Quantum Genetic Algorithm for Scheduling Jobs on Computational Grids

Zan Mo, Guangwen Wu, Yunhui He, Hongwei Liu

Abstract—**The core of grid task scheduling model is scheduling algorithm. Generally, grid task scheduling algorithm is a traditional genetic algorithm. However, the traditional genetic algorithm has some shortcomings, such as converging too slowly and early maturity, which often cause decrease in efficiency and effectiveness of grid task scheduling. This paper takes grid task scheduling for research object, and builds a new grid task scheduling simulated system. By introducing the quantum genetic algorithm as the grid task scheduling algorithm, this paper uses grid task simulated platform to test and verify the new grid task scheduling model. Experimental results show the grid task scheduling model based on quantum genetic algorithm can increase grid efficiency and effectiveness significantly.**

I. Introduction

GIRD Computing is an emerging technology based on Internet. To provide computing power, geographically distributed resources (computing resources, storage resources, data resources) need to be logically coupled together to make them work as a virtual super computer based on sharing network [1] .Challenges arise in limited grid resources when grid applications are becoming increasingly widespread with continuous development of grid technology, namely how to make the most use of limited resources.

Task scheduling in Grid has been proved to be a NP-complete problem [2]. Heuristic optimization algorithm is the best approach to solve NP-complete problem, such as Genetic Algorithm (GA), Simulated Annealing (SA), Ant Colony Optimization (ACO) and search (TS).Time span in GA is superior to that in other heuristic algorithms when the task is on small scale. With enlargement of the scale, genetic algorithm gets local minimization too easily and converges slowly, the best chromosome is easily lost resulted in immature convergence [3].

With the deficiency in the traditional scheduling algorithm, this paper presents quantum genetic algorithm based on quantum algorithm, and applies it to grid task scheduling model. This paper uses the experimental simulated platform SimJava to test and verify the new grid task scheduling model, so as to prove that performance of the grid task scheduling under quantum genetic algorithm can be highly improved.

II. Grid Task Scheduling Model

In grid computing, task scheduling is essentially that n independent tasks are distributed to m heterogeneous resources that are available, in order to minimize the total time and fully utilize the limited resources [4]. As follows:

1) J is a set consisted of n tasks needed to be scheduled, J_i means task i .

2) R is a set consisted of m available resources, R_i means resource i .

3) $Execute$, a $n \times m$ matrix, is composed of executing time of n tasks operated on m different resources. $Execute(i, j)$ means executing time of task i operated on resource j .

4) $Communicate(i, j)$ is the transmission time when data required in task i is transmitted to resource j through storage system.

5) $makespan$ is the time needed to execute all the tasks, namely

$$makespan = Max\left\{Execute(i, j) + Communicate(i, j)\right\}$$

In this paper, grid simulated system which has been widely applied [5] is improved, as shown in Figure 1. The new grid simulated system includes task module, task analysis module, scheduling management module and resource management module.

Task Analysis Module: collect information of different tasks, including task completion time, dependencies among tasks, network delay, etc. Then, statistically analyze these data. Finally, transmit the result to scheduling management module.

Scheduling Management Module: select appropriate scheduling strategy, and then assign the tasks to various resources.

In this paper, through quantum genetic algorithm, task analysis module and scheduling management module schedule and manage the tasks and resources.

Manuscript received September 30, 2009. This work was supported in part by the Natural Science Foundation of Guangdong Province: No.06300278, and the Guangdong Philosophy and Social Sciences Planning Program: No. 7SJY020.

Zan Mo is with the Guangdong University of Technology, School of Management, Guangzhou, CO 510520 China (phone: 86-020-87083016; fax: 86-020-87083017; e-mail: mozan@126.com).

Guangwen Wu is with the Guangdong University of Technology, School of Management, Guangzhou, CO 510520 China (e-mail: wgwboss123@163.com).

Yunhui He is with the Guangdong University of Technology, School of Management, Guangzhou, CO 510520 China (e-mail: woniuhyh@163.com).

Hongwei Liu is with the Guangdong University of Technology, School of Management, Guangzhou, CO 510520 China (e-mail: hwliu@gdut.edu.cn).

