Task Scheduling Using Parallel Genetic Simulated Annealing Algorithm

Shijue Zheng Wanneng Shu Li Gao

***Abstract*—Task scheduling is a NP-hard problem and is an integral part of parallel and distributed computing. This paper combined with the advantages of genetic algorithm and simulated annealing, brings forward a parallel genetic simulated annealing algorithm and applied to solve task scheduling in grid computing. It first generates a new group of individuals through genetic operation such as reproduction, crossover, mutation, etc, and than simulated anneals independently all the generated individuals respectively. When the temperature in the process of cooling no longer falls, the result is the optimal solution on the whole. From the analysis and experiment result, it is concluded that this algorithm is superior to genetic algorithm and simulated annealing.**

***Index Terms*—Grid computing, task scheduling, genetic algorithm, simulated annealing, PGSAA algorithm**

I. INTRODUCTION

Grid computing is a hot topic in the current internet research, and a developing direction of the parallel and distributed process. Resource management and task scheduling are the key problems in grid computing as in traditional distributed computing. Grid computing is made up of large instruments through LAN, which offers collateral computing system. The power of computing nodes is much stronger than that of traditional one, yet the cost of communication between nodes is much higher than of the traditional one. So that grid computing environment could be regarded as a computing pool, which is suitable to deal with collateral task between which there is no communication. The computing pool will distribute appropriate resource according to the way of resources management when the required task is put into it. It also will manage the resource and carry out the task in a certain scheduling way. Each resource will make further scheduling to the distributed task according to their scheduling strategy [1] Since the task scheduling in grid computing faces a NP-hard problem, it has drawn attention from many scholars and become the focus in the field of the
current grid computing research.

In recent years, two global random and optimal algorithm have been widely studied and applied in the field of the grid computing research: GA(Genetic Algorithm) and SA(Simulated Annealing).Vincezo once introduced a task scheduling method in grid computing based on GA which aims at enhancing as greatly as possible the utilization rate of resources and throughput[2,3]. Abraham and many other scholars also introduced the application of such evolutionary algorithm as SA to the task scheduling in grid computing [4].

PGSAA (Parallel Genetic simulated Annealing Algorithm) is a optimal algorithm combing GA with SA.GA is weak in local search but powerful in global search while SA is weak in global search but powerful in local search. The introduction of PGSAA in the grid computing task scheduling is similar to the general process of standard GA. PGSAA begins its global optimal search process with a initial population randomly generated. It first generates a new group of individuals through genetic operation such as production, crossover, mutation ,etc, and than simulated anneals independently all the generated individuals respectively, whose results are counted as the individuals of the next population .This operational process proceeds continuously until a certain termination condition is coincident .Thus, PGSAA can greatly enhance the running efficiency of algorithm and its solution quality by fully combining the advantages of GA and SA .

II. THE DEFINITION OF PROBLEM

In the grid computing, task scheduling is essentially to distribute n interdependent tasks to m isomerous available resources to make the whole task fulfilled in the shorten time and the resources used as fully as possible. Here is the general definition of this problem:

$\Pi=(Q,D,A,\Theta,\Delta,L)$

$Q=\{q_1,q_2,\ldots,q_m\}$

$D=\{d_1,d_2,\ldots,d_n\}$

$A=(\Omega,E,P_0,M,\Phi,\Gamma,\Psi,\Xi,T_0)$

$L=\{L_1,L_2,\ldots,L_m\}$

$Q=\{q_1,q_2,\ldots,q_m\}$represents the collection of m isomerous resources available in the grid computing;

$D=\{d_1,d_2,\ldots,d_n\}$represents the collection of n interdependent tasks in the grid computing;

$A=(\Omega,E,P_0,M,\Phi,\Gamma,\Psi,\Xi,T_0)$ is the PGSAA designed in this paper, Ω is the way of encoding; E is the fitness function of the individuals ; P_0 is the initial population; M is the size of the population; Φ is the production operator; Γ is

Shijue Zheng is with the Department of Computer Science, Hua Zhong Normal University, Wuhan 430079, China (e-mail: zhengsj@mail.ccnu.edu.cn).

Wanneng Shu is with the Department of Computer Science, Hua Zhong Normal University, Wuhan 430079, China (e-mail: shuwanneng@yahoo.com.cn).

