Grid Workflow Scheduling Based on Improved Genetic Algorithm

Xue Zhang
Cognitive Science Department, Xiamen University
Fujian Key Laboratory of the Brain-like Intelligent Systems (Xiamen University)
Xiamen, China, 361005
E-mail: cotgirl@126.com

Wenhua Zeng
School of Software, Xiamen University
Fujian Key Laboratory of the Brain-like Intelligent Systems (Xiamen University)
Xiamen, China, 361005

***Abstract*—Grid Workflow Scheduling represented by DAG(Directed Acyclic Graph) is a typical NP-complete problem, and thus a scheduling algorithm of high efficiency is required. So an improved genetic algorithm is proposed to solve this problem. In the algorithm, chromosomes of poor fitness make secondary preferential hybridization and mutation with the overall best individual. It not only guarantees the population diversity but increases the convergence rate of population. Experiment results based on Gridsim prove it available and better than standard genetic algorithm.**

Keywords-Grid workflow; scheduling problem; improved genetic algorithm; secondary preferential hybridization and mutation; Gridsim

I. INTRODUCTION

Grid technology is an important development direction of the Internet computing technology. As an important component of grid computing, Grid workflow is proposed to resolve problems encountered in Grid environment using workflow technology.

Grid workflow [1] is a synthesis of grid services; these grid services are performed on the heterogeneous and distributed grid nodes in a defined sequence, and ultimately complete a specific target. Task scheduling is an important research topic in grid workflow, which is different from the general task scheduling, grid workflow scheduling must consider not only choose a best resource for the task, but also consider the timing or causal constraints between the various tasks. Tasks and the mapping relations between the nodes compose grid workflow Scheduling Program.

This paper presents an improved genetic algorithm-based resource scheduling algorithm. In order to avoid slow convergence rate and easy to fall into local minimum, the algorithm add the process of secondary preferential hybridization and secondary preferential mutation. The algorithm has good convergence and efficiency.

II. DESCRIPTION OF THE PROBLEM

Grid workflow modeling let grid workflow-related tasks or workload operations make the appropriate model building in accordance with the implementation of the order between them, finally, the real problem is converted into the corresponding task sequences through the modeling language, and then converted into a programming language description through the relationship between the task sequences.

There are some common models: model based on ECA rules, transaction-based model, based on the DAG, based on the Petri Net [2]. In this paper, grid workflow modeling is established based on the DAG.

Figure 1 show a grid workflow instance which is described by a directed acyclic graph (DAG).

DAG workflow modeling is to describe the workflow as a series of sub-tasks. Dependency and priority relations between sub-tasks generate a directed acyclic graph G = (V, E, W), in which V represent all the nodes in the graph, that is, all sub-tasks of the workflow; E represent directed edges in the graph, that is, dependency and priority relations between tasks; W represent weights of directed edges, that is, the implementation costs required by task V_i.

Critical path is the path of the longest running time. In figure 1 we define the path with the biggest ΣW as the critical path. Here, we chose (V_1-V_3-V_5-V_6) as the critical path, chose (V_1-V_2-V_4-V_6) as Non-Critical Path. Therefore, the critical path exists ((V_1-V_3), (V_3-V_5), (V_5-V_6)) sub-tasks, Non-critical path exists ((V_1-V_2), (V_2-V_4), (V_4-V_6)) sub-tasks. Then we allocate fast-running resources to tasks on the critical path and allocate Low-costing resources to tasks on the Non-Critical path [3], thereby reducing the overall computing costs for grid workflow task scheduling.

The essence of the workflow scheduling is to assign appropriate sub-tasks of critical path or Non-Critical Path to a specific grid resource to conduct operations, which make the sum of the overall task computing cost lowest.

Suppose there are n sub-tasks T_i=(i=1,2,3,…,n) and m available grid resources R_j=($_i$=1,2,3,…,m). According to the priority and constraints relationship between the tasks, workflow scheduling allocate n sub-tasks to m available grid resources and make full use of grid resources, C_i is defined as completion time of job T_i, the goal of scheduling is to make ΣC_i minimum, thus reducing computing costs.

