ISBN 978-981-11-3671-9
Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering (WCSE 2017)
Beijing, 25-27 June, 2017, pp.30-36
doi: 10.18178/wcse.2017.06.005

Research and Implementation of Workflow Scheduling Scheme Based on Cloud Computing

Xi Sun[1+], Yijie Shi [1]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China.

Abstract. With the rapid development of cloud computing, task scheduling and resource allocation have been widely concerned in cloud computing. In this paper, the workflow scheduling in cloud computing is studied. It is found that the value of each dimension of the particle position coordinates in the PSO represents the index of the computing resources in the resource pool, which is not associated with the characteristics of the computational resources. About this issue, the current research is only associating the cost of computing resources and the index of computational resources. In this paper, we propose an improved strategy which associates the index of computational resources with the cost of computing resource and its memory together. After a lot of experiments, it has been proved that this improved strategy is more effective.

Keywords: cloud computing, resource scheduling, PSO, workflow

1. Introduction

The workflow scheduling in the cloud computing is the process of mapping tasks, which depend on each other, to distributed resources. Schedule is the core and difficult technology in the workflow of cloud computing, which determines the success or failure of workflow execution and execution efficiency. Generally, workflow scheduling in cloud computing is the NP-Hard problem. For the workflow given by the user, there is a large improvement and optimization space in the scheduling process.

In the cloud environment, how to reduce the cost and achieve the maximum benefit under the premise of no reducing the quality of service is the current research focus. The workflow in the cloud environment presents an abstract definition of the task, flexible configuration and automatic operation to improve the quality of service. The workflow scheduling is to manage the task, optimize the resources, and thus reduce the cost. In addition, the merits and demerits of task scheduling directly determine the success rate and efficiency of workflow execution. As the advantages of cloud computing has been widespread concerned, workflow scheduling in cloud computing has become the focus of the current research. In 2014, Rodriguez and Buyya [4] provided a new computational model for the scientific workflow on the cloud, defining the CMDCWS model in which the task was executed at a minimum cost in a certain period of time and using PSO [5] to obtain the resource scheduling sequence. In 2015, Hai-Hao Li et al. [1] associated the cost of resources with their index. M. Mao [2] and M. Malawski[3] use the cost and deadline as the standard to design the scheduling. SuchaSmanchat and Kanchana Viriyapant [6] classified the problems and techniques in cloud workflow scheduling. Mohammad Masdari et al. [7] conducted a comprehensive analysis of the workflow in cloud. Zhongjin Li et al. [8] added task security level scheduling to the scheduling. But they did not fully consider the balance between the cost and time. There are three major cloud providers in cloud [15]. The paper is based on IaaS.PSO is a simple and efficient algorithm, and has developed rapidly in recent years [9,10]. In addition to PSO, there are other algorithms can also be integrated into the cloud [11-13]. G.

+ Corresponding author. Tel.: +18811611228; fax: +86-010-62283192.
E-mail address:sunxi_mail@163.com.

Juve et al. [14] gave a detailed introduction to the workflow. They only use these algorithms but not consider the attributes of the resources and the balance between efficiency and cost.

The rest of this paper is organized as follows. The second part introduces the knowledge of workflow task scheduling in cloud computing. The third part introduces the improved strategy proposed in this paper. The fourth part is the experimental design and experimental results analysis. The fifth part gives the conclusions.

2. Backgrounds

2.1. Basic knowledge

Typically, we model a workflow in the cloud as a DAG (Directed Acyclic Graph). The nodes in DAG represent the tasks in the workflow. The weights of the directed edges in DAG represent the time for two tasks to transfer data between different virtual machines. The topology of the DAG represents the order in which the tasks are executed, and the child tasks can be executed only if their parent tasks are completed.

2.2. CMDCWS Model

In 2014, Rodriguez and Buyya [4] proposed a model called "cost-minimization and deadline-constrained workflow scheduling" (CMDCWS), which applies to the business needs in the cloud computing, and it completes the workflow task and minimizes execution costs in a certain time.

