Tasks Scheduling Method Based on Competitive Co-evolutionary Algorithm

Haijie Yu[1,*], Sheng Su[2] and Qingguang Guo[2]
[1]School of Economics and Management, Southwest Petroleum University, Room 202, BoXue Building, No.8, Xindu Ave, Chengdu, 611731, China
[2]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[*]Corresponding author

***Abstract*—In previous studies about cloud computing scheduling algorithm, people usually ignore impact of data transmission between tasks, but data locality and data transmission have a great influence on the data-intensive task, especially for scientific workflow scheduling. This thesis explains importance of data locality and data transmission. Taking into account that users are more concerned about the completion time, a competitive co-evolutionary immune genetic algorithm, combining with immune algorithm and competitive co-evolutionary algorithm, then a new algorithm is proposed for scheduling problem in scientific workflow for data intensive tasks, which aims to minimize the completion time. Finally, In the cloud simulation platform building by CloudSim, scheduling model and corresponding algorithm proposed in this paper are carried out experiments. The results demonstrate effectiveness of the proposed scheduling algorithm.**

Keywords-cloud computing; co-evolution; task scheduling; workflow

I. INTRODUCTION

Cloud computing is one of the main trends of future information technology development [1], which attracted a large number of scholars and experts and engineers to make a systematic research. Many Internet companies, such as Google, IBM, Amazon, Yahoo, Microsoft, Salesforce, take cloud computing as a core business strategy and begin commercial process of cloud computing.

Cloud computing is development of parallel computing, grid computing, and it is a kind of distributed computing. Its basic idea is automatically split huge calculation processing program into numerous smaller subroutine through the network, then send to the huge system composed of multi department server, and give processing results back to the users after search and calculation and analysis. Resources network of providing those is called cloud.

User group for cloud computing providing service is huge, so task number in the cloud is enormous. System handles large number of tasks every minutes, therefore, task scheduling is the focus and difficulty in cloud computing[2-4] .Due to data transmission problem in cloud computing, data transmission is added to problem model of this paper, and study scientific workflows scheduling in the cloud computing environment, then pursue minimization of workflow completion time.

II. SCHEDULING PROBLEM OF SCIENCE WORKFLOW IN CLOUD COMPUTING

In the past research, DAG (directed acyclic graph) was usually used to represent task scheduling algorithm for scientific workflow [5-6], workflow scheduling model in this paper in the cloud computing environment are also using the DAG diagram to describe the problem.

This paper introduced pseudo input node and pseudo output node, pseudo input node as a parent node of all input nodes; pseudo output node as son node of all the output node.

As shown in Figure I, $t_1, t_2, t_3, t_4, t_5, t_6, t_7$ is the real task, solid lines between them indicate that they are a true dependency;

while t0 and T8 were pseudo input nodes and pseudo output nodes, and between them and real task node with a dotted line connection. Relationship between pseudo node and real node is assumed, and pseudo nodes have no input and output as assumed nodes, running time is also 0.

This only provides a more general description means of workflow DAG graph and convenience for the study on DAG diagram.

According to scientific workflow definition of DAG diagram $G = (T, E)$, T indicates task set, E indicates dependent relationship of father and son. Definition $T = \{t_1, t_2, t_3, ..., t_n\}$, t_i represents i task, $i \in [1, n]$, n is total number of tasks.

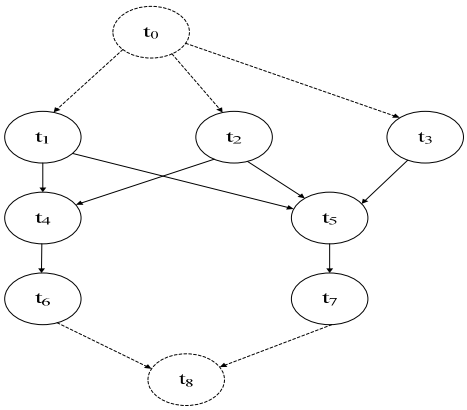

FIGURE I. WORKFLOW WITH PSEUDO INPUT NODE AND PSEUDO OUTPUT NODE

Each task execution can not be interrupted, and task scheduling is non preemptive. E is the set of edges for DAG , $< t_i, t_j >$ indicates t_i is parent node of t_j , t_j is dependent on t_i , t_j need to wait for completion of t_i and output results of t_i .No son node task is as the output tasks t_{out} , no parent node task is as the input task t_{in} . $Result$ is a $n*n$ matrix, and it is for the transmission of data between memory task. $Result[i][j]$ is used to store data transmission from the task t_i to the task t_j .

Define $RG = (R, V)$ as virtual resources and relationships among them $R = \{R_1, R_2, R_3, ..., R_m\}$, R_i indicates virtual resource i, m is the number of virtual resources; V is $m*m$ matrix, which is used to represent transmission bandwidth between virtual resources.