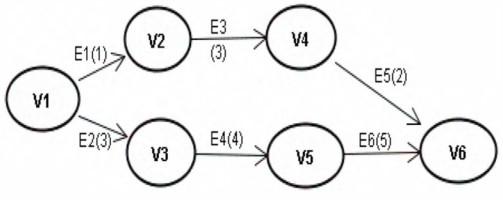

Figure 1. DAG-based workflow model

III. DESCRIPTION OF THE ALGORITHM

A. Genetic Algorithm

Genetic Algorithm [4] (Genetic Algorithm, Abbreviated as GA） is proposed by Professor Holland from Michigan University in the United States, 1975. It is a computing model that simulates natural selection of Darwin's evolution theory and biological evolution process of genetic mechanism. It is the optimal solution search methods by simulating natural evolution process, has a solid biological basis.

The main operating target of genetic algorithm is a group of chromosomes; each chromosome represents a scheduling solution. Therefore, priority scheduling problem can be transformed into demand for chromosome with the best genes. The actual issue solution transforms into chromosome by coding, then generals populations, selection, hybridization, mutation and finally get a chromosome with the best genes, which is the approximate optimal solution to scheduling problems.

B. Improved Genetic Algorithm Design

Task scheduling problem is a typical NP-complete problem. If we use the standard genetic algorithm scheduling, there are two disadvantages: easy to fall into local optimal solution and slow rate of convergence. To solve the above shortcomings, this paper improves the traditional genetic algorithm. New algorithm uses natural number to code, adds the processes of secondary preferential hybridization and secondary preferential mutation, and retains the best individual after breeding of each generation, effectively guaranteeing strong performance of global search and local search.

C. The specific design of improved algorithm

1) Coding Design

New algorithm encodes chromosomes with natural numbers [6], because it is better able to maintain the diversity of species than the binary-coded chromosomes, and it also enable encoding and decoding process simple.

Chromosome length is equal to the amount of assigned tasks, assuming that use $Y = (X_1, X_2, X_3 ... X_i ... X_n)$ to represent a chromosome. Each node of a chromosome can be defined as a two-tuple X_i (s_i, p_i) $(i = 1,2, ..., n)$, s_i represents resource that allocated to task i, and the task number i is unique, each task gets one resource, different tasks can get the same resource; pi is the priority value of the task.

2) Initialize population generation

In order to improve the coverage of individual stocks, using random initialization, that is randomly selected a resource to run the task i.

Specific initial steps are as follows: Suppose that population size is M, chromosome length is N.

```
From m to M
Begin
    From i to N
    Begin
        Select resource Xi randomly;
        Allocate Xi to task i;
    End
End
```

Through the above steps, will be able to complete the initial population generation, random selection of resources enables the population diversity and improves the ability of future breeding of generations.

3) Fitness function

Fitness function is used to assess the adaptability of current chromosomes, effectively reflect the gap between each chromosome and the optimal solution chromosome. The value of fitness function plays a very decisive role for solving the problem.

Build two-dimensional array of expected value Value=[T_i, R_j], Ti represents task i, R_j represents resource j, and calculate the corresponding theoretical expectations through the relationship between the assignments of Ti and the computing power of R_j.

In this paper, the fitness function depends on the expected completion time of all the task scheduling, and take countdown of the expected completion time of all the task scheduling as evaluation conditions, that is, take

$$1 / \left(\sum_{i=0}^{n} value(i, Y_i) \right) \tag{1}$$

Formula (1) as chromosome fitness evaluation, n is chromosome length; Y_i is the corresponding resource value of i-bits of the chromosome. Chromosome is better if the fitness is bigger; otherwise, chromosome is more inferior.

4) Selection

Selection operator determines which chromosomes get into the next generation. The algorithm used “roulette wheel” as selection method. The roulette wheel selection is to map collection elements to an interval, this interval is divided into several sub-segments (each segment corresponds to an element), resulting in n effective random numbers, statistics frequency that the random number falls in each segment, the element of segment with the highest frequency shall be selected.

The algorithm determines the selected probability of the chromosome based on the fitness value of chromosomes. If the fitness value of chromosome is greater, then probability of its being selected is greater. In order to achieve roulette wheel selection, each one round produces random numbers during [0, 1] interval as a reference probability. The probability of an individual r_i was selected is defined as follows:

$$p(r_i) = Fitness(C_i) / \sum_{j=1}^{pSize} Fitness(c_j) \tag{2}$$

pSize represents the population size.

