Jestr Journal of Engineering Science and Technology Review 6 (5) (2013) 33-37

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Research Article

Time-Constrained Workflow Scheduling In Cloud Environment Using Simulation Annealing Algorithm

Chengfeng Jian*, Yekun Wang, Meng Tao and Meiyu Zhang

Computer Science and Technology College, Zhejiang University of Technology, Hangzhou-China

Received 19 August 2013; Accepted 21 November 2013

Abstract

In cloud environment, it is necessary to find the efficient algorithms to optimize time-cost of cloud-oriented workflow scheduling. In this paper, a Simulated Annealing algorithm based heuristic is put forward in order to solve the time-constrained scheduling problem. Time-cost of scheduling is consisted of tasks execution and data transmission. The simulation testing results demonstrates that, the time-cost of scheduling using this algorithm can cost less time compared with particle swarm optimization algorithm, SA-based scheduling can also balance the load on resources, and SA with good convergence can find global optimal solution faster.

Keywords: Cloud-oriented, SA, Time-cost, Scheduling

1. Introduction

Cloud computing has become the mayor focus of companies' exploration in Internet when we were into the era of information. An important feature of the cloud computing is to pay as required[1]. The suppliers in cloud platform provide their own resources to take part in the activities of the cloud platforms for charging fees according to the needs of customers[2].For customers, how to choose efficient suppliers is the particularly important issue. The research on cloud-oriented workflow is a new trend of applied science, and cloud-oriented workflow is the product of the combination of cloud computing and science workflow. For cloud-oriented workflow systems, one of the most important missions is to dispatch tasks to resources based on customer's requirements and the characteristics of cloud-oriented workflow tasks, as well as time-cost of scheduling[3,4].

In the process of cloud-oriented workflow scheduling, time-cost is one of the most important issue customers must consider when they choose the appropriate resources to run their cloud-oriented workflow tasks. Oliveira provided a adaptive scheduling algorithm to solve parallel resources scheduling problem in a cloud platform[5]. Pandey solved a scheduling problem by PSO considering the time-cost of tasks execution and data transmission[6].A two-step scheduling algorithm is proposed to use cloud resources to the greatest extent and reduce the time-cost at the same times[7].By reducing the time-cost to optimize cloud-orient workflow scheduling is essentially a multi-objective optimization problem, and the main task of evolutionary computation is to deal with multi-objective optimization problems. Many heuristic algorithms, such as genetic algorithm(GA) and Particle Swarm Optimization (PSO),are applied in the application of cloud-oriented workflow scheduling[8,9].

Simulated Annealing (SA) algorithm[10,11] is a heuristic adaptive algorithm with good convergence to solve optimization problem. It has been widely used owing to its little requirement to optimize problems. Ideas about Simulated annealing algorithm is first put forward by n. Metropolis et al., in 1953.Since the late 1980's, the idea was gradually applied to the solving process of the mass of single objective and multi-objective optimization problem. Lots of algorithms to solve such problems are derived from simulated annealing algorithm, such as CBSA algorithm[12] and Pareto Simulated Annealing (PSA) algorithm [13].By using the simulated annealing (SA) algorithm we solve the cloud-oriented workflow scheduling problem to minimize the time-cost of tasks execution and data transmission in this paper.

2.Time-constructed Cloud-oriented Workflow Scheduling

2.1. Structure of cloud-oriented workflow

The distributed deployment of resources lay the foundations in the clouds. Moreover, we generally propose workflow applications represented by the Distributed acyclic Graph(DAG) so that we can regard various workflow tasks in cloud platform as nodes of a DAG.The connection between each node represents the interdependence between tasks. We choose an workflow application represented by G=(V,E). V is the set of various nodes, and E is the set of

* E-mail address: zf_jian@163.com
ISSN: 1791-2377

edges. The set of workflow tasks can be represented by T={T1,T2,……,T_n}in the DAG, so each node represent a cloud-oriented workflow task. And $E = \{E_{ij} = \mathrm{Re}(V_i, V_j) \mid V_i, V_j \subseteq V\}$ is the connection of various nodes representing the interdependence among different tasks. Fig.1 depicts a set of cloud workflow tasks consisted of ten tasks which are represented by nodes. Additionally the connection between nodes represents the dependencies among tasks. T1 is the task of "entrance" and T10 is the task of "export". The arrows in the figure indicate the direction of data transmission.

