

Research on Matching Mechanism of Resource Based on Genetic Algorithm

Rui Gu*

Depart of Information Engineering, Suzhou Industrial Park Service Outsourcing, China

*Corresponding author

Abstract—Resource matching is designed to make use of limited resources to optimize the processing task. In this paper, genetic algorithm is adopted to optimize the matching of resources. First, the principle of genetic algorithm and resource matching are introduced. Then, the detailed steps of genetic algorithm to solve resource matching are given. Finally, the simulation program is written by eclipse. By comparing with the prediction based resource matching algorithm, which is proved that the genetic algorithm has better superiority in time span and resource utilization ratio.

Keywords—resource matching; genetic algorithm

I. INTRODUCTION

Machine learning is aimed at studying the use of computer to simulate human learning activities. It is a way to study computer recognition of existing knowledge, acquire new knowledge, improve performance and achieve self-perfection. The development of machine learning is very rapid, and the application field is becoming wider and wider, and many excellent algorithms have appeared in this process. The most representative of them is the genetic algorithm.

Grid technology mainly utilizes the heterogeneous resources of the geographically distributed resources to achieve resource sharing. Grid task scheduling is based on the task requirements to schedule tasks to the appropriate resources to achieve the purpose of the task. The traditional resource matching algorithm mainly based on the user's resource description on the grid resource traversal to achieve matching and coordination of resources, as far as possible the comprehensive performance of the target system in similar resource nodes, in order to reduce the scheduling when selecting a processing unit of the search space, so as to shorten the running time of the whole scheduling.

This paper mainly focuses on the resource matching algorithm based on machine learning. The application of machine learning strategies to resource matching makes the time and resource utilization rate of each task can be as close or reach the optimal solution, so as to minimize the time of task running and improve the resource utilization rate of the system.

II. THE CONCEPT AND PRINCIPLE OF GENETIC ALGORITHM

Genetic algorithm originated from computer simulation research of biological system. It is a random search algorithm based on natural selection and genetic mechanism of biological boundary, and its basic flow chart is shown in Figure I. The

algorithm abandons the traditional way of searching the optimal solution. The algorithm abandons the traditional search optimal solution. It regards each solution in the target solution space as a chromosome of the group. And each chromosome encoding string form, genetic manipulation of the chromosomes in the group based on the unit. The chromosomes are evaluated according to the predetermined fitness function, so that the individuals with high fitness are more likely to reproduce than those with low fitness. The group can converge to the optimal solution by the iterative genetic manipulation of the generation to generation.

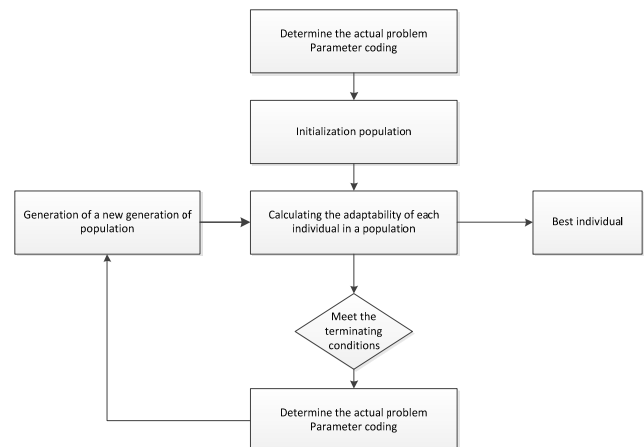


FIGURE I. GENETIC ALGORITHM FLOW

III. RESOURCE MATCHING MODEL

A. Problem Description

The resources mentioned in this paper mainly refer to computing resources, which may encounter various complicated situations in the research of actual grid scheduling strategy and also need to deal with many parameters. In general, the assumption of simplification is given first.

- A large application task has been decomposed into a number of subtasks with certain constraints, without mutual preemptive execution
- Do not consider transmission delays between subtasks.

We give the general initial conditions for the problem as follows:

- The task set $T = \{ T_1, T_2, \dots, T_n \}$, is a collection of N subtasks, where T_i represents the sub task of the i .
- The resource set $R = \{ R_1, R_2, \dots, R_m \}$, is a collection of M computing resources, where R_j represents the j resource.

B. Related Definitions

- Definition of task time span:

$$Makespan(x) = \text{Max}\{Fin(T_i), i = 1, 2, \dots, n\} \quad (1)$$

$Makespan(x)$ represents the time span of a scheduling x , which is the time to perform all tasks, and $Fin(T_i)$ is the time for completion all tasks, which is the span of the whole task processing.