After the selection probability of each chromosome has been determined, compare reference probability with the selection probability of chromosome, if the former is greater

than the latter, the chromosome is selected, otherwise be eliminated.

5) Crossover

Crossover changes partial structure of two parent individuals and combines them to a new individual. The new individual inherits characteristics from two parent individuals. So the problem can converge to the optimal direction.

In this algorithm, in order to guarantee the population diversity, crossover occurs in random location, and then a fitness value assessment is needed for the new individual. The algorithm makes appropriate adjustments based on the result of the assessment. Secondary preferential hybridization happens between new individual whose fitness value is lower than the average fitness value and the best individual of the population in order to improve the convergence speed of the solution space.

Concrete realization steps are as follows: the probability of crossover is R_s, generally ranges from 0.4 to 0.9[5]:

Select two individuals randomly;
Generate random numbers r;
If r >=R_s
Begin
Randomly selected crossover position pos;
The parent individuals 1 and parent 2 hybridize starting at the pos-bit;
Individual child 1 and individual child 2 generate after hybrid;
Begin
Judge Individual child 1 and individual child 2 separately;
If fitness value of Individual child < the average fitness value
Begin
Randomly selected crossover position pos1;
Individual child is replaced with the best individual starting at the pos1-bit;
End
End
End

6) Mutation

Mutation changes chromosomes with a small probability, in order to guarantee the chromosome diversity, and determine the local search ability of genetic algorithms.

The algorithm has a small probability of chromosomal mutation, and then a fitness value assessment is needed for the new individual. The algorithm makes secondary preferential mutation based on the result of the assessment. Single-point mutation happens between new individual whose fitness value is lower than the average fitness value and the best individual of the population in order to maintain the chromosome diversity and speed up the optimal reproductive capacity.

Concrete realization steps are as follows: the probability of mutation is R_c, generally ranges from 0.001 to 0.1[5]:

Get new individual (generated by step (4) selection operation);
Begin
Generate random numbers r;
If r >= R_c
Begin
Chromosomal variation in local areas;
If fitness value of Individual child after mutation< the average fitness value
Begin
The best individual value of the random bit assigned to a new Individual child;
End
End
End

7) Retain the global best individual

There is the best individual of the current reproduction after the end of each generation of breeding, which is an approximate optimal solution. Then compare the fitness value between the best individual of the current reproduction and the global best individual, if the best individual fitness of the current generation of reproduction is bigger than the overall best individual fitness, then the best individual replaces the global best individual, otherwise the overall best individual unchanged. And determine whether the reproduction meet the withdrawal conditions, if meet the withdrawal conditions, it stops breeding, otherwise continues breeding.

D. Algorithmic process

Through the above steps, it determines the operation relationship between the module and by way of flow chart to describe. The main process of improved algorithm is shown in Figure 2.

IV. RESULTS

This article uses Gridsim [7] tool for simulation in heterogeneous grid environment, and the improved genetic algorithm and standard genetic algorithm [5] are analyzed and compared in this article. Heterogeneous environment is simulated by some parameters such as machine, PE, MIPS. The main parameters: population size is 50, population number of iterations is 50, crossover probability is 0.8, and mutation probability is 0.01.

The experimental results are shown in Figure 3 and Figure 4 below. Figure 3 indicates the scheduling result while different tasks in 100 resources. Figure 4 indicates scheduling result while 100 tasks in different resources. The value of each coordinate point is equivalent to average value of 10 current consecutive scheduling.