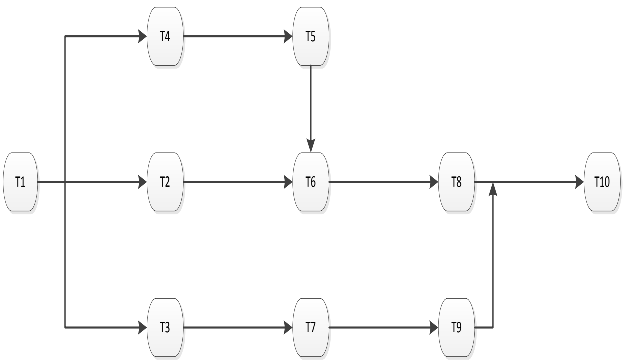

Fig. 1. An example of cloud-oriented workflow

2.2. Structure of cloud-oriented workflow

The systems of resource services in the cloud platform is huge service systems, and when choosing suppliers customers often refer to the time-cost of cloud-orient workflow tasks execution. As data transmission existing between different tasks, the time-cost of data transmission should be considered at the same time. Set $TIME_{sum}$ as the total time-cost of tasks execution and data transmission. Function(1) is the object function, which means the problem is to obtain the minimize of $TIME_{sum}$. $Run(T_i)$ represents the cloud resource that cloud-oriented workflow task T_i uses. Equation (2) presents the total execution time-cost of all the cloud-oriented workflow tasks. The time-cost of data transmission must be considered as equation (3) presented. If cloud-oriented workflow tasks T_i and T_j run on the same supplier's host computers, the time-cost of data transmission is 0.However,if they run on different ones, the transmission time-cost between T_i and T_j is equal to the time-cost of data transmission between suppliers' host computers these two tasks execute on. The total time $TIME_{sum}$ is the sum of $TIME_{run}$ and $TIME_{tran}$,which is presented as equation (4). The total time-cost of cloud-oriented workflow tasks is limited by the time-constrain, which is presented as $TIME_{\max}$.

$$\min TIME_{sum} \tag{1}$$

s.t $$TOTALTIME_{run} = \sum_{T_i \subset T} (TIME_{Run(T_i)})$$
$$Run(T_i) \subset P \tag{2}$$

$$TOTALTIME_{tran} = \begin{cases} 0, Run(T_i) = Run(T_j) \\ \sum_{T_i \subset T} \sum_{T_j \subset T} TIME_{tran}(Run(T_i), Run(T_j)), Run(T_i) != Run(T_j) \end{cases}$$
$$\forall E_{ij} \subset E \tag{3}$$

$$TIME_{sum} = TIME_{run} + TIME_{tran} \tag{4}$$

$$TIME_{sum} \le TIME_{\max} \tag{5}$$

2.3. Mathematic model

We first initialize the solution space and the objective function.The value of objective function is equal to total time-cost, namely equation[4]. Then we initialize temperature and the parameters of the temperature.The number of iterations on each given temperature and temperature decay function are given at the time.When temperature reach T_k ($T_k \subset [0, t_0]$), we can calculate the objective function $f(x_k)$ with x_k .

In each iterative process, we can determine whether the current solution is global optimal solution or not according to the Etropolis criterion[14]. Calculate $\Delta f = f(x_k) - f(x_{k-1})$,if Δf >0, x_k is the optimal solution. If it cannot meet the condition, then step into next operation: (1) Calculate the probability of selection;(2) Generate a random number K in the interval (0,1) and compare it with P. If K<P, we set the current solution as the optimal solution.Otherwise the optimal solution remain unchanged.

Cool down the temperature according to the temperature decay function, then repeat steps 2,3. If the algorithm reaches the maximum number of iterations, then the algorithm stops.

3. Simulation and Analysis of Results

Experiments were carried out to simulate cloud-oriented workflow scheduling by using simulated annealing algorithm(SA) .We mainly consider the time it takes to perform the task and the time for data transmission.

According to the description of the second part, time-cost is consist of time cost for tasks execution and data transmission between tasks in the scheduling. In the experiment, we used the Matrice to present time-cost of tasks to execute in different suppliers' host computers.Time-cost of data transmission between different tasks and the relationship of workflow tasks in the cloud platform are also present by Matrices. EXE_{ij} presents the time-cost of the cloud-oriented workflow task T_i running on the host computers of the resource supplier P_j . DT_{ij} presents the time-cost of data transmission between task T_i and task P_j ,which is a symmetric matrix.There are examples as following:

$$EXE_{3\times4} = \begin{bmatrix} 17.1 & 7.4 & 9.2 \\ 3.2 & 12.3 & 2.1 \\ 5.2 & 4.9 & 9.0 \\ 10.7 & 6.9 & 5.7 \end{bmatrix}$$

$$DT_{3\times3} = \begin{bmatrix} 0 & 9.6 & 7.8 \\ 9.6 & 0 & 4.2 \\ 7.8 & 4.2 & 0 \end{bmatrix}$$

