ATLANTIS PRESS

Advances in Computer Science Research, volume 63

International Conference on Artificial Intelligence and Engineering Applications (AIEA 2016)

An Improved Genetic Algorithm for Task Scheduling in Distributed Computing System

Shuhao Cui [a], Hua Zhang [b]

State Key Laboratory of networking and switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China.

[a]zhanghua_288@bupt.edu.cn, [b]csh9284@hotmail.com

Abstract. In distributed computing environment, task scheduling is one of the most important factors that affect the overall efficiency of the system. Task scheduling has been proved to be a NP-completeness problem, and the only way to solve this kind of problem is the method of exhaustion. Genetic algorithm is one of the best algorithms can solve the NP-completeness problem, which has the ability to quickly approach the optimal solution. However, there are still some shortcomings of genetic algorithm, such as the problem of premature convergence. In this paper, an improved genetic algorithm called genetic algorithm with geographical isolation is proposed. It restrains the excessive growth phenomenon of individuals by dividing the population, effectively solves the problem of premature convergence in genetic algorithm. In this paper, several sets of experiments are made to compare the operating efficiency and the final allocation effect of the algorithm and other scheduling algorithms, which prove that the algorithm can improve the efficiency of the algorithm without affecting the quality of the final solution.

Keywords: Distributed computing system; job scheduling; genetic algorithm.

1. Introduction

Due to the rise of high speed network, low network latency allows us to combine the use of computers distributed in different geographical locations, so that they do not have to consider the overhead of information exchange in the system for parallel computing. In this kind of environment, people begin to use the distributed computing system more and more widely. A distributed system is a model in which components located on networked computerscommunicate and coordinate their actions by passing messages [1]. The components interact with each other in order to achieve a common goal.

In distributed computing system, task scheduling is an important aspect that affects the efficiency of the whole system. How to allocate a large number of tasks to each processor, and to maximize the total utilization of the distributed computing environment, is the urgent problem to be solved in the distributed computing system.

Genetic algorithm is first proposed by Professor J. Holland [2], it is a common method to solve the task scheduling problem. Compared with other algorithms, it has the best performance [3]. However, the genetic algorithm still has the problem of premature convergence [4].This phenomenon can lead to low algorithm efficiency, still need to be further improved.

In this paper, an improved genetic algorithm called genetic algorithm with geographical isolation is proposed. This paper aimed at the requirements of the task scheduling module in distributed computing system, designed the coding and operation steps in detail, and proposed a method of dividing the population to suppress the excessive propagation of individuals in genetic algorithm. The algorithm successfully solved the problem of premature convergence and improves the convergence efficiency of genetic algorithm.

The remainder of this paper is organized as follows. In Section 2, the improved algorithm is proposed and the application and improvement of the algorithm are explained in detail. Section 3 describes the environment and architecture of the system and presents the experimental data which proves the advantage of my algorithm. Section 4 summarizes the whole paper.

2. Genetic Algorithm with Geographical Isolation

According to Building Block Hypothesis and Schema Theorem [5, 6], schema with short defining length, low rank, and high fitness value will exponentially grow under the action of genetic operators. This process is similar to the behavior of building blocks. Building Block Hypothesis and Schema Theorem proved that the genetic algorithm has the possibility to find the best solution and the ability to find the optimal sample.

However, the genetic algorithm still has the problem of premature convergence. After the first few generations of evolution, blocks with high rank, long distance and high fitness value are formed. These big blocks have strong competitiveness, resulting in excessive copying. Because big blocks are full of the population, small blocks are obsolete, the diversity of genes is reduced, premature convergence happens.

Genetic algorithm with geographical isolation is based on the concept of geographical isolation in the process of biological evolution. This algorithm can protect the genetic diversity and prevents excessive copying of big blocks by adopting an isolate operation. It has played an effective role in restraining premature phenomenon.