978-0-7695-3962-1/10 $26.00 © 2010 IEEE
DOI 10.1109/ICMTMA.2010.505

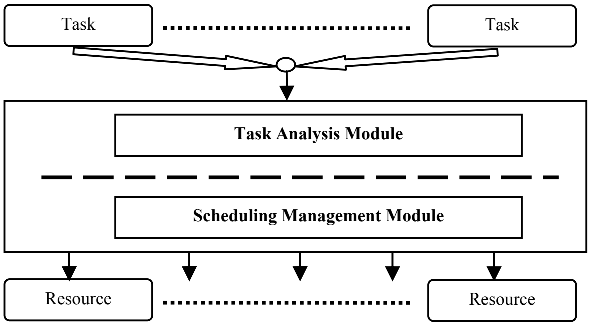

Fig.1. The Grid Simulated System

III. ANALYSIS OF QUANTUM GENETIC ALGORITHM

Genetic Algorithm (GA) is an adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic mechanism. To simulate the processes in natural system necessary for evolution, GA uses evolutionary techniques such as selection, crossover, and mutation to gradually approach the optimal solution from a set of feasible initial population. However, traditional genetic algorithm has some shortcomings including huge computing quantity, losing the best chromosomes, early maturing, cannot ensure obtaining the optimal solution. To address these issues, this paper presents a new genetic algorithm - Quantum Genetic Algorithm, which is based on the theory quantum computing and genetic algorithm.

A. Quantum Algorithm

Compared to classical computation, quantum computation is based on quantum mechanics rather than the idea of classical physics. The essential characteristics of quantum computation are quantum superposition states, quantum correlation and quantum entanglement.

A classical computer has a memory made up of bits, where each bit represents either a one or a zero. A quantum computer maintains a sequence of qubits. Quantum bits, called qubits, are implemented using quantum mechanical two state systems; they are not confined to their two basic states but can also exist in superpositions: effectively this means that qubit has two states and they are state 0 and state 1(typically written $|0\rangle$ and $|1\rangle$). Qubit state can be normalized to any state: $\alpha|0\rangle+\beta|1\rangle$, where α and β are complex numbers that satisfy the condition $|\alpha|^2+|\beta|^2=1$. Thus quantum computer can perform many different calculations in parallel: a system with n qubits can perform 2^n calculations at once! It has impact on the execution time and memory required in the process of computation and determines the efficiency of algorithms.

In quantum theory, the quantum state can be pure state or mixed state, the latter is statistical mixtures of the former. Consider the two noninteracting systems A and B , with respective Hilbert spaces H_A and H_B. The Hilbert space of the composite system is the tensor product $H_A \otimes H_B$.

If the first system is in state $|\phi_A\rangle$ and the second in state $|\phi_B\rangle$, the state of the composite system is $|\phi\rangle_A \otimes |\phi\rangle_B$.

States of the composite system which can be represented in this form are called separable states, otherwise are entangled states.

Not all states are entangled states. Fix a basis $\{|i\rangle_A\}$ for H_A and a basis $\{|i\rangle_B\}$ for H_B. The most general state in $H_A \otimes H_B$ is of the form $|\psi\rangle_{AB}=\sum_{i,j} c_{ij}|i\rangle_A \otimes |j\rangle_B$.

This state is separable if $c_{ij}=c_i^A c_j^B$, yielding $|\psi\rangle_A=\sum_i c_i^A|i\rangle_A$ and $|\phi\rangle_B=\sum_j c_j^B|j\rangle_B$. It is inseparable if $c_{ij}\neq c_i^A c_j^B$. If a state is inseparable, it is called an entangled state[6].

B. Design of Quantum Genetic Algorithm

Quantum Genetic Algorithm (typically written QGA) is a genetic algorithm based on quantum theory (such as quantum superposition and quantum entanglement). QGA uses qubit to code chromosomes, then introduces quantum superposition and quantum entanglement to genetic coding, obtains the evolution of chromosomes with quantum rotating gate.

1) Chromosome Coding: In QGA, chromosome coding adopts probability amplitude. These probability amplitudes have special significance because they act in quantum mechanics as the equivalent of conventional probabilities, with many analogous laws. Thus a quantum chromosome can express information in various states, chromosomes can be evolved more rapidly which enriches the population, so as to maintain the diversity of the group to overcome the premature [7].

Quantum chromosome population in the t generation is $Q(t)=\{q_1^t,q_2^t,q_3^t,...,q_n^t\}$, where n represents population size, t represents evolutionary generation, and chromosome q_n^t is defined as follows

$$q_n^t=\begin{bmatrix}\alpha_1^t & \alpha_2^t & ...... & \alpha_m^t\\ \beta_1^t & \beta_2^t & ...... & \beta_m^t\end{bmatrix}$$ （j=1,2,…, n ）, (m is the length of quantum chromosome)

Each chromosome is a collection of columns, and each column represents a scheduling scheme on particular processor. Thus, every chromosome is two-dimensional. Where one dimension is a certain index table of processor, and the other dimension is the task sequence of the processor scheduled.

2) Fitness Function: As the goal of grid task scheduling is shorten the time span as much as possible, fitness function can be defined as:

$$fitness(i,j) = 1/makespan(i,j)$$
$$= 1/Max\{Execute(i,j) + Communicate(i,j)\}$$

Means the fitness value of chromosome i is the reciprocal of its time-span. Smaller the time-span is, larger the fitness value is and more possible it is to be selected.