CMDCWS scheduling problem can be expressed as a five-tuple:$S = (\mathrm{T}, \mathrm{R}, \mathrm{M}, \mathrm{TEC}, \mathrm{TET})$. The meaning of the elements in the five-tuple is shown in Table 1. $\mathrm{R} = \{r_1, r_2, \dots, r_n\}$is a collection of available virtual machines. Each resource r_j uses ST_j to indicate the start time, and $ET_j$is used to indicate the end of the time. M represents the set of task-resource mappings, consisting of $r_i^j = \{t_i, r_j, ST_i, ET_i\}$, indicating that task t_i runs on resource r_j, the execution of the start time is ST_i, and the end time is ET_i. The cost of the resource per unit time is expressed by C_j, then the TET can be calculated using the equation (1):

$$\mathrm{TEC} = \sum_{i=1}^{|R|} C_i \times \lceil ET_i - ST_i \rceil \quad (1)$$

The TET can be calculated using the equation (2):

$$\mathrm{TET} = \{ET_t,\ \ \mathrm{t} \in \mathrm{T}\} \quad (2)$$

The constraints of this schedule can be expressed by the equations (3) and (4):

$$\text{Minimize } \mathrm{f} = \mathrm{TEC} \quad (3)$$
$$\mathrm{TET} \leq \text{deadline} \quad (4)$$

Table I: The meaning of the five-tuple

symbol	meaning
T	The collection of tasks in the workflow
R	The collection of resource
M	The mappings between tasks and resources
TEC	The execution cost of all the tasks in the workflow
TET	The execution time of all the tasks in the workflow

2.3. Particle Swarm Optimization(PSO)

The particle swarm represents the solution of the search space.If there are n tasks in the workflow, the dimension of each particle in the particle swarm is n. The particle i is associated with a position vector $\vec{x_i}$and a velocity vector $\vec{v_i}$ to indicate the current condition. The best historical position of the particle i is represented as$\overrightarrow{pBest_i}$and the best of all the $\overrightarrow{pBest_i}$is considered as the global optimal location$\overrightarrow{gBest}$. The initial values of velocity and position are random and are calculated by equations (5) and (6):

$$\overrightarrow{v_i(t+1)} = \omega \overrightarrow{v_i(t)} + c_1 r_1 \left(\overrightarrow{pBest_i} - \overrightarrow{x_i(t)}\right) + c_2 r_2 \left(\overrightarrow{gBest} - \overrightarrow{x_i(t)}\right) \quad (5)$$

$$\overrightarrow{x_i(t+1)} = \overrightarrow{x_i(t)} + \overrightarrow{v_i(t)} \quad (6)$$

In the equations (5) and (6), the inertia weight ω is set to 0.5 in the experiment, c1 and c2 are the acceleration coefficients, which are usually set to a random value of 2.0, r1 and r2 are random values in [0,1].

The pseudo-code of the PSO algorithm is presented in Algorithm 1.

PSO
1. Initialize the velocity and position of each particle in the particle swarm, and the initial value of the velocity and position of the particle is random. 2. Calculate the fitness value of each particle, if the fitness value is better than pBest, then the fitness value in the pBest. 3. Compare the pBest values of all particles, and gBest holds the optimal result in the pBest value. 4. Update the velocity and position of the particles according to equation (5) and equation (6). 5. Repeat steps 2 to 4 until the iteration ends.

Fig. 1: Algorithm 1

3. Improved scheduling strategy

3.1. Improved scheduling strategy

When the cloud providers provide the computing resources, the process of the particle self-learning is very blind. For example, when the value r_d in the dimension of the particle moves toward to its pBest or gBest, it is unclear how the task execution cost is expensive or cheap, and the r_d , pBest and gBest do not represent any information. Soit is easy to make pBest a resistance rather than to update pBest, especially in a workflow contains a large number of tasks, the performance was more obvious. As for the problem, we propose a new strategy that the cost per unit time of the resource provided by cloud providers and the memory of the resource should be the standard to renumber the resource and make it basic order.