The steps of Genetic algorithm with geographical isolation are shown in Figure 1. On the basis of genetic algorithm, the population is divided into some sub populations. Selection operation and Crossover operation are executed individually. Each sub population cannot affect others. Every several generations, genes are shuffled and constitute new sub populations. Next, the operation steps are introduced.

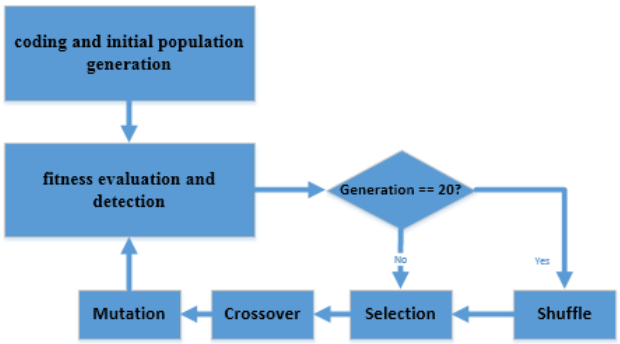

Fig. 1 Operation flow of genetic algorithm with geographical isolation

2.1 Coding and Initializing

In Genetic algorithm with geographical isolation, each gene is represented as a τ*1 vector, where position i ($0 \leq i < \tau$) represents task t_i, and the entry in position i is the machine to which the task has been mapped.

The genes in the initial population are randomly generated. In this paper, the scale of population is 30. After initializing, the main loop of Figure 1 is entered.

2.2 Fitness Evaluation and Detection

Fitness function will be introduced in section 3. In this paper, larger fitness value leads to a better change to be duplicated in selection operation.

2.3 Isolation

Isolation is executed every 20 generations. The algorithm randomly divided the whole population into 3 sub populations, and make marks on the genes. This operation enables selection and crossover to be performed independently.

2.4 Selection, Crossover and Mutation

Roulette wheel scheme [7] is used for selection, this scheme may duplicate, remain or delete each gene randomly, where gene has larger fitness value has a higher probability of being duplicated in the next generation.

Next, it put all the genes in one sub population in a pool. Then randomly selects a pair of genes from the pool, operate with probability of 60%. Crossover operation chooses a random point in the first gene, and two genes exchange the fragments and make up two new genes. Loop to perform this operation till the pool is empty. The operation of crossover is shown in Figure 2.

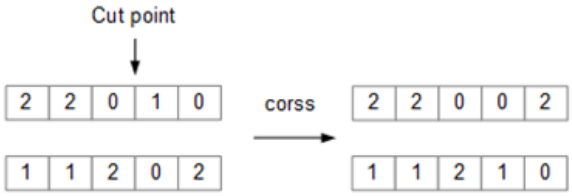

Fig.2 Crossover operation

After crossover, mutation operation is performed. Mutation operation randomly changes an assignment of a task with probability of 0.5%.

2.5 Termination Condition

Commonly, the algorithm stops when any of the 3 conditions is met, 1000 total iterations, population converge or the result reaches the expected value.

But in this paper, in order to observe the calculation trend of algorithm. The algorithm only stops after 1000 total iterations.

3. Simulation Model and Experimental Results

3.1 Simulation Model

In this paper, a distributed computing system is implemented. The whole system consists of a scheduler and five computer servers. The experiment begins with sending tasks. After receiving the tasks scheduler calculates the assignment with scheduling algorithm and send tasks to specific computers. When all the tasks are completed, the process ends.

In addition, the servers in this system are connected with the laboratory network to achieve the transmission speed of 1000Mbps. For kilobit level information, network latency is negligible, and it can meet the requirements of the experiment.