3) Algorithm Description: According to the principle of QGA, this paper constructs QGA flowchart as shown in Figure 2.

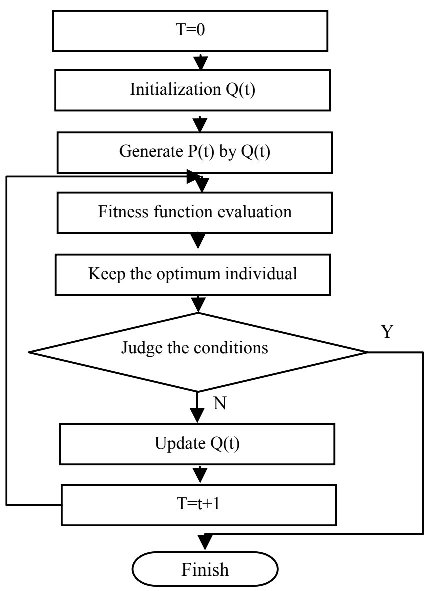

Fig.2. QGA flow chart

Initialization: firstly initialize evolutionary generation $T=0$, then initialize α_i^t, β_i^t in quantum chromosome $Q(t)$ and all q_n^t to be initialized to $\sqrt{2}/2$, namely all possible superposition states have the same probability.

Generate $P(t)$ by $Q(t)$: through the quantum collapse, generate $P(t)$ by $Q(t)$, where $P(t)=\{x_1^t, x_2^t, x_3^t, ..., x_n^t\}$, $x_j^t (j=1,2,...,m)$ breeded by $|\alpha_i^t|^2$ or $|\beta_i^t|^2$ is a serial ($x_1, x_2, x_3, ..., x_m$) with the length of m. At the same time, for each qubit, a binary value is computed according to its probabilities $|\alpha_i^t|^2$ and $|\beta_i^t|^2$. $|\alpha_i^t|^2$ and $|\beta_i^t|^2$ are interpreted as the probabilities to have respectively 0 or 1[8].

Use fitness function to evaluate the function: evaluate each individual of $P(t)$ and keep the optimum individual in this generation. If the optimum individual is a satisfactory solution, then terminate the algorithm, otherwise continue to update $Q(t)$.

Update $Q(t)$: obtain the variation of $Q(t)$ by the optimum quantum, that is , derive the guiding chromosome from the optimum individual obtained, then, randomly spread quantum chromosomes as the quantum population of the next generation around the guiding chromosome:

$$Q_{guide}(t) = a \times P_{currentbest}(t) + (1-a) \times (1 - P_{currentbest}(t))$$
$$Q(t+1) = Q_{guide}(t) + b \times (1 - normrnd(0,1))$$

Where, $P_{currentbest}(t)$ is the optimum individual in generation t, $Q_{guide}(t)$ is the guiding quantum chromosome, a is influencing index, b is the variance of quantum population randomly spreaded, and $a \in [0.1, 0.5], b \in [0.05, 0.15]$.

IV. THE GRID TASK SCHEDULING EXPERIMENT

For the sake of testing the performance of algorithm under random circumstance, this experiment uses platform SimJava to randomly get service parameters, including CPU, memory, etc. It adds a new task partition function QoS, that is, first divide and then arrange QoS.

Min-Min algorithm based on QoS, traditional genetic algorithm, and quantum genetic algorithm are compared in this experiment. Where, the proportion of QoS is $1:1:1$; in traditional genetic algorithm the number of population is 200 and evolutionary generation is 200; in quantum genetic algorithm the number of quantum chromosome is 10 and evolutionary generation is 200, moreover, every quantum chromosome can generate 5 ordinary chromosomes. Fitness function in quantum genetic algorithm is the same with that in traditional genetic algorithm, and the fitness function is

$$f = \frac{1}{\max\{Execute(T_i, a_i) + Communication(T_{ci}, a_{ij})\}}.$$

The result of 10 experiments is shown in table 1 as follows: *(in which the unit of total task time is simulated time unit in SimJava. In order to facilitate handling these data, it better to maintain 4 digits after decimal point.)*

From this table, we can say that in 10 trials, the overall performance of task scheduling in quantum genetic algorithm is superior to that in Min-min algorithm and in traditional genetic, which is shown as follows:

1) Grid task scheduling under QGA is less time-consuming and more effective: As shown in Figure 3, with the same number of tasks and severs, total time consumed of QGA grid task scheduling is 6317.7122 in 10 experiments, which is less than that(8209.5763) of Min-min and less than that(6947.1157) of traditional genetic algorithm. Moreover, each time consumed of QGA is apparently less than that of Min-min, and many times consumed of QGA are less than that of traditional genetic algorithm. Obviously, the grid task scheduling under QGA is less time-consuming and more effective.