Li Gao is with the Department of Computer Science, Hua Zhong Normal University, Wuhan 430079, China (e-mail: gaoli@mail.ccnu.edu.cn).

1-4244-0318-9/06/$20.00 ©2006 IEEE

the crossover operator , Ψ is the mutation operator; Ξ is the termination condition , T_0 is the initial temperature.

Θ is a m×n matrix, in which Θ_{ij} stands for the complete time of the task d_j at the resource node q_i in the ideal state(the resource load unconsidered).

Δ is a m×n matrix, Δ_{ij}=1 stands for the task d_j distributed to the resource node q_i, otherwise =0. The matrix Δ comprised of the variable has the following features: (1) all the elements are 0 or 1; （2）There is one element (i.e. 1) in every column; （3）If the rows of every column in the matrix in which the element 1 exists are marked with, $K_1,K_2,\ldots,K_j,\ldots,K_n$, then X=（$K_1,K_2,\ldots,K_n$）corresponds to a distributive scheme.

L= {L_1, L_2, …, L_m}respectively represent the dynamic load weight of m resource nodes.

In addition, such formalized description as the follows are specially made so that the current load of each resource node can be properly evaluated and the target function of the task scheduling can be established.

Definition 1 :Supposing the load parameter of resource node q_i such as usage rate of CPU, usage rate of memory, current network flow, access rate of the disk I/O, response time, process amount are respectively represented by Cpu(i)%, Memory(i)%, W(i), Io(i)%, Rt(i), Pr(i).the dynamic load weight of resource node r_i can be formulated as:

$$L(i)=\pi_1\times C(i)\%+\pi_2\times M(i)\%+\pi_3\times N(i)+\pi_4\times Io(i)\%+\pi_5\times P(i)$$

In which $\Sigma\pi_i$= 1（i=1，…,n），π_i reflecting the importance degree of the each load parameters. Because in the different kinds of system, the parameters differ among themselves in the degree of importance, which is the reason why it is especially adopted ,and moreover it can be properly adjusted in the proportion of the application of different parameters in the running process of the system[5]

Definition 2: Supposing X represents one scheme of all possible distributive scheme, then under such a strategy, the target function (complete time) of the task scheduling in grid computing is

$$F(x)=\sum_{i=1}^{m}\sum_{j=1}^{n}G(\Delta_{ij},\Theta_{ij},L(i)) \qquad (1)$$

$G(\Delta_{ij},\Theta_{ij},L(i))=\dfrac{C\times\Delta_{ij}\times\Theta_{ij}\times\sqrt{L(i)}}{\sqrt{\sum_{i=1}^{m}L(i)}}$, it is a function displaying the cost of distribution and C is a constant. It shows the cost estimate of the task d_j distributed on the resource node q_i the bigger load weight the resource node has the higher cost it makes when it implements the task on it at the ideal state. The target function above is intended to adjust the tasks distributed to the each resource node, so that the least total time is needed.

III. THE APPLICATION OF PGSAA

A. Encoding and the Initiation Population

This paper employs the natural number to encode the variable Δ_{ij}. That is, the row of every column in the matrix Δ in which the element 1 exists is regarded as a gene. The genes are independent of each other. They are marked by K_1, K_2,..., K_j,..., K_n ,in which $K_j\in[1,m], j\in[1,n]$ and K_n may be a repeatedly equal natural number.

When the distributive method at random is employed to produce the initial population comprised of certain individuals, the population must be in a certain scale in order to achieve the optimal solution on the whole. Here is the concrete way: Generate M individuals randomly that the length is n, and then the chromosome bunch encoded by the natural number is calculated as the initial population.

B. The Fitness Function and Linear Transformation

Formula (1) is properly transformed into $f(X)=e^{-F(X)/T}$, and $f(X)$ is regard as the fitness function ,in the formula , X=（$K_1,K_2,\ldots,K_n$）, T as the temperature of the evolution ,and F(X) as the target function.

