulation results

###### 1) The experimental results of simulation scenario 1

The simulation scenario 1 is mainly aimed the case of a few tasks in cloud computing resource scheduling. The comparison of the total time *TotalTime* from the three algorithms is shown in Table 3. The time which the three algorithms perform a single task is presented in Table 4.

TABLE 1: THE PARAMETERS CONFIGURATION OF SIMULATION SCENARIO 1

Clou	ClouL	Time			Datace
		r0	r1	r2	
$t_0$	4000	8.0	1.333	4.0	2
$t_1$	2000	4.0	0.667	2.0	2
$t_2$	2500	5.0	0.833	2.5	2
$t_3$	10000	20.0	3.333	10.0	2
$t_4$	500	1.0	0.167	0.5	2

From Table 3, we can see that the *TotalTime* using the PBIL algorithm proposed in this paper is shorter than RR and SFLA. The observations are also shown in Figure 1. But in fact the improvements are slightly less.

TABLE 2: THE PARAMETERS CONFIGURATION OF SIMULATION SCENARIO 2

Cloudlets	Vms	Simulations
50	20	300
150	20	300
450	40	300
4000	40	300

TABLE 3: THE COMPARISON OF THE *TOTALTIME* OF THE SIMULATION SCENARIO 1

Algorithm	Cloudlets	TotalTime(ms)
RR	5	40.231
SFLA	5	28.456
PBIL	5	23.542

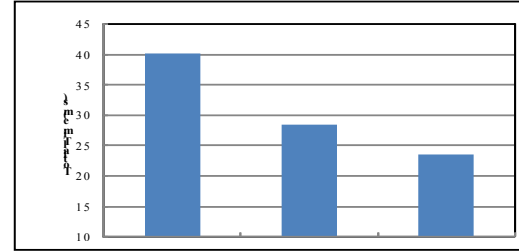


Figure 1. The comparison of the total time of the simulation scenario 1.

TABLE 4: THE COMPARISON OF THE TIME PERFORMING WITH A SINGLE TASK

Clou	ClouL	Time(ms)		
		RR	SFLA	PBIL
$t_0$	4000	16.12	13.26	2.93
$t_1$	2000	0.91	2.57	3.98
$t_2$	2500	2.56	3.20	2.54
$t_3$	10000	28.17	23.13	4.87
$t_4$	500	0.33	0.32	0.48

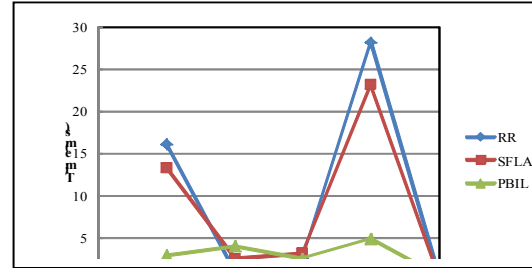


Figure 2. The comparison of the time performance with a single task.

TABLE 5: THE COMPARISON OF THE TOTAL TIME IN THE SIMULATION SCENARIO 2

Cloudlets	Vms	TotalTime(ms)		
		RR	SFLA	PBIL
50	20	498.23	253.62	183.42
150	20	1214.56	713.48	320.16
450	40	3289.56	1432.27	917.49
4000	40	31076.43	15324.26	7543.21

Both Table 4 and Figure 2 suggest that after using the PBIL algorithm, in terms of a single task execution time,  $t_0$  and  $t_3$  are shorter obviously, while the improvements of execution time  $t_1, t_2, t_4$  are not so great, which means that there are differences in resource nodes computing power in cloud computing environment. But using the PBIL algorithm makes resource load scheduling easier to reach balance.

## 2) The experimental results of simulation scenario 2

The simulation scenario 2 mainly explores the performance differences of three algorithms on the resource scheduling in more onerous tasks case. Table 5 shows the comparison among the total time *TotalTime* (the average of 300 times experiment results) taken to complete task by the three algorithms respectively in certain resources case.

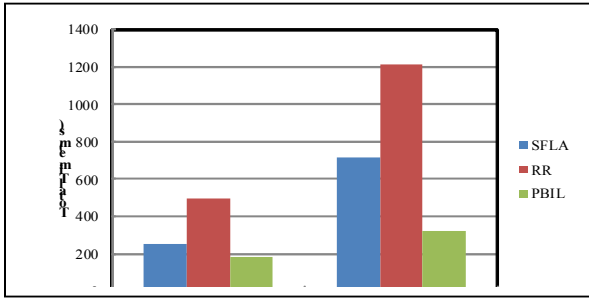


Figure 3. The comparison of the total time in the simulation scenario 2 ( $Vms=20$ ).

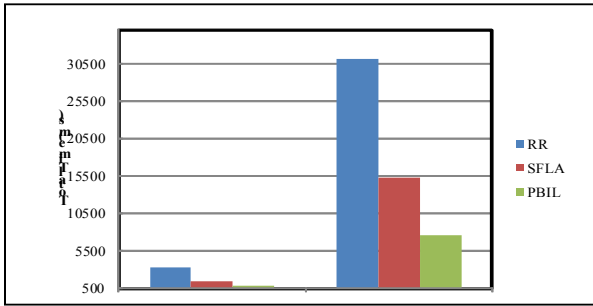


Figure 4. The comparison of the total time in the simulation scenario 2 ( $Vms=40$ ).