3.2. Particle coding

First, a workflow contains n tasks, and then the dimension of each particle is the number of tasks in the workflow.For example, as shown in Fig. 2, the workflow contains 10 tasks, and then the dimension of the particle is 10-dimensional, and the position can be determined by 10 coordinates.

Secondly, there is a range of values for each dimension of each particle, assuming that there are currently m available resources in the resource pool for use by the workflow, and the range is $[1.0, m+1)$.The position of the particle is represented by 10 dimensional coordinate, which can be represented by the table 2(when the number of resources is 7):

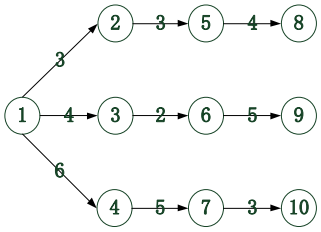

Fig. 2: An example of workflow

Table II: Particle's position

d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
1.2	3.4	2.3	5.4	6.7	7.4	5.3	3.3	2.7	6.9

The value of each dimension of the particle in Fig. 2 is rounded down to correspond to the index of the resource.In this algorithm, the position of the particle corresponds to the index of the resource. Table 2 is the position of a particle.According to Table 2, we can get the mappings between the task and the resource in

Table 3. For example, d1 in Table 2 represents task 1, and its corresponding value of 1.2 represents resource 1, so the task establishes a mapping relationship with the resource, that is, task 1 runs on resource 1. Similarly, d2 represents task 2, and its corresponding value of 3.4 represents resource 3, that is, task 2 runs on resource 3.

Table III: Mapping of tasks and resources

$d_1 \rightarrow r_1$	$d_2 \rightarrow r_3$	$d_3 \rightarrow r_2$	$d_4 \rightarrow r_5$	$d_5 \rightarrow r_6$	$d_6 \rightarrow r_7$	$d_7 \rightarrow r_5$	$d_8 \rightarrow r_3$	$d_9 \rightarrow r_2$	$d_{10} \rightarrow r_6$

3.3. Fitness value

In this paper, the fitness value function is the overall cost of the entire workflow task, that is, TEC (Total Execution Cost), which includes the cost of data execution and the cost of data transmission.

Firstly we initialize a pool of resources, using two arrays to represent the time of data transfer between tasks and the execution time of tasks on each resource.

Equation (7) shows the time of data transfer between 10 tasks in the workflow shown in Fig. 2.Since the task can nottransfer data with itself,the data transfer time matrix is an upper triangular matrix. TransferTime [i] [j] represents the data transfer time between task i and task j. If task i and task j are running on different virtual machines, the value of TransferTime [i] [j] is nonzero.If the task i and task j are running on the same virtual machine, the value of TransferTime [i] [j] is zero.

Equation (8) shows the execution time which the tasks are executed on seven resources, so the ExecutionTime matrix is 10 × 7. ExecutionTime [i] [j] indicates that if task i is assigned to the virtual machine j in the schedule, it is executed on virtual machine j for ExecutionTime [i] [j].

$$\text{TransferTime} = \begin{bmatrix} 0 & 3 & 4 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (7)$$

$$\text{ExecutionTime} = \begin{bmatrix} 2 & 3 & 5 & 6 & 7 & 8 & 4 \\ 9 & 10 & 4 & 6 & 8 & 5 & 3 \\ 5 & 2 & 4 & 7 & 9 & 10 & 11 \\ 4 & 8 & 6 & 7 & 5 & 3 & 2 \\ 3 & 2 & 4 & 7 & 5 & 6 & 9 \\ 6 & 2 & 7 & 4 & 5 & 8 & 9 \\ 8 & 3 & 5 & 7 & 8 & 9 & 10 \\ 10 & 9 & 7 & 6 & 3 & 4 & 2 \\ 13 & 2 & 5 & 3 & 7 & 8 & 9 \\ 12 & 11 & 4 & 7 & 9 & 6 & 5 \end{bmatrix} \quad (8)$$

3.4. Scheduling process

The processing steps to improve the scheduling strategy are as follows:

(1) Initialize T, R, M, TET, TEC.