TABLE I
CONTRAST TABLE OF PERFORMANCE

Total number of tasks			The number of nodes	Total execute time			Number of discarded tasks								
							Min-Min			GA			QGA		
High QoS	Middle QoS	Low QoS		Min-Min	GA	QGA	H	M	L	H	M	L	H	M	L
150	18	32	7	700.7138	666.7676	648.1350	0	1	1	0	1	0	0	0	0
146	28	26	7	883.3019	758.0812	631.0113	0	1	2	0	2	1	0	2	1
147	21	32	7	742.7956	634.7404	594.1643	0	0	0	0	0	0	0	0	0
132	34	34	7	633.3883	538.3040	573.2163	0	1	0	0	1	0	0	1	0
143	35	22	7	946.0059	739.1978	585.5538	0	0	1	0	0	0	0	0	0
134	28	38	7	877.1140	795.4615	740.2231	0	0	0	0	0	0	0	0	0
140	27	33	7	739.8798	735.7684	641.8545	0	0	1	0	0	1	0	0	0
143	29	28	7	941.5910	710.2904	635.6896	0	1	0	0	0	0	0	0	0
154	23	23	7	843.7254	651.7108	662.1863	0	1	1	0	1	0	0	1	0
143	30	27	7	901.0606	716.7936	605.6780	0	0	0	0	0	0	0	0	0

2) Grid task scheduling under QGA has less tasks discarded and is more authentic:

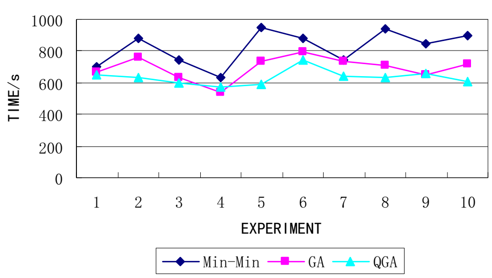

Fig.3 Grid task scheduling time of three algorithms

In 10 experiments, there are 5 discarded experiments in Min-min grid task scheduling under middle Qos and low Qos; in traditional genetic algorithm grid task scheduling, there are 4 discarded experiments under middle Qos and 3 discard times under low Qos; but in QGA grid task scheduling, there are only 3 discard times under middle Qos and 1 discard times under low Qos, as well as the number of discarded tasks is less. Therefore, the grid task scheduling model of QGA has less tasks discarded, has better stability and performance, and is more authentic.

V. CONCLUSION AND FUTURE WORK

In this paper, we have used the simulated grid platform Simjava to evaluate the new QGA scheduling algorithm. Experimental results show that grid task scheduling based on QGA outperforms that based on Min-min algorithm and traditional genetic algorithm. Furthermore, it also tolerates inaccurate execution estimation. Above all, grid task scheduling based on QGA not only shorten time span, but also increase task scheduling authenticity. Analytical and experimental results evince its great potential for grid task scheduling.

ACKNOWLEDGMENT

The paper is supported by the Natural Science Foundation of Guangdong Province: No.06300278, and the Guangdong Philosophy and Social Sciences Planning Program: No. 7SJY020.

REFERENCES

[1] *Foster I, Kesselman C. The grid: blueprint for a new computing infrastructure[M]. Morgan Kaufmann, 2004.*
[2] *Zhang Y F, Li Y L. Grid Computing Resource Management Scheduler Based on Evolution Algorithm[J]. Computer Project. 2003, 29(15): 110-111.*
[3] *Carretero J, Xhafa F. Use of Genetic Algorithms for Scheduling Jobs in Large Scale Grid Applications[J]. ūKIO TECHNOLOGINIS IR EKONOMINIS VYSTYMAS. 2006, 12(1): 11-17.*
[4] *Abraham A, Buyya R, Nath B. Nature's heuristics for scheduling jobs on computational grids[C]. 2000.*
[5] *Yan H, Shen X Q, Li X, et al. An improved ant algorithm for job scheduling in grid computing[C]. 2005.*
[6] *Jaeger G, Shimony A, Vaidman L . Two Interferometric Complementarities [J]. Phys. Rev. 1995, 51: 54.*
[7] *Cong B, Cheng Z R. Brief Analysis Contrast Between Genetic Quantum Algorithm and Genetic Algorithms in Resolving Extremal Problem of the Function [J]. JILIN NORMAL UNIVERSITY JOURNAL(NATURAL SCIENCE EDITION) . 2008, 29(001): 34-37.*
[8] *Yang S Y, Liu F, Jiao L C. A novel genetic algorithm based on the quantum chromosome [J]. JOURNAL OF XIDIAN UNIVERSITY(NATURAL SCIENCE) . 2004, 31(001): 76-81.*