Since GA is liable to result in the phenomenon of prematurity convergence. The fitness function needs a linear transformation. The linear transformation of fitness value, that is the early evaluation stage, will amplify the fitness value of those individuals whose fitness value is smaller than the average fitness value of the population and lessen the fitness value of those individuals whose fitness value is larger than the average fitness value of the population .the late stage of evolution takes a reverse process. Therefore, it is vital to distinguish between the early stage and late stage of evolution. Such a distinction is made in this paper by the sample standard warp, which is the appropriate measure to scale the difference of individuals within a population, as shown in formula (2),

$$\sigma=\sqrt{\frac{1}{M-1}\sum_{i=1}^{M}|f(x)-\overline{f(x)}|} \qquad (2)$$

In which, M is the population scale, $f(X)$ is the fitness value of the individuals X, $\overline{f(x)}$ is the average fitness value of the population.

If the sample standard warp of population is above the given critical value δ, the population is regarded as in the early evolution stage, otherwise in the late stage, and the linear transformation of fitness value can be shown in formula (3)

$$f(x)'=\alpha\times f(X)+\beta \qquad (3)$$

The value of α and β is divided into the following two cases:

!)if $\sigma > \delta$, $\begin{cases} \alpha = r \\ \beta = (1-r) \times \overline{f(X)} \end{cases}$ (4)

In which , r=random[0,1], $\overline{f(x)}$ is the average fitness value of the population.

2)if $\sigma \leq \delta$, $\begin{cases} \alpha = k \\ \beta = -f_{\min}(X) \times k \end{cases}$ (5)

In which, K= $\frac{\overline{f(x)}}{f(x)-f_{\min}(x)}$, $\overline{f(x)}$ is the average fitness value of the population and $f_{\min}(X)$ is the minimum fitness value in the population.

C. Reproduction Operator

Reproduction is the transmission of personal information from the father generation to the son generation. Each individual in each generation determines the probability that it can reproduce the next generation according to how big or small the fitness value is. Through reproducing, the number of excellent individuals in the community increases constantly, and the whole process of evolution head for the optimal direction. We are adopting roulette selection strategy; each individual reproduction probability is proportion to fitness value.

1) Computing the reproduction probability of all the individuals $P(i)=\frac{f(i)}{\sum_{i=1}^{M} f(i)}$

2)Generate a number r randomly, r=random[0,1];

3)If P(0)+ P(1)+…+ P(i-1)<r< P(0)+ P(1)+…+ P(i),the individual i is selected into the next generation.

D. Crossover Operator

Crossover is the substitution between two individuals of the father generation that is to generating new individual .The crossover probability Pc directly influences the convergence of the algorithm. The larger Pc is the most likely is the genetic mode of the optimal individual to be destroyed .However, the over-small of Pc can slow down the research process .Here is the definition of the crossover operator:

1) Computing crossover probability Pc

$$Pc=\begin{cases} 0.9-\frac{0.01\times(f(X)-\overline{f(X)})}{f_{max}(X)-f_{min}(X)} & f(X)\geq\overline{f(X)} \\ 0.9 & f(X)<\overline{f(X)} \end{cases}$$

In which, $f_{max}(X)$ is the maximum fitness value of the population, $f_{min}(X)$ is the minimum fitness value of the population and $\overline{f(X)}$ is the average fitness value of the population.

2) Generate a number r_c randomly, r_c =random [0,1] .if r_c < Pc , then conducting the following crossover operation: $V_1'=r_c* V_1, V_2'=r_c* V_2$, V_1 , V_2 are both father individuals, and V_1' , V_2' are both son individuals.

3) Computing the fitness value $f(V_1')$ and $f(V_2')$, if $\min\{1,\exp(-(f(V_1')-f(V))/T_k)\}$>random[0,1], it accepts the new solution. T_k is the evolution temperature of the k^{th} time.

E. Mutation Operator

The role of the mutation operator lies in that it enable the whole population to maintain a certain variety through the abrupt change of the mutation operator when the local convergence occurs in the population; the selection of the mutation probability Pm is the vital point because it influence the action and performance of the GSSA .if Pm is over-small, the GSSA will become a pure random research .Here is the definition of the mutations operator

1) Computing the mutation probability Pm

$$Pm=\begin{cases} 0.01-\frac{0.09\times(f(X)-\overline{f(X)})}{f_{max}(X)-f_{min}(X)} & f(X)\geq\overline{f(X)} \\ 0.01 & f(X)<\overline{f(X)} \end{cases}$$

In which, $f_{max}(X)$ is the maximum fitness value of the population, $f_{min}(X)$ is the minimum fitness value of the population and $\overline{f(X)}$ is the average fitness value of the population.