Matrix RE shows the relationship between different tasks. If $RE_{ij} = 1$,it means that T_j depends on T_i and there are data transmission between them. Otherwise, data transmission between them does not exist. There is an example which can describe the relations between tasks in Fig.1 :

$$RE_{10\times10} = \begin{bmatrix} 0&1&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

Matrices EXE , DT and RE are set as the initial data of experiments. They store the information of tasks and the resources of supplier, as well as the relationship between them.

Firstly, we compare the time-cost of scheduling by SA and particle swarm optimization algorithm(PSO) when dealing with different sizes of problems. PSO is firstly proposed by Kennedy and Eberhart in 1995,which is a optimization technique on account of self-adaptive global research[15,16].When PSO is used in this paper for the simulation of cloud-orient workflow scheduling, inertia weight value ω is set as 1.2, and the value of acceleration coefficient $c1, c2$ are both set as 2.The scales of problems can be reflected by the number of tasks, which were varied during the experiments from 20 to 80.The first experiment is mainly to test the efficiency of scheduling by SA and PSO when dealing with different scales of problems. The results are showed in Fig.2.The X-axis represents the number of tasks in the scheduling process. Larger number of tasks means lager scales of the problems that SA and PSO must deal with. The Y-axis show the time SA and PSO spend on dealing with different scales of problems. With the size of problems increasing,the structure of DAG becomes more complex, and the time-cost of tasks execution and data transmission increases at the same time. When number of tasks is small, SA and PSO spend almost the same time to schedule tasks. As the increasing number of tasks, the time-cost of scheduling by SA increases slower than PSO. When the number of tasks is lager than 30, SA spends less time than PSO. Especially when number of tasks is 60, scheduling by SA can save 18.4% time-cost of PSO. As the number of tasks raises to 70, SA can save 22.4% time-cost. Scheduling by SA cost less time because of the better quality of optional solution selected by Etropolis criterion and the lower possibility to miss the optional solution. When choosing optional solution, general search algorithms may reduce the solution diversity and make the searching scope narrow. However, SA can avoid these problems.Fig.2 reflects the higher efficiency and better performance of scheduling by SA. So when handling large a scale of problems, SA is easy to be chose.

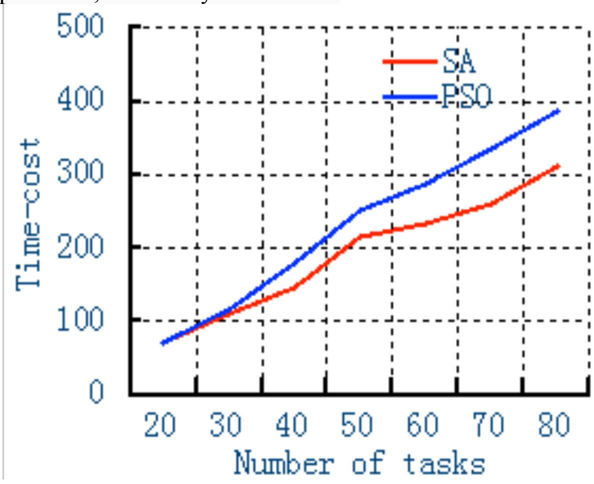

Fig. 2 Time-cost for different scales of problem

How to achieve a comprehensive resources loading is a big problem for tasks scheduling in cloud platform with many web resources. At the same time, we should take the issue of load balancing into consideration. So load balancing is one of the most important targets of cloud-oriented workflow scheduling, and it depends on tasks scheduling strategy. Fig.3, Fig.4 and Fig.5 show the tasks loading on suppliers' host computers when number of tasks is 15,20and25.In this experiment, there are 3 resources. When dealing with different scales of problems by SA, there are approximate 33% of cloud-orient workflow tasks in every supplier's resources so that all the machines of suppliers can be in the stable working condition. The results indicate that scheduling by SA can balance the load of tasks on suppliers' host computers to reduce pressure of suppliers. Scheduling by SA can improve rate of load balance in cloud resources. It can also improve servers' performance and efficiency through reducing the burden on a single server and take advantage of resources adequately.