Assume that the amount of task to be performed is τ, the number of computers is μ, and the execution time of the task on each machine has been calculated. The input of tasks can be represent by a $\tau \times \mu$ matrix, called ETC (expected to compute) matrix. The time required to perform the task t_i on the computer m_j is

$$\text{time} = \text{ETC}(t_i, m_j), 0 \le i < \tau \quad 0 \le j < \mu \tag{1}$$

In the experiment, the value of τ is 10, and the value of μ is 100. The experience value will be used to construct the matrix. Due to the same task characteristics, the error between the experience value and the actual value can be controlled within the acceptable range.

Before the experiment, some parameters must be predefined. Mat (m_j) represents machine available time, which is the time when machine m_j can complete execute all the tasks assigned to it.

$$\text{mat}_j = \sum_{i=0}^{\tau} \text{ETC}(t_i, m_j) \quad 0 \le j < \mu \tag{2}$$

Dot (done time) represents the overall system running time, also equals to the maximum mat.

$$\text{dot} = \max(\text{mat}_j)\ 0 \le j < \mu \tag{3}$$

Fitness value of the algorithm is expressed as the reciprocal of dot. The faster tasks are finished, the better the solution is.

$$\text{fitness} = \frac{1}{\text{dot}_c} \tag{4}$$

3.2 Experimental Results

A distributed computing system is implemented based on the architecture designed in this section. Four algorithms are applied to the scheduling module. The four algorithms are genetic algorithm, genetic algorithm with geographical isolation, Max-Min and Min-Min. Max-Min and Min-Min are also common algorithms used for solving job scheduling problem, but simpler in terms of calculation and implementation.

The first experiment is used to compare the evolutionary speed of the two algorithms. In the line graph, the horizontal coordinate represents the generation, and the vertical coordinate indicates the

system completion time. Each point represents the completion time of the optimal solution of the whole population. In order to eliminate the error caused by the randomness, the two algorithms use the same initial population. The result of the experiment is shown in figure 3.

In figure 3, red line represents the process of evolution of classic genetic algorithm, and blue line represents the process of evolution of genetic algorithm with geographical isolation. Apparently, the genetic algorithm with geographical isolation can converge to the optimal solution earlier. It saves 30% of the computation time.

The advantages of genetic algorithm with geographical isolation are proved by this experiment. It can make the whole population maintain enough genetic diversity and continuous evolution ability. By comparison, classic genetic algorithm has a lot of obvious evolution platform, which is the result of low genetic diversity in the population.

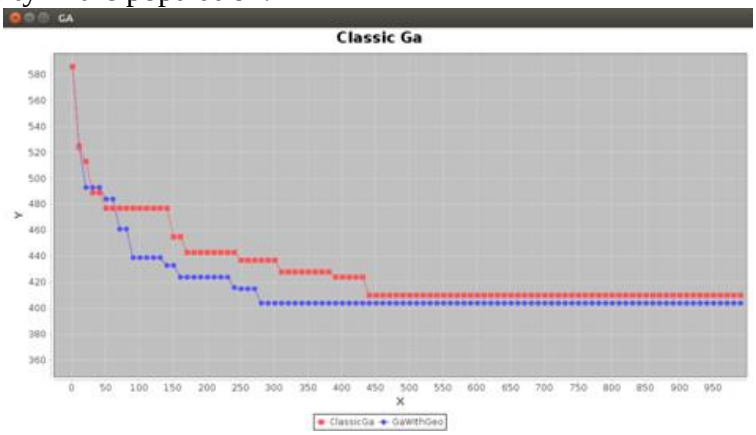

Fig.3 Speed comparison

In the second experiment, the scheduling ability of the four algorithms is compared. The same batch of tasks is send to the system for four times, each algorithm are used to schedule one of the four batches. The result is shown in figure 4.

The completion times of Max-Min algorithm, Min-Min algorithm, classic genetic algorithm and genetic algorithm with geographical are 432s, 440s, 408s and 404s. It can be seen that the scheduling strategy two genetic algorithms has similar results, far better than the former two algorithms.

