Research on Resource Scheduling based on Improved Mutation Operator in Cloud Computing

Junwei Ge[1, a], Fangfang Sun[1, b] and Yiqiu FANG[2, c]

[1]School of Software Engineering, Chongqing University of Posts and Telecommunications,Chongqing 400065, China;

[2]School of Computer science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China.

[a]gejw@cqupt.edu.cn, [b]sunfangfang12@126.com, [c]fangyq@cqupt.edu.cn

Keywords: Cloud computing, task scheduling, mutation operator ,Genetic algorithm.

Abstract. With the increasing development of cloud computing,task scheduling problem become a crucial aspect.There are many scholars research on resource scheduling problems,but load balancing and scheduling in how to balance time and cost is still not achieved satisfactory results.Genetic algorithm(GA) has strong robustness and inherent parallel computer system,particularly suitable for the combinatorial optimization problems, but the basic genetic algorithm easy to fall into local optimal solution. So ,we propose a crossover-mutation operator in mutation to solve this problem. The performance is analyzed using Cloudsim simulator and compared with existing GA ,Min-min algorithm. Simulation results demonstrate that the proposed algorithm has better performance in load balancing ,finish time and costs.

Introduction

Cloud computing is a new style of computing which is getting more important in our daily life.

Cloud computing links a large number of resources, software and IT infrastructure as a resource pool for outside services.The continuous development of cloud computing results in great diversification of user requirements and huge complexity of applications, which puts forward higher requests for resource allocation, load balancing and scheduling management[1].The high scalability of cloud computing also needs the resource management system to support varieties of heterogeneous resources[3]. The low cost of cloud computing services needs plenty of cheap PCs to be organized and needs efficient resource scheduling policy. Intelligent optimization algorithm such as GA is very suitable for application in cloud computing resource scheduling.

Genetic algorithm. The basic idea of genetic algorithm[4]was derived from Darwin theory of evolution and Mendel theory of heredity,inspired by Holland in1975,which was inspired by the theory of biological evolution. As an NP problem,task scheduling in cloud computing is difficult to be solved by means of classic algorithms. Genetic algorithm, because of its global convergence in solving multi-objective Optimization problems, has excellent results. It has strong robustness and inherent parallel computer system, particularly suitable for the complex multi-space extreme optimization and combinatorial optimization problems[5,6], but the basic genetic algorithm has premature convergence phenomenon, and is easy to fall into local optimal solution. So in this paper we put forward an Improved GA algorithm[2].We use crossover-mutation operator variation strategy to enhance the local search ability and overcome the traditional GA shortcoming .

Task scheduling model in Cloud Computing

$Ect(i, j)$ represents the execution time of the task i on the virtual machine j.

$w(t)$ represents the cost of using cloud computing resources per time.

Execution time for each virtual machine:

$$Vm(j) = \sum_{i=1}^{n} Ect(i, j) \tag{1}$$

As the cloud computing tasks are executed in parallel, so the task finish time is the most time consuming virtual machine:

$$RunTime(I) = \max_{j=1}^{k} (Vm(j)) \tag{2}$$

The total cost is the completion time plus the unit cost of VM:

$$Cost(I) = \sum_{j=1}^{k} \sum_{i=1}^{m} Ect(i, j) * w(t) \tag{3}$$

Load balancing of the tasks

$$LB = \sqrt{\sum_{i=1}^{k} \frac{(\overline{B} - B_i)^2}{m}} \tag{4}$$

Improved Genetic algorithm

Fitness Function. The selection of fitness function is very important, which affects the convergence of the algorithm and load balancing directly. In this paper,we consider the execution time ,cost and load balancing comprehensively. Defined the fitness functions as follows:

$$f(t) = \alpha \frac{1}{RunTime(t)} + \beta \frac{1}{Cost(t)} \tag{5}$$

$$\alpha + \beta = 1$$

Roulette Selection. When choosing a selection operator, elite strategy is used to reserve a few of the optimal chromosome. Then according to the fitness of chromosome adopts the roulette selection,the optimal chromosomes are more likely to be selected. The probability of selection is:

$$P(k_i) = \frac{f(k_i)}{\sum_{j=1}^{n} f(k_j)} \tag{6}$$

Crossover Operator. In the crossover operation ,we choose partially mapped crossover. Each chromosome fragment of two parent chromosomes sequentially combined into set A and B,according to the variation point ,keep the genes between corresponding variation ,delete duplicate genes in A,B and fill in the blank spaces in sequence,produce a new collection C,D.According to the division chromosome fragments, C,D back into the new offspring chromosomes.

Mutation Strategy. Mutation operator determines the local search ability of genetic algorithm. In order to maintain the population diversity and avoid premature in the early stages ,you need to use a larger variation factor.

According to the above load balancing model and coding design, we can use exchange two gene location methods. When the mutation probability is high , indiscriminate variation will lead to poor population fitness. So, we put forward the following crossover-mutation operator:

Random select of two chromosome segments, exchange of two genes which random select in any location of the selected chromosome segments.The variation not only ensure the effectiveness but also guarantee the global consistency.

Select two chromosome fragment to merge randomly, division of points randomly, two new chromosome segments were produced according to the segmentation point. When the fitness of the sub generation of the chromosomes is better than the old one , save the child chromosome, otherwise not variation.