(2) Calculate the start time of each task. If the task t_i hasno parent task, then the task runsimmediately. If task t_i has a parent task, it runs only if all of its parent tasks are completed.

(3) Calculate the execution time of each task and the time it takes to get to the virtual machine. When the task tj is running on a virtual machine that is different from the task ti, the data transfer time TransferTime[i][j] should be added to the execution time of the task ti.

(4) Calculate the time for each virtual machine to end the lease.

Through the above steps, we can get the workflow TET (Total Execution Time) and TEC (Total Execution Cost), which can get a scheduling S.

The pseudo code of the improved scheduling strategy is as follows:

Improved scheduling strategy

1. Initialize T, R, M, TET, TEC
2. //n is the total number of tasks in T
//di represents the i-dimensional coordinate of the particle
//j represents the number of the computing resource corresponding to the task
for i=0 to i=n-1
j=di;
if(task i has no parent task)
STi=ETj;
else
STi=max{max{ETp,p is the parent task of task i},ETj};
endif
3. executiontime= ExecutionTime[i][j];
transfertime=0;
//task c is the child task of the task i
for(c:i){
if the mapping resource of the task c is different to the task j
transfertime+= TransferTime[i][c];
endif
}
ETi= executiontime+ transfertime+STi;
mij={i,j,STi,ETi};
M=M∪{ mij};
4. if (j∉R){
ETj=STj;
R=R∪{j};
}
ETj= executiontime+ transfertime+ETj;
end for i
5. $TEC = \sum_{i=1}^{\lvert R\rvert} C_i \times \lceil ET_i - ST_i \rceil$
6. $TET = \{ET_t,\ t \in T\}$
7. S=(T,R,M,TEC,TET)

Fig. 3: Algorithm 2

4. Experiment analysis

4.1. Experimental environment

In this paper, we use CloudSim to simulate the cloud environment and the task of workflow scheduling algorithm. In order to test the performance of proposed strategy in this paper, the experiment is simulated for different scale. In case of the number of resources is 10, we make the experiment for the tasks with the scale of 10,20 and 100.

4.2. Parameters setting

In the experiment, the parameters are set as follows:

Table IV. Parameters setting

parameter name	parameter	Parameter value
inertia weight	ω	0.5
acceleration coefficients	$c_1 c_2$	2.0
random factor	$r_1 r_2$	[0,1]

4.3. Experimentaldata and results analysis

Table V. Experimental results

scale	algorithm	average cost	maximum cost	minimum cost
10	PSO	159204.1	166844.77866666665	155582.1982222222
	RNPSO	150208.8	152906.3031111111	147387.36466666666
	Improved PSO	122661.3	131624.59444444443	103350.33133333334
20	PSO	371738.9	377265.66177777783	367679.1113333333
	RNPSO	360480	365318.3275555555	356792.79888888897
	Improved PSO	328638	345205.13711111114	293666.61933333334
100	PSO	3494156.7	3516480.456666663	3465400.3808888905
	RNPSO	3428422.3	3445977.1771111093	3412812.171777777
	Improved PSO	3301909.1	3344337.4997777725	3214558.649333328

When the number of tasks in the workflow is 10, it can be seen that the proposed is better than PSO and RNPSO algorithm. In addition, with the increase of the number of tasks, the proposed strategy in this paper is better than PSO and RNPSO in cost optimization. That is to say, when the scale is 200, the proposed is better than the others.

