An optimization scheduling model for multi-user software under cloud computing environment

WANG Xiang-gang

Vocational College of DongYing, Shandong 257091, China

Keywords: cloud computing; software; scheduling;

Abstract. The multi-user software scheduling optimization under cloud computing environments is studied. Under the cloud computing environment, multi-user scheduling software possessed characteristics like multi-objective, multi-constraint and dynamic, the traditional Particle Swarm Algorithm can only be utilized to schedule for the single-user software, therefore, multi-user software scheduling algorithm based on genetic - ant colony algorithm under cloud computing environment is proposed. To start with, a multi-user software scheduling mathematical model is established, then the genetic algorithm is introduced to quickly find a feasible solution for multi-user software scheduling, and finally the feasible solution obtained through the genetic algorithm is converted into Ant Colony Optimization (ACO) initial pheromone, and the optimal solution for multiuser software scheduling is acquired through local optimization and a positive feedback mechanism of ant colony algorithm. Simulation results show that the improved algorithm has not only the global optimization capability of genetic algorithm, also possess local optimization and the positive feedback capability of ant colony algorithm, compared to a single optimization algorithm, it can quickly find scheduling solutions to meet real-time requirements, augment scheduling speed, which favors scheduling for multi-user software reasonably and effectively.

1 Introduction

With the continuous development and popularization of computer processing technology, computer software scheduling has become a hot issue to be studied in computer field, and attracts widespread concern of many experts [1]. How to schedule software is the core of computer software scheduling. Therefore, the software scheduling method, has been paid a great deal of attention [2]. Currently, the main software scheduling method includes software scheduling method based on Particle Swarm Algorithm, software scheduling method based on adaptive algorithm and software scheduling method based on neural network algorithm [3-5]. Among them, the most commonly used method is software scheduling method based on Particle Swarm Algorithm. Since the software scheduling method has a very broad space for development, it attracts the attention of many experts, and has become a hot issue for everyone to consider, with great potential for development [6].

2 principle of the optimization scheduling method for multi-user software

2.1 The relevant principle of genetic algorithm

2.1.1 Individual coding

Chromosome coding is the key to solving for multi-user software scheduling. In order to take full account of the interdependence of the position information factors and various scheduling objects, a three-dimensional coding scheme is adopted to reduce the total execution time of multi-user software scheduling significantly, reduce system overhead, and the coding process is simple as follows:

\[
ST = \begin{bmatrix}
List_{t1} & List_{t2} & \cdots & List_{tp} \\
List_{t21} & List_{t22} & \cdots & List_{t2p} \\
\vdots & \vdots & \ddots & \vdots \\
List_{tn1} & List_{tn2} & \cdots & List_{tnp}
\end{bmatrix}
\]
Where, List\(_{ij}\) represents the sequence of performing scheduling tasks on software scheduling model \(P_{ij}\).

According to the above formula, through encoding for the individual of the population in multi-user scheduling software can have a good description of the multi-user software scheduling scheme.

2.1.2 The fitness function

The mainly consideration for multi-user scheduling software is the scheduling time span (Makespan), fitness calculated by fitness function should be non-negative, generally, larger fitness means better individuals. The task value of the individual should be as small as possible, thus, Makespan calculation need to be transformed necessarily to make it more suitable for fitness function, therefore, the maximum Makespan value is selected from the population, the value of \(2 * T_{\text{max}}\) minus Makespan of individuals [i] is regarded as a fitness function, and the fitness function \(Fit(i)\) of individual \(i\) is defined as:

\[
Fit(i) = 2 * T_{\text{max}} - \text{Makespan[i]}
\]

(2)

Where, \(T_{\text{max}}\) is the makespan value of the largest individual in population.

2.1.3 Initialization Population

Population initialization is a key for multi-user software scheduling, excellent early group populations can save computing time for solution, however, the traditional population is generated randomly, which leads to large probability for a large number of individuals focused on a local area, and easier to produce locally optimal solution. Therefore, the initial generation method need to be improved as follows:

Firstly, \(N\) individuals are generated randomly, the individual length is \(K\), and then the similarity of the two individual is compared, the similarity is defined as:

\[
\text{Similar}(i, j) < \frac{\text{Num}_{\text{same}}(ij)}{K}
\]

(3)

By comparing the similarity between individuals, the similarity of initial individuals which are eligible to be selected have to meet the following conditions:

\[
\text{Similar}(i, j) < \frac{K - c}{K} \quad i \neq j
\]

(4)

Wherein, \(c\) represents the adjustment constant, which used for controlling the similarity of expectation.

Therefore, by the above method to generate the initial population, can not only ensure obvious differences between each particle and the even distribution, also effectively reducing the probability of local optimum and improve the chance of finding the global optimal solution.

2.1.4 Genetic Operators

According to pre-set probability to choose part of optimum individuals for next generation of computing, crossover and mutation operation are performed to other individuals, in which a single-point crossover and random reverse mutation is able to generate new individuals. Reverse mutation is to conduct inverted operation for genetic value "0" and "1", to achieve chromosome mutation.