Among them, $V[i][j]$ indicates transmission speed between virtual resource R_i and virtual resource R_j . since transmission speed between two virtual resources connected is the same value, matrix V is a diagonal matrix. Data transmission time between tasks(Transfer, Time, TTran) is:

$$T_{Tran}(t_i, t_j) = \begin{cases} \dfrac{Result[i][j]}{V[R(t_i)][R(t_j)]} & R(t_i) \neq R(t_j) \\ 0 & R(t_i) = R(t_j) \end{cases} \quad (1)$$

Among them, $R(t_i)$ and $R(t_j)$ respectively represent virtual resources of task t_i and task . t_j . Calculation ability of each virtual resource is different, even if the same task is assigned to different resources virtual machine, execution time of tasks are not the same. Define calculation ability of virtual machine resource R_j is $sp(R_j)$, Calculation amount of task t_i is indicated as $sz(t_i)$, execution time of task t_i in a virtual machine resource (Execution Time, TExe) can be defined as:

$$T_{Exe}(t_i, R_j) = \frac{sz(t_i)}{sp(R_j)} \quad (2)$$

Among them, measurement unit of task size is number of instructions, and millions of instructions per second (MIPS) is used as unit of calculation speed of virtual machine resource. Aim of problem model is to minimize completion time, and target function is defined as start time of pseudo output task.

Each task t_i in the workflow has two time points, respectively are task start time $T_{Start}(t_i)$ and task end time $T_{Finish}(t_i)$. Pseudo input node is special node, start time and end time for it:

$$T_{Finish}(t_{in}) = T_{Start}(t_{in}) = 0 \quad (3)$$

As start time and end time of pseudo input node t_{in} and its son node t_s :

$$T_{Start}(t_s) = T_{Finish}(t_{in}) + T_{Wait}(t_s) \quad (4)$$

$$T_{Finish}(t_s) = T_{Start}(t_s) + T_{Exe}(t_s, R_j) \quad (5)$$

But for the start time and end time of real task t_i are:

$$T_{Start}(t_i) = \max_{t_p \in pred(t_i)} \{T_{Finish}(t_p) + T_{Tran}(t_p, t_i)\} + T_{Wait}(t_i) \quad (6)$$

$$T_{Finish}(t_i) = T_{Start}(t_i) + T_{Exe}(t_i, R_k) \quad (7)$$

Among them, every real task node has one or more father nodes, $pred(t_i)$ represents a collection for father node of task t_i , $T_{Wait}(t_i)$ represents that duration time of task t_i waiting for resources. Task requires all of the father nodes data, therefore, we must wait for all the father nodes execution

completing and all data transferring to the virtual machine resources of performing the task.

Only after all the tasks being executed, can pseudo output node be executed. Pseudo output node execution time is zero, however, in order to collect the final data, pseudo output node needs to wait for all the data transfer completion of the father nodes . Completion time of the objective function of the whole workflow (makespan) is:

$$makespan = T_{Start}\left(t_{out}\right) \tag{8}$$

III. ALGORITHM AND EXPERIMENT VERIFY

For scheduling problem of this paper, combined with evolutionary algorithm and immune algorithm, competitive co-evolutionary immune genetic algorithm is proposed. The algorithm is described as follows:

1. set variables: initialization of population size of population and population and corresponding maximum value are respectively , , and antibody similarity threshold value is , crossover probability , mutation probability , and the maximum number of population iterations .

2. Randomly generated two initialization population, and every individual using coding way. A population evolution algebra.

3 When population is non empty, selection of the population (using tournament selection method), crossover and mutation, produce new; otherwise, initialization of population .

4, When population is non empty, selection of the population (using tournament selection method), crossover and mutation, produce new; otherwise, initialization of population .

5. Population is as main population, while populations is as competitive population, then generate population fitness of every individual by competitive strategy, then calculate polymerization fitness of every individual according to formul.

6. Population is as main population, while populations is as competitive population, then generate population fitness of every individual by competitive strategy, then calculate polymerization fitness of every individual according to formula .

7. Calculate competition degree of population and, enlarge of shrink population size, and keep the best individual, then result in a new population of the next generation and.

8. if less than or equal to , then jump to the third step.

9, Find individual of minimum completion time in population and population.

IV. SIMULATION AND ANALYSIS OF ALGORITHM

This paper mainly simulate cloud computing environment by cloud computing simulation software -CloudSim[7], simulation experiment scheduling algorithm based on CCIGA algorithm. CCIGA algorithm needs to set their parameters when it is experimentally verified , and set two initial populations size as 50, the maximum population size is 300, and the opponent is 0.1 times contemporary population size. Population iterations are 400, and crossover probability is 0.8, and mutation probability is 0.1, and similarity threshold of antibodies is 0.9, and polymerization adaptive regulation factor $\alpha = 2$.