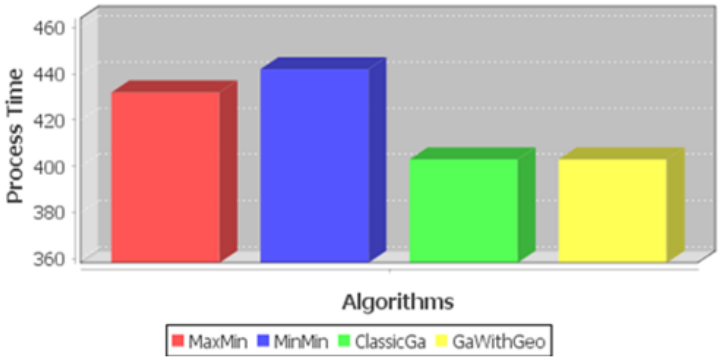

Fig. 4 System efficiency comparison

The running time of the four algorithms is compared with the third experiment. The result is shown in figure 5. Because of the simplicity of Max-Min algorithm and Min-Min algorithm, their running times are very short. However, due to the advantages of the allocation scheme, genetic algorithm is still a better algorithm.

In the comparison of two kinds of genetic algorithms, because the genetic algorithm with geographical isolation can calculate the optimal solution earlier, the algorithm takes less time. This algorithm can effectively save both computation time and computing resource.

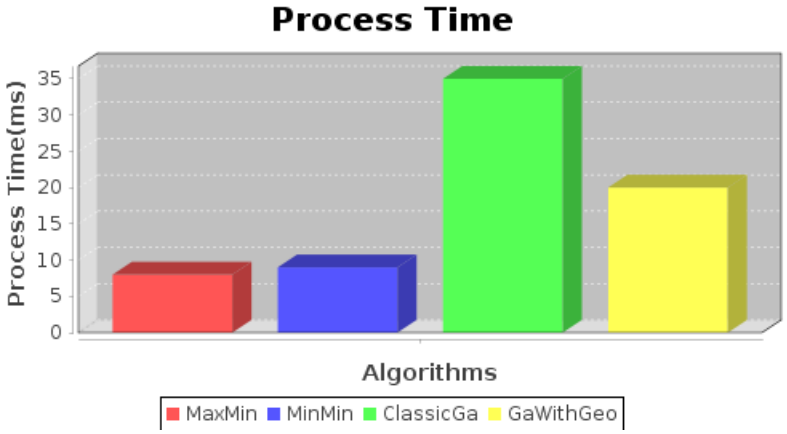

Fig.5 Execution time comparisons

4. Summary

In this paper, we mainly studied the task scheduling problem in distributed system, and select the genetic algorithm to solve it. Aiming at the premature problem of genetic algorithm, we proposed genetic algorithm with geographical isolation. The experimental results show that this algorithm can solve the problem of premature convergence which leads to a higher efficiency than the genetic algorithm in terms of the speed of the algorithm. Meanwhile, its task scheduling strategy is much better than other simple algorithms as the overall system running time is shorter.

Acknowledgements

This work is supported by NSFC (Grant Nos. 61300181, 61502044), the Fundamental Research Funds for the Central Universities (Grant No. 2015RC23).

References

[1] Coulouris G F, Dollimore J, Kindberg T. Distributed systems: concepts and design [M]. pearson education, 2005.

[2] Holland J H. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence [M]. U Michigan Press, 1975.

[3] Tracy D. Braun, Howard Jay Siegel, Noah Beck, A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems, Journal of Parallel and Distributed Computing, 2000, p. 810-837.

[4] Yu Z, Song S, Duan G. On the mechanism and convergence of genetic algorithm [J]. Control and Decision, 2005.

[5] Golberg D E. Genetic algorithms in search, optimization, and machine learning [J]. Addion wesley, 1989.

[6] Holland J H. Adaptation in Nature and Artificial Systems. MIT Press, 1992

[7] M. Srinivas and L. M. Patnaik, Genetic algorithms: A survey, IEEE Comput. 1994