- Definition of resource utilization ratio

$$\text{Resource_}U(T_i) = \text{Cost}(T_i) / \text{Budget}(T_i) \quad (2)$$

$\text{Resource_}U(T_i)$ is The resource utilization ratio, $\text{Budget}(T_i)$ Size of budgetary resources, $\text{Budget}(T_i)$ the size of the actual cost of resources. The resource utilization ratio completed by the whole task is:

$$\text{Resource_}U(x) = \frac{\sum_{i=1}^n \text{Resource_}U(T_i)}{n} \quad (3)$$

IV. RESOURCE MATCHING DESIGN BASED ON GENETIC ALGORITHM

A. Chromosome Design

The design of chromosomes is the key to genetic algorithms, and binary coding is usually used. However, binary coding has some shortcomings. For example, when mutation occurs, the bit weights of binary alleles are different, which leads to the mutation of incremental solution after the mutation. The result may lead to the omission of individuals closest to the best. In this paper, we use the floating point coding method. The chromosome length is the number of tasks in the task queue. The number of each locus represents the number of the corresponding sub task, and the gene value represents the resource number allocated to the sub task. The specific expression of the chromosome is:

$$X_i^t = [x_{i1}^t, x_{i2}^t, x_{i3}^t, \dots, x_{in}^t] i = 1, 2, 3, \dots, \text{pop_size}$$

Among them, x_{ik}^t for floating point variables, the superscript t is evolutionary algebra. The initial generation is 0. The subscript is the number of the individual, representing the chromosome number in the first generation, which is the size of the population.

B. Initial Population Generation

The size of the group directly affects the rate of convergence and the production of the optimal solution. The population is too large, it will reduce the speed of convergence; and the population is too small, it is difficult to produce the optimal solution. The method of generating the initial population here is to divide n sub tasks randomly into $h+1$ subsets. $T(i)$, $0 \leq i \leq h$, h is the largest number of subsets. For each subset $T(i)$, all tasks are assigned to m resources randomly.

Because all tasks in $T(i)$ can be executed in parallel, if we can assign them relatively evenly to different resources, we can improve the parallelism of the whole task and shorten the total execution time of the task. Set $T(i) = p$, then the number of tasks assigned on the resource I is:

$$ti = \mu + \delta \quad (4)$$

$1 \leq i \leq m$, $\mu = p/m$ is the average number of tasks, δ for the random Delta

C. Fitness Function

The selection of fitness function is very important, which directly affects the convergence speed of genetic algorithm and whether find the optimal solution. In general, the fitness function is converted from the objective function. In order to select a good chromosome, the fitness function is defined as follows:

$$f(x) = \frac{\text{Resource_}U(x)}{\text{Makespan}(x)} \quad (5)$$

x representing a scheduling, $\text{Resource_}U(x)$ the utilization of resources after completion of the whole task. $\text{Makespan}(x)$ the time span of chromosomes.

D. Selection Operation

The selection mechanism is used to form the next generation of solution groups after the hybridization and mutation operation is completed, the new group and the operator before operation are obtained according to the operator. In this paper, we adopt the best individual protection strategy to copy the individuals with the highest fitness in the population to the next generation without genetic operation, so as to prevent the best individuals from being damaged or to mix the parent and offspring populations, and to sort them in descending order according to their fitness. And the pop_size individuals in front of the population are intercepted into the mating pool according to the population size of pop_size, so that the survival of the fittest can be achieved better. The specific operation steps are as follows:

- 1) Calculation of the fitness value of each chromosome $f(x)$.

2) Selecting the best individual from the current population to copy to the next generation according to the size of the adaptive value.

3) Select the pop_size-1 chromosome through the "roulette" method to enter the next step of genetic manipulation. The selection probability of each chromosome P_i , $i=1, 2, 3, \dots, pop_size-1$.

E. Cross Operation

For cross operation, the "inbreeding" can be avoided by controlling the Hamming distance of the cross individuals. In the later stage of evolution, the population is close to homogeneous population, so it is difficult to generate new offspring. Controlling Hamming distance helps to maintain diversity and improve the efficiency of evolution. In order to ensure the steady improvement of population fitness, we can carry out 2/4 survival competition selection among two cross individuals, and two of the four individuals enter the next operation and the other two discard, so that the offspring fitness will not deteriorate. Set the T generation Hamming control distance as $H_t=H_0/t$, H_0 is the average Hamming distance of the initial population. The specific calculation formula is as follows:

$$H_0 = \frac{\sum_{i=1}^{pop_size-1} \sum_{j=i+1}^{pop_size} H_{ij}}{\sum_{i=1}^{pop_size} pop_size - i} \quad (6)$$

$$H_{ij} = \sqrt{\sum_{k=1}^{pop_size} (x_{ik} - x_{jk})^2} \quad (7)$$

H_{ij} is the Hamming distance between the X_i and the X_j , x_{ik} , x_{jk} are the value of the K Gene on the chromosomes i and j respectively. pop_size is the number of variables. Select two cross individuals X_i , X_j and calculate of their Hamming distance. If $H_{ij} \geq H_t$, cross operation is allowed. When $H_{ij} \geq H_t$, cross operation is allowed. If $H_{ij} < H_t$, compare the good and bad of the individual, eliminate the inferior individual and choose a new individual to judge. X'_i , X'_j are produced by a mixed cross method for the two chromosomes that meet the requirements.