5. Conclusion

In this paper, due to the PSO algorithm has a problem that the process of the particle self-learning is blind, so we propose a new strategy that the cost per unit time of the resource provided by cloud providers and the memory of the resource should be the standard to renumber the resource and make it basic order, so that the process of the particle self-learning is more reasonable. According to the experimental results, we conclude that the improved scheduling strategy is better and with the increase of task size in the workflow, the advantages of the proposed strategy are more obvious. Our future work is to associatethe index of the computing resources combine with the other characteristics.

6. References

[1] Hai-Hao Li, Student Member, IEEE, Yu-Wen Fu, Student Member, IEEE,Zhi-Hui Zhan. Renumber Strategy Enhanced Particle Swarm Optimizationfor Cloud Computing Resource Scheduling.IEEE Conference Publications,2015.

[2] M. Mao and M. Humphrey, “Auto-scaling to minimize cost and meet application deadlines in cloud workflows,” in Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal., 2011, pp. 1–12.

[3] M. Malawski, G. Juve, E. Deelman, and J. Nabrzyski, “Cost-and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds,” in Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal., 2012, pp. 1–11.

[4] M. A. Rodriguez and R. Buyya, “Deadline based resource provisioning and scheduling foralgorithm scientific workflows on clouds,” IEEE Transactions on Cloud Computing, vol. 2, no. 2, April-June 2014.

[5] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. IEEE Int. Conf. Neural Networks, 1995, pp. 1942-1948.

[6] SuchaSmanchat,KanchanaViriyapant.Taxonomies of workflow scheduling problem and techniques in the cloud.Future Generation Computer Systems,2015.

[7] Mohammad Masdari,SimaValikardan,ZahraShahi,Sonay Imani Azar.Towards workflow scheduling in cloud computing:A comprehensive analysis.Journal of Network and Computer Application,2016.

[8] ZhongjinLi,JidongGe,HongjiYang,LiguoHuang,HaiyangHu,HaoHu,BinLuo.A security and cost aware scheduling algorithm for heterogeneous tasks of scientific workflow in clouds.Future Generation Computer Systems,2016.

[9] M. Shen, Z. H. Zhan, W. N. Chen, Y. J. Gong, J. Zhang, and Y. Li, “Bi-velocity discrete particle swarm optimization and its application to multicast routing problem incommunication networks,” IEEE Transactions on Industrial Electronics, vol. 61, no. 12, pp. 7141-7151, Dec.2014.

[10] Y. H. Li, Z. H. Zhan, S. Lin, J. Zhang, and X. N. Luo, “Competitive and cooperative particle swarm optimization

with information sharing mechanism for global optimization problems," Information Sciences, vol. 293, no. 1, pp. 370-382, 2015.

[11] Y. L. Li, Z. H. Zhan, Y. J. Gong, W. N. Chen, J. Zhang, and Y. Li, "Differential evolution with an evolution path: A DEEP evolutionary algorithm," IEEE Trans. on Cybernetics, DOI:10.1109/TCYB.2014.2360752, 2014.

[12] M. D. Zhang, Z. H. Zhan, J. J. Li, and J. Zhang, "Tournament selection based artificial beecolony algorithm with elitist strategy," in Proc. Conf. Technologies and Applications of ArtificialIntelligence, Taiwan, Nov. 2014, pp. 387-396.

[13] Z. H. Zhan, J. Zhang, Y. H. Shi, and H. L. Liu, "A modified brain storm optimization," in Proc.IEEE World Congr. Comput. Intell., Brisbane, Australia, Jun. 2012, pp. 1-8.

[14] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, "Characterizing and profiling scientific workflows," Future Generation Comput. Syst., vol. 29, no. 3, pp. 682–692, 2012.

[15] P. Mell and T. Grance, "The NIST definition of cloud computing—recommendations of theNational Institute of Standards and Technology," Special Publication 800-145, NIST,Gaithersburg,2011.