Termination Conditions. Termination conditions for iteration are mainly in two ways. The first is the iteration number is more than the maximum number of iterations, the iteration is naturally terminated. The second is the best fitness value of global optimal chromosome meet certain requirement:

$$abs(\kappa - 1/F_{opt})/\kappa < \varepsilon$$

```
Algorithm steps  of Improved GA
BEGIN
1. Initialize population by random solutions.
2. Evaluate each candidate by fitness formula.
3. While (not termination condition)
      Select operator
       Mutate  the resulting  offspring
       If $\theta$ is TRUE
            The iteration number is an odd number
                  Running two-variation
         Evaluate new candidate
 4. Select individuals for next generation
END
```

Experiment and Simulation

This experiment adopts the cloudsim platform, rewrite Classes,such as DatacenterBroker, Cloudlet. The proposed algorithm did a large number of experimental simulation. In the experiments,the algorithm implemented in Java language. Compared with Min-min and standard genetic algorithm,under the same parameters.

The initial conditions	
VM numbers: 16	population size: 200
Pc: 0.05	Pm: 0.85
θ: TRUE	

Comparing three batch of tasks with execution time ,load balancing and cost. The comparison results are shown below.

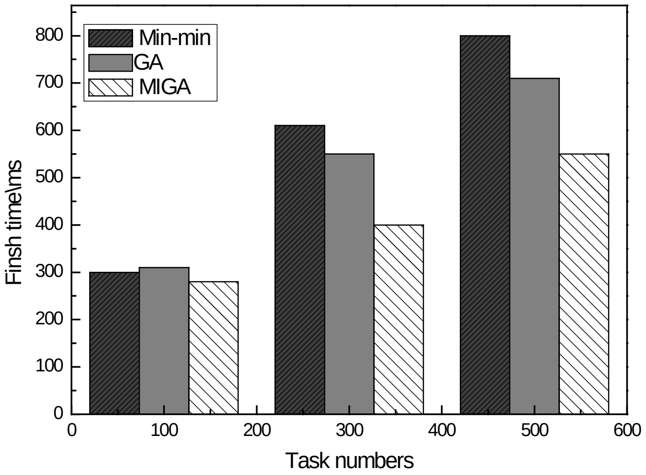

Fig. 1 Finish time

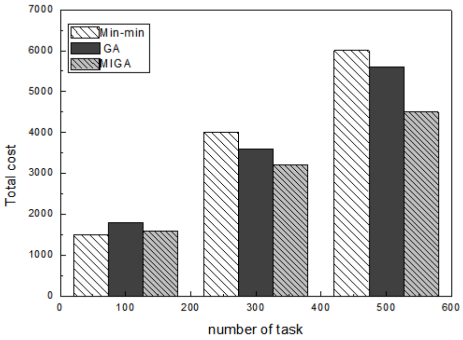

Fig. 2 Total cost

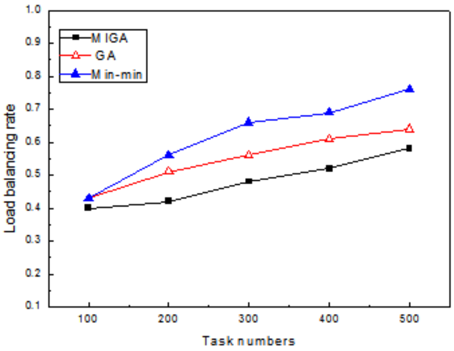

Fig.3 Load balancing rate

Summary

As can be see form above figures ,Min-min algorithm will be better than GA in the small and short tasks ,because GA is relatively complicated in such scheduling environment. As the long tasks ,more batch tasks scheduling, the advantages of GA and MIGA are obviously can be seen .Figure 2 shows the Total cost of MIGA is 15% lower the Min-min and 9% lower than GA.Fig.3 shows the load-balancing rate is 24% lower than Min-min. The improved GA algorithm is superior to the other two methods not only in cost or time ,but also in load balancing.

The future research will focus on considering the dependencies of tasks as well as the price of different virtual machine in a more complex cloud environment.

References

[1] James J, Verma B. Efficient VM load balancing algorithm for a cloud computing environment [J]. International Journal on Computer Science & Engineering, 2012, 4(9). 1658-1663.

[2] Epitropakis M G,Tasoulis D K. Enhancing differential evolution utilizing proximity-based mutation operators[J].IEEE Transactions on Evolutionary Computation 2011,15（1）.

[3] Drira K, Kacem A H, Jmaiel M. Towards the optimal synchronization granularity for dynamic scheduling of pipelined computations on heterogeneous computing systems[J]. Concurrency & Computation Practice & Experience, 2012, 24(18):2302-2327.

[4] Chen S, Wu J, Lu Z. A Cloud Computing Resource Scheduling Policy Based on Genetic Algorithm with Multiple Fitness[C]// Computer and Information Technology, International Conference on. IEEE, 2012:177-184.

[5] Jianfeng Li, Peng Peng. Task scheduling algorithm based on Improved Genetic Algorithm in cloud computing environment [J]. computer application, 2011, 31 (1): 184-186.

[6] Xiong Congcong, Feng Long, Chen Lixian, et al. Research on task scheduling algorithm based on Genetic Algorithm in cloud computing [J]. Journal of Huazhong University of Science and Technology: Natural Science Edition, 2012 (S1): 1-4.