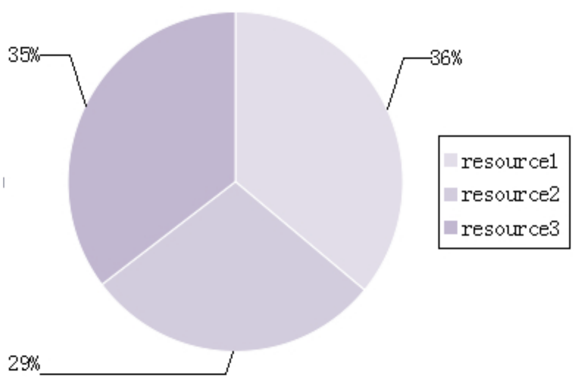

Fig. 3 Number of tasks=15

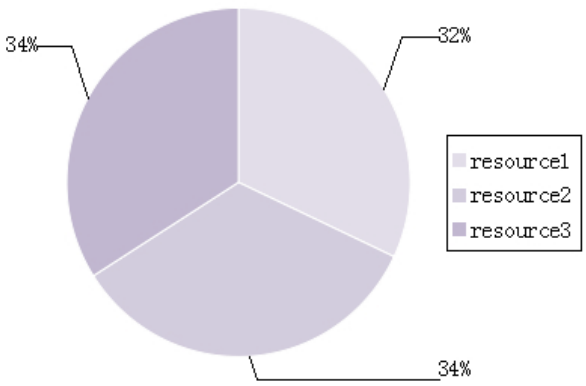

Fig. 4 Number of tasks=20

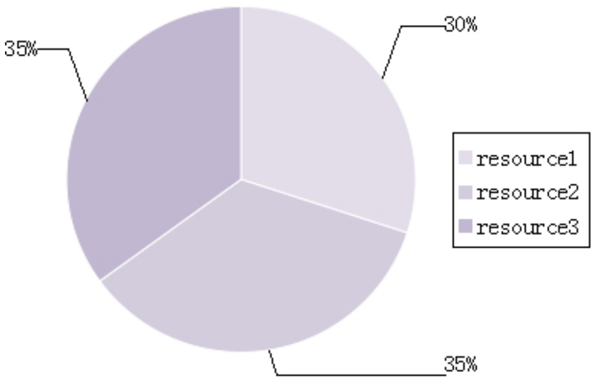

Fig. 5 Number of tasks=25

As the size of problems increasing, the complexity of the algorithm and the run time of clouds suppliers' CPU increase at the same time. When handling different scales of problems, the running time of CPU also shows the ability of the algorithm to solve the different scales of problems. It also shows whether this algorithm can be applied to practical cloud-oriented workflow scheduling. In this experiment, different scales of cloud-oriented workflow systems were represented by different scales of DAGs, which were randomly produced by programs. We did simulations many times to reduce the impact of accident factors. Table I, Table II, and Table III show the CPU running time using SA to complete these tasks scheduling when the number of tasks varying from 10 to 120. The growth of running time by using SA is evenly. It spent about 7 seconds to complete scheduling according to our model descried in part II When the number of tasks reaches to 120, which means that the time spent in cloud-oriented workflow scheduling by using SA is acceptable When handling large scales of problems. SA can obtain better performance with faster running speed because it cost less time to select optional solution according to Etropolis criterion.

Table 1. Time-cost of the tasks' number between 10 and 40

The number of tasks	10	20	30	40
Time(s)	0.143	0.747	0.874	1.128

Table2. Time-cost of the tasks' number between 50 and 80

The number of tasks	50	60	70	80
Time(s)	1.632	2.193	2.862	3.601

Table3. Time-cost of the tasks' number between 90 and 120

The number of tasks	90	100	110	120
Time(s)	4.417	5.322	6.309	7.362

Fig.6 shows the convergence of time-cost of scheduling by SA, with number of iterations changing and number of tasks varying form 10 to 30.Convergence is an important characteristic to show the ability of an algorithm to find the stable solution quickly. When handling a multi-objective optimization problem, SA can converge soon and avoid premature convergence. From Fig.6 we can see that when the number of iteration is about 10,a solution which is close to the optional solution can be found.Starting from that the number of iteration is 20,the trend of convergence is clear. The results describe that scheduling by SA can find the minimum value of time-cost very fast.