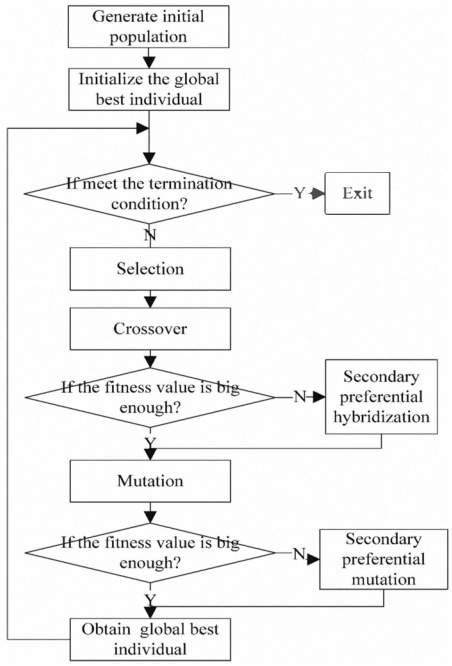

Figure 2. Improved genetic algorithm implementation of the process

Figure 3 shows simulation by changing the numbers of tasks in the case of retaining the numbers of grid resources. By comparing the experimental data, we can see the completion of improved genetic algorithm is shorter than the standard genetic algorithm; the costing of improved genetic algorithm is lower.

Figure 4 shows simulation by changing the numbers of grid resources in the case of retaining the numbers of tasks. By comparing the experimental data, we can see the completion of improved genetic algorithm is shorter than the standard genetic algorithm; the costing of improved genetic algorithm is lower.

Through two sets of experimental simulation above, we know that the costing of improved genetic algorithm is lower than traditional genetic algorithm, and thus verified the effectiveness of the improved algorithm.

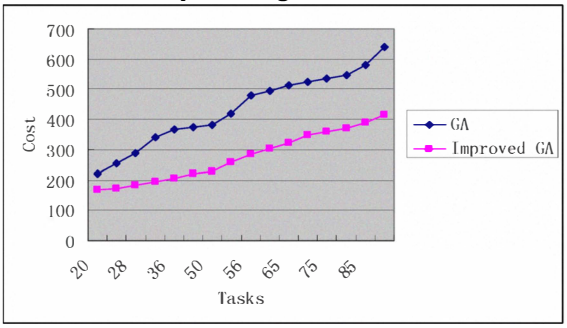

Figure 3. Different tasks schedule in 100 resources

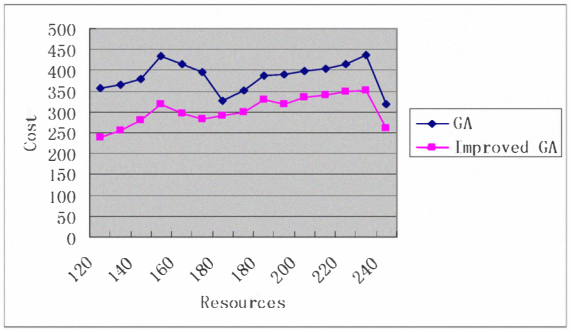

Figure 4. 100 tasks schedule in different resources

V. CONCLUSION

In this paper, improved genetic algorithm based on secondary preferential hybridization and secondary preferential mutation is proposed, the algorithm effectively improve the convergence speed of the solution space in the case of not changing population diversity. In the end, through the simulation, it verifies the algorithm is more efficient than the standard GA algorithm. Of course, there are more questions worthy of further research, such as how to solve the load balancing problem of grid workflow.

REFERENCES

[1] Jia Yu, Rajkumar Buyya. A Novel Architecture for Realizing Grid Workflow using Tup leSpaces[C]. Fifth IEEE /ACM International Workshop on Grid Computing, November 08 - 08, 2004, pages 119 – 128

[2] Hong Fen, Lu Wang, Hui Yang. Grid Workflow Technology and Its Application in Teacher Professional Development [J]. Audio-Visual Education Research, 2007

[3] Yuan Ying-Chun, Li Xiao-Ping, Wan Qian, Zhang Yi. Bottom Level Based Heuristic for Workflow Scheduling in Grids [J]. Computers, February, 2008

[4] M. Srinivas and L. M. Patnaik. Genetic Algorithms: Asurvey[C]. IEEE Comput. 27, June 1994:17_26

[5] Wang Xiao-ping, Cao Li-ming. Genetic Algorithm [M]. Xian: Xi'an Jiaotong University Press, 2002

[6] Wu Xiong-qi，Zeng Wen-hua. Grid Resource Scheduling Algorithm Based on Improved Genetic Algorithm [J]. Microelectronics and Computer, 2006, 23 (9) : 26-29

[7] Anthony Sulistio, Chen Shin Yeo, and Rajkumar Buyya. Visual Model for Grid Modeling and Simulation (GridSim) Toolkit[C]. ICCS 2003, LNCS 2659 :1123–1132. 2003