Set population size of contrast genetic algorithm as 100, and crossover probability is 0.8, and mutation probability is 0.1. Five experiments for workflow balance structures of the three scales were performed, and analysis of results is as follow. As shown in Figure II, scheduling result of CCIGA algorithm is better than GA algorithm, along with the increase of workflow task scale, difference between results of CCIGA algorithm and results of GA algorithm is more big. Search ability of CCIGA algorithm for workflow scheduling problem of larger scale is stronger than GA algorithm, which is consistent with dynamic fitness and maintain diversity of the population by use of population competition and polymerization fitness of CCIGA .

As shown in Figure III, for protein identification workflow of non-balance structure and nerve scientific workflow of balance structure, we contrast average value of 5 scheduling results with GA, CCIGA algorithm is more outstanding than GA algorithm for scheduling problem of complex structure.

As shown in Figure IV, shows the best individual changes in the iteration process of CCIGA algorithm and GA algorithm, as can be seen from the graph, optimal individual CCIGA algorithm not only produces excellent than GA algorithm, but convergence speed is also faster than GA algorithm.

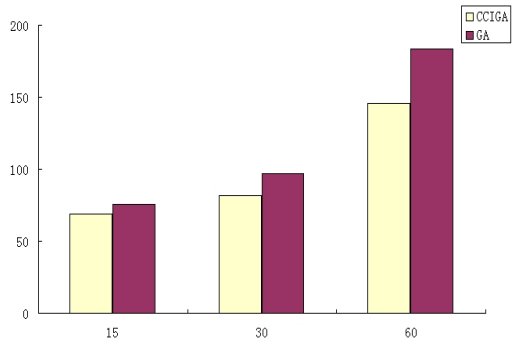

FIGURE II. SCHEDULING RESULTS FOR DIFFERENT TASK SIZE

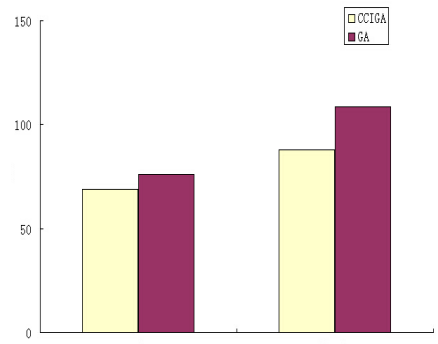

FIGURE III. SCHEDULING RESULTS OF BALANCE AND NON-BALANCE WORKFLOW FOR 15 NODES

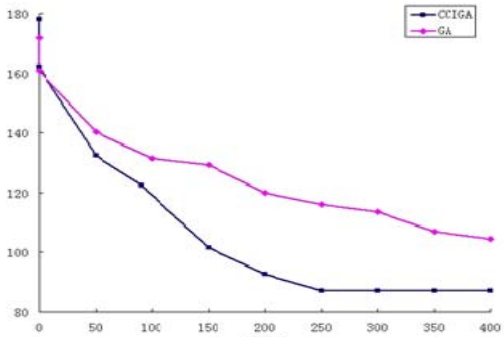

FIGURE IV. DYNAMIC PERFORMANCE DIAGRAM FOR PROTEIN IDENTIFICATION WORKFLOW

V. SUMMARY

For scientific workflow models with data dependencies, we added data transmission into tasks, and took scientific workflow minimum completion time as a goal. We proposed a scheduling algorithm based on immune genetic algorithm and competitive co-evolution algorithm. Results analysis and simulation experiment of scheduling model and scheduling algorithm are completed by using CloudSim.

ACKNOWLEDGMENT

This work was supported by National Science Foundation of China (No. 71301018) , National Science Foundation of China (No. 61701075) .

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] Liu K, Jin H, Chen J, et al. A compromised-time-cost scheduling algorithm in swindew-c for instance-intensive cost-constrained workflows on a cloud computing platform[J]. International Journal of High Performance Computing Applications, 2010, 24(4): 445-456.

[3] Chang R S, Chang J S, Lin P S. An ant algorithm for balanced job scheduling in grids[J]. Future Generation Computer Systems, 2009, 25(1): 20-27.

[4] Garg S K, Yeo C S, Anandasivam A, et al. Energy-efficient scheduling of HPC applications in cloud computing environments[J]. arXiv preprint arXiv:0909.1146, 2009.

[5] Yu J, Buyya R. Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms[J]. Scientific Programming, 2006, 14(3): 217-230.

[6] Wieczorek M, Prodan R, Fahringer T. Scheduling of scientific workflows in the ASKALON grid environment[J]. ACM SIGMOD Record, 2005, 34(3): 56-62.

[7] 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.