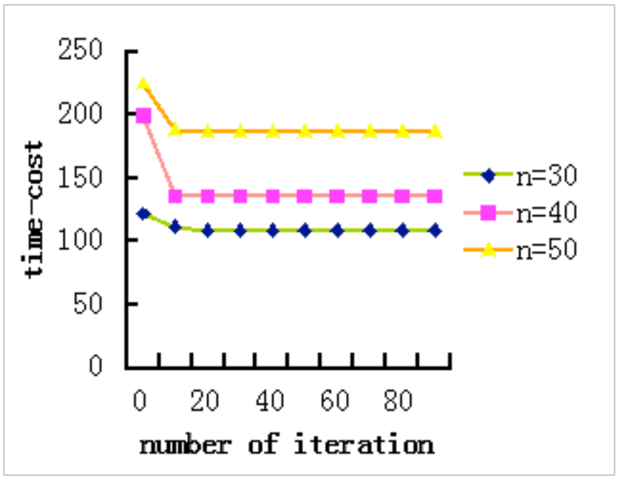

Fig.6 Time-cost convergence

4. Conclusions

Time-cost of cloud-oriented workflow scheduling is mainly considered by customers when choosing the providers of cloud resources. In this paper, we used Simulated Annealing(SA) algorithm to solve the cloud-oriented workflow scheduling problem for obtaining the minimize of time-cost under time-constrain. We considered both the time-cost of tasks execution and time-cost of data transmission between different tasks. Simulated Annealing(SA) is a heuristic adaptive algorithm with good convergence to solve optimization problem. In the experiments, SA is compared with particle swarm optimization algorithm (PSO),and the results show that SA can spend less time on scheduling than PSO when dealing with large scale of problems. Moreover, scheduling by SA can balance the task-load on resources to avoid so many tasks running on a single resource. It proves that SA can deal with large scale of problems.

References

1. Gutierrez-Garcia JO, Sim KM, “Agent-based cloud workflow execution”, Integrated Computer-Aided Engineering,19(1),2012,pp. 39-56.
2. Tolosana-Calasanz R., BañAres JÁN, Pham C., et al. “Enforcing QoS in scientific workflow systems enacted over cloud infrastructures”, Journal of Computer and System Sciences, 78(5),2012, pp. 1300-1315.
3. R.Kopitov,“Formalization of a reliable enterprise design”,Computer Modelling and New Technologies,16(1),2012,pp.15-29.
4. Chunlai Chai, “Modelling resource-constrained project scheduling problem and its solution by genetic algorithm”, Journal of Digital Information Management,11(2),2013,pp.87-92.
5. Oliveira D., Ogasawara E., Ocaña K., et al. “An adaptive parallel execution strategy for cloud□based scientific workflows”, Concurrency and Computation: Practice and Experience, 24(13),2012, pp. 1531-1550.
6. Pandey S., Wu L., Guru SM., et al. “A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments”, Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on. 2010,pp. 400-407.
7. Wu Q., Zhao Y., “A cost-effective scheduling algorithm for scientific workflows in clouds”, Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International. 2012,pp.256-265.
8. Barrett E., Howley E., Duggan J., “A learning architecture for scheduling workflow applications in the cloud”, Web Services (ECOWS), 2011 Ninth IEEE European Conference on Web Services. 2011,pp. 83-90.
9. Zhang X., Ling-zhi M., “Improved multi-objective particle swarm optimization algorithm for service-workflows scheduling”, Mechanic Automation and Control Engineering (MACE), 2010 International Conference on.2010,pp. 5548-5552.
10. Aarts E., Korst J., “Simulated annealing and Boltzmann machines”, 1988.
11. Van Laarhoven PJM, Aarts EHL, “Simulated annealing”, Springer Netherlands, 1987.
12. Prakash PKS, Ceglarek D.,Tiwari MK.,“Constraint-based simulated annealing (CBSA) approach to solve the disassembly scheduling problem”, The International Journal of Advanced Manufacturing Technology, 2012, 60(9-12),pp.1125-1137.
13. Abdelsalam HM, Mohamed AM.,“Multi-objective simulated annealing algorithm for partner selection in virtual enterprises”, Artificial Intelligence, Evolutionary Computing and Metaheuristics. Springer Berlin Heidelberg, 2013,pp.751-774.
14. Vasan A., Raju K S., “Comparative analysis of simulated annealing, simulated quenching and genetic algorithms for optimal reservoir operation”, Applied Soft Computing, 2009, 9(1),pp.274-281.
15. Eberhart R., Kennedy J., “A new optimizer using particle swarm theory”, Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on. 1995,pp. 39-43.
16. M.A. Zaman, S.A. Mamun, Md. Gaffar,et al. “Phased array synthesis using modified particle swarm optimization”, Journal of Engineering Science and Technology Review,4(1),2011,pp.68-73.