TABLE 6: THE COMPARISON OF THE TOTAL TIME WITH 4000 TASKS SCHEDULING

Vms	TotalTime(ms)		
	RR	SFLA	PBIL
20	45007.35	27034.64	16683.42
25	43278.78	20438.23	14782.73
30	38456.47	17079.73	10742.79
35	33689.28	16234.23	8452.95
40	31089.32	15343.67	7543.21

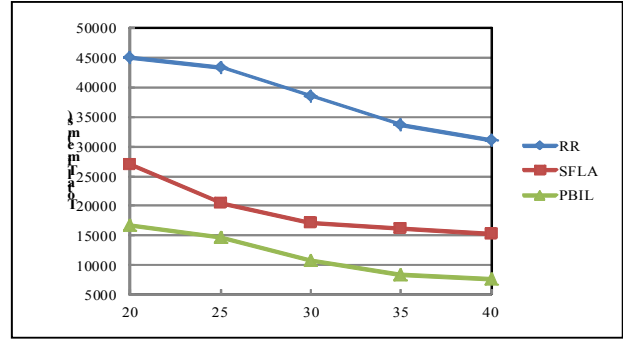


Figure 5. The comparison of the total time with 4000 tasks scheduling and different  $Vms$ .

As noted in Figure 3 and 4, it can be seen from Table 5 that in certain resources case, when it comes to cloud computing resource scheduling of multitask, the PBIL algorithm proposed in this paper uses the shortest time.

Table 6 shows the comparison among the total time *TotalTime* taken to complete task by three algorithms respectively when 4000 tasks call 20-40 virtual resources.

It can be seen from Table 6 that in cloud computing resource scheduling, when the task number is given, the total time taken to complete task reduces gradually with the increase of the number of the resources. From Figure 5, we find the trend of the curve. But the margins of decrease of total time taken by the PBIL algorithm to complete task is the fastest, which suggests that the PBIL algorithm has a significant advantage in the multi-task resource scheduling.

## V. CONCLUSION

In view of the problem of the resource allocation scheduling in a cloud computing environment, a cloud computing resource scheduling algorithm based on estimation of distribution algorithm is proposed and its basic implementation process is also given. At the same time, the experimental simulation and performance comparison are made on the CloudSim platform for the included RR algorithm, SFLA algorithm of CloudSim platform and PBIL algorithm proposed in this paper. The simulation results show that PBIL algorithm can well realize the resource load balancing and is very suitable for the resource scheduling with large-scale task in cloud computing environment.

## REFERENCES

- [1] CHU Ya, MA Ting-huai and ZHAO Li-cheng, "Cloud computing resource scheduling: Policy and algorithm," *Computer Science*, vol. 40, pp. 8-13, November 2013.
- [2] Beloglazov A, Buyya R, "Adaptive threshold-based approach for energy-efficient consolidation fo virtual machines in cloud data centers," *Proccddedings of the 8<sup>th</sup> International Workshop on Middleware for Grids, Clouds and e-Science*, Bangalore:ACM, 2010, pp. 1-6.
- [3] Matei Zaharia, Andy Konwinski, Anthony D. Joseph, Y Katz and Ion Stoica, "Improving mapreduce performance in heterogeneous environments," *Proceedings of the 8<sup>th</sup> USENIX symposium on operating systems design and implementation*, San Diego: The USENIX Association, 2008, pp. 29-42.
- [4] Wei Wanga, Guosun Zenga, Daizhong Tange and Jing Yao, "Clouding-DLS: Dynamic trusted scheduling for cloud computing," *Expert systems with Applications*, Vol. 39, pp. 2321-2329, February 2012.
- [5] Kumar S K S, Balasubramanie P, "Dynamic scheduling for cloud reliability using transportation problem," *Journal of Computer Science*, vol. 8, pp. 1615-1626, October 2011.
- [6] MA Li, LIU Gaoyuan, "Improved Min-Min Scheduling algorithm," *Computer Engineering and Applications*, vol. 48, pp. 69-73, June 2012.
- [7] LIU Qing, WU Zhefu, HE Xiongxiang and LIU Kai, "Analysis and Improvement of Min-Max Node Localization Algorithm," *Chinese Journal of Sensors and Actuators*, vol. 25, pp. 1126-1131, August 2012.
- [8] ZHANG Bin-qiao, "Cloud Computing Resource Scheduling Strategy Based on Catfish Particle Swarm Optimization Algorithm," *Computer Simulation*, vol. 30, pp. 392-395, November 2013.
- [9] Zhen Xiao, Weijia Song and Qi Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment," *Parallel and Distributed Systems*, vol. 24, pp. 1107-1117, June 2013.
- [10] IHsun Chuang, YuTing Tsai, MongFong Horng, YauHwang Kuo, JangPong Hsu, "A GA-Based Approach for Resource Consolidation of Virtual Machines in Clouds," *Intelligent Information and Database Systems Lecture Notes in Computer Science*, vol. 8397, pp. 342-351, Janurary 2014.
- [11] Fu Xie, Yunyun Du and Hongwei Tian, "A Resource Allocation Strategy Based on Particle Swarm Algorithm in Cloud Computing Environment," in: *Digital Manufacturing and Automation (ICDMA)*, Qingdao: IEEE, 2013, pp. 69-72.
- [12] Baluja S, "Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning," *Technical Report CMU-CS-94-163*, Pittsburgh, PA: Carnegie Mellon University, 1994.