F. Cross Operation

Before the mutation, the optimal individual protection measures are directly copied into the next generation by using the individuals with the highest fitness in the cross population. For each chromosome, a random choice is selected and a new resource number is produced. The survival and competition mechanism of pre - and post - variant individuals is introduced, with high adaptation to the survival and the next generation, and the other is eliminated. The formula for calculating the variation rate of P_m in this paper is as follows:

$$P_m = \alpha + NG2g\beta \quad (9)$$

α , β are very small constant. NG is the number of iterations. When NG is hourly, P_m very small, the effect of variation can be ignored.

G. Cross Operation

The evolutionary termination condition set in this paper is that the optimal individual adaptation of evolutionary algebra to 50 or successive m generations is less than 0.001, and evolution ends. M is one of the parameters set.

V. SIMULATION EXPERIMENT

A. Experimental Process

The experimental program is written in Java language, and 3 classes are mainly defined in the program. The first Resource class represents the resources in the grid computing, which contains the attributes of the number of resources, the memory of the resource, the peak load, and so on. The second task classes represent the tasks to be handled, which contains a number of tasks, task size, task delay, task processing strategies and other attributes, the main function is Resource matching. The main task is to randomly assign tasks to each resource in the task list, through the genetic algorithm from the selected task group and the best resources the matching process. The third class is the Utils tool class, which is responsible for the sending of user tasks and the acceptance of the resource machine task.

B. Experimental Results and Analysis

There are many criteria for measuring a resource matching algorithm, and there is no doubt that space and time are the most important among the many factors. Therefore, this paper selects the resource utilization ratio and the completion time of the whole task as the standard, and compares the resource matching algorithm based on the prediction. The number of tasks is 10-60, the size of the initial population was 50, the crossover probability is set to 0.5, the mutation probability is calculated as the formula 9, the alpha value of 0.004, the beta value of 0.001, the number of iterations for 50 times, the 3 consecutive generation of optimal conditions of termination of individual fitness change rate is less than 0.001, then the termination. The results of the experiment are as follows:

TABLE I. EXPERIMENTAL RESULT

Number of tasks	Matching based on genetic algorithm		Prediction based on matching	
	Makespan	Resource_U	Makespan	Resource_U
10	0.13	0.51	0.21	48
20	0.26	0.48	0.40	0.36
40	0.59	0.40	0.78	0.29
60	0.88	0.35	1.214	0.24

The above experimental data show that with the increasing number of tasks, the time span of resource matching based on genetic algorithm is increasing and resource utilization rate is decreasing. The time span of the prediction based resource matching algorithm is increasing, and the utilization of resources is decreasing. Resource matching based on genetic algorithms is smaller than prediction based resources on time span. Resource matching in genetic algorithms is higher in resource utilization than based on prediction based resources. This gap is widening as the number of tasks continues to increase.

The experimental results show that the genetic algorithm based resource matching is superior to the prediction based resource matching algorithm in time span and resource utilization.

VI. CONCLUSION

In this paper, a resource matching based on genetic algorithm is proposed, and the optimal resource set of the processing task is found through the iterative process. It not only changes the blindness of resource search and shortens the time of task processing, but also changes the status of resource waste in the process of task processing, and improves the resource utilization and throughput of the system.

In the process of designing genetic algorithms for resource matching, the priority of task processing is not taken into account. This defect has an impact on the timeliness of the system, and does not give priority to a number of tasks that need to be given priority or emergency treatment. I will continue to improve the priority of task handling in the later work.

REFERENCES

- [1] Garlan D, Monroe R, Wile D. Acme: an architecture description interchange language [C]. In: CASCON First Decade High Impact Papers, IBM, 2010: 159-173.
- [2] Jianzhong L, Srivastava J. Efficient aggregation algorithms for compressed data warehouses[J]. IEEE Trans on Knowledge and Data Engineering, 2002, 14(3): 515-529.
- [3] Zar Ni Mg. Concept hierarchy-based cube aggregation for ETL process in matriculation warehouse[C]. ICCOMP'10 Proceedings of the 14th WSEAS international conference on Computers. Stevens Point, Wisconsin, USA : WSEAS, 2010. 510-515
- [4] Missaoui R, Goutte C, Choupo AK, et al. A probabilistic model for data cube compression and query approximation[C]. DOLAP'07. New York, USA: ACM 2007. 33-40
- [5] Xi JQ, Chen FQ, Zhang PJ. A New Bitmap Index and a New Data Cube Compression Technology[C]. ICCSA '08. Berlin, Heidelberg: Springer-Verlag, 2008. 1218-1228
- [6] Dallen Q, Jennifer W. On-line Warehouse View Maintenance[C]. SIGMOD'97. New York, USA: ACM, 1997. 393-404