2.2 The relevant principles of ant colony algorithm

2.2.1 Setting the pheromone concentration of ants

Multi-user software scheduling feasible scheme is converted to initial pheromone concentration, specifically:

\[
\tau_{ij}(\alpha) = fit(x_i)
\]

(5)
2.2.2 Transition rules for ant colony state

After a period of time, ant colony have to choose the crawling node along with the pheromone concentration, at time t, the ant colony state transition rule of the ant i is:

\[
p_g(t) = \begin{cases} 
[r_g(t)]^\rho, & \text{rand}(0,1) \leq \rho \\
\sum_{j \in \text{allowed}} [r_g(t)]^\rho, & \text{Rand} \in \text{allowed}, \ \text{rand}(0,1) > \rho 
\end{cases}
\]

(6)

Wherein, \( \rho \) represents a random number between 0 and 1.

The changing of \( \rho \) is:

\[
\rho = 0.1 \times \frac{\log(9 \times t)}{e^{\alpha t}}
\]

(7)

Where, \( t \) and \( t_{\text{max}} \) denote the current generation and the largest generation of ant colony.

2.2.3 Global updating rules for the pheromone concentration

\[
\Delta r_g(t,t+1) = \begin{cases} 
Q, & \text{if ant } k \ \text{go through } i \ j \\
0, & \text{otherwise}
\end{cases}
\]

(8)

2.3 Steps for multi-user software optimization scheduling based on genetic - ant colony algorithm

During the multi-user software scheduling process, scheduling optimization steps are as follows:

Step1: Set initial parameters for ant colony, such as: population size, the biggest generation of evolution, selection, crossover and mutation probability and end conditions for the algorithm.

Step2: population initialization, and they are encoded.

Step3: Calculate the fitness function value of the individual based on fitness function.

Step4: selection, crossover and mutation for the individual to generate new individuals.

Step5: If the value meet the ends conditions of the iterative process, then jump to Step6, otherwise back to Step3.

Step6: The optimal solution for multi-user software scheduling portion obtained from the above iterative process is converted into pheromones initial value.

Step7: The ants are placed on the node of directed graph.

Step8: Depending on the state transition rule, every ants choose the next node of crawling.

Step9: According to the pheromone update formula, the pheromone concentrations for crawling paths of every ant is updated.

Step10: the fitness function of crawling path of each ant is calculated, and global updates are performed to its pheromone concentration.

Step11: if the algorithm terminates condition is meet, the optimal solution for multi-user software scheduling is output, otherwise jump back to Step8 to solve the optimal schedule.

The combined idea of two stages for multi-user software scheduling algorithm: Firstly, the feasible solution for multi-user software scheduling is searched quickly, and then converted into the initial pheromone, finally the globally optimal solution of multi-user software scheduling is obtained by a positive feedback mechanism, so as to accomplish multi-user software scheduling.

3. Simulation experiment

In order to verify the performance of multi-user software optimization scheduling method based on genetic - Performance ant colony algorithm, simulation testing is conducted on Matlab7.0 platform.

The experiments for multi-user software scheduling are conducted with conventional particle swarm optimization, adaptive algorithms, neural networks and genetic algorithms - ant colony
algorithm, respectively. The maximum number of iterations of particles is set as 500, the obtained result can be described by the following table:

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of iteration processing (times)</th>
<th>Scheduling Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Particle Swarm Algorithm</td>
<td>56</td>
<td>89</td>
</tr>
<tr>
<td>Adaptive Algorithm</td>
<td>64</td>
<td>78</td>
</tr>
<tr>
<td>Neural network algorithm</td>
<td>58</td>
<td>82</td>
</tr>
<tr>
<td>Genetic - Ant Colony</td>
<td>34</td>
<td>56</td>
</tr>
</tbody>
</table>

From the table, it can be known that the optimal solution obtained with genetic - ant colony optimization algorithm is superior to the other three traditional algorithms, mainly due to the design of the algorithm parameters is simple, global search capability is strong, and has a parallel search feature, reducing the probability of appearance of local optimal solution, convergence speed is faster significantly, which provide a better solution for the multi-user software scheduling optimization.

4. Conclusion

In order to solve the problem that the traditional Particle Swarm Algorithm can only be utilized to schedule for the single-user software, therefore, multi-user software scheduling algorithm based on genetic - ant colony algorithm under cloud computing environment is proposed in this paper. To start with, a multi-user software scheduling mathematical model is established, then the genetic algorithm is introduced to quickly find a feasible solution for multi-user software scheduling, and finally the feasible solution obtained through the genetic algorithm is converted into Ant Colony Optimization (ACO) initial pheromone, and the optimal solution for multiuser software scheduling is acquired through local optimization and a positive feedback mechanism of ant colony algorithm. Simulation results show that the improved algorithm has not only the global optimization capability of genetic algorithm, also possess local optimization and the positive feedback capability of ant colony algorithm, compared to a single optimization algorithm, it can quickly find scheduling solutions to meet real-time requirements, augment scheduling speed, which favors scheduling for multi-user software reasonably and effectively.

References