2)Produce a number r_m =random [0,1], if r_m <Pm, then conducting the following mutation operation to the individual V_i : Using the approach of random to produce w=[random(1,m)], then using w to substitute Kw in V_i , if Kw = w, then produce a new w'=[random(1,m)] randomly again, until $Kw' \neq w'$.

3) Whether the solution is accepted or not after the mutation accord to step 3) in the crossover operator.

F. Decrease Function

$T_{K+1} = T_K \times (1-\frac{\kappa}{M})$, T_K , T_{K+1} stands for the temperature in the evolution process of the k^{th} time and the $(k+1)^{th}$ time ,M is the scale of the population.

G. Termination Condition

When the matching error $\varepsilon \approx 0$ or the temperature T in the process of cooling no longer falls, the annealing process will natural stop.

H. The Solution Process of PGSAHA

1) Initialize control parameters: M, Ξ ,T_0,k=0;

2) Encode and generate mother population P0 randomly;

3) Estimate the fitness value $f(X_i)$ of each individual in population P_k,i=1,2,…,M;

4)Compute the sample standard warp σ， if $\sigma > \delta$, α and β gains their respective value according to formula (4),otherwise to formula (5).

5) Change the fitness value as linear transformation

6) Conduct the operations as follows at the current temperature T_k:

a make a reproduction process of P_k to generate the father population F_K.

b. make a crossover process of F_k to generate the crossover population C_K;

c. make a mutation process of C_K to generate the medium population M_K

d. generate the new population $P_{k+1}=F_k \cup M_K$. by combining the father population F_k with the medium population Mk.

7) When the terminal condition is coincident, the annealing process will natural end; otherwise, $T_{K+1} = T_K \times (1 - \frac{\kappa}{M})$,k=k+1, return to step 3).

IV. Simulation Experiment and Result Analysis

To check the research capability of the operator and its operational efficiency, such a simulation result is given compared with the GA in literature [4] and the SA in literature [6].the major parameters employed in the algorithm such as the population scale M=100,the initial annealing temperature T_0=1,the terminal annealing temperature Ξ =0,the crossover probability Pc and the mutation probability Pm are attained in the dynamic operational process.

The platform of the simulation experiment is a Dell power Edge2600 server(double Intel Xeon 1.8GHz CPU,1G memory , RedHat Linux 9.0， Globus Toolkit3.0 as middle ware).

Figure 1 ~ 2 show the static quality of the algorithm and list the optimal solutions search by the algorithm in the different evolution algebra. They illustrate the simulation results of 50 tasks at 3 resource nodes and 5 resource nodes respectively.

From Figure 1 ~ 2, we can educe that PGSAA has a higher convergence speed and more reasonable selective scheme which guarantees the non-reduction performance of the optimal solution. Therefore, it is better than GA and SA through the theoretic analysis and the experimental results.

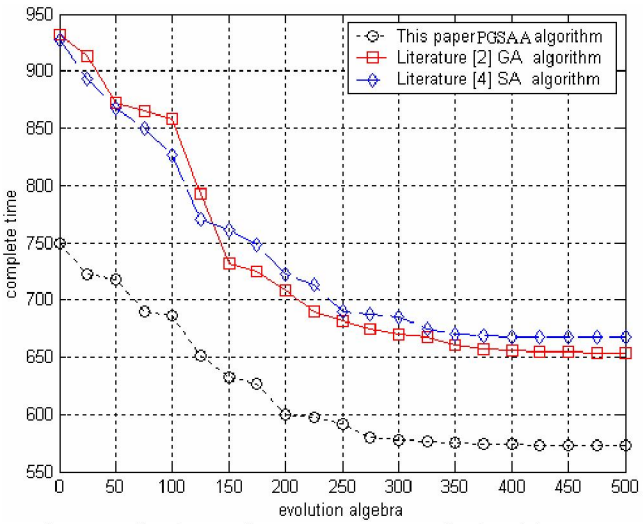

Figure 1.Static performance curve of algorithm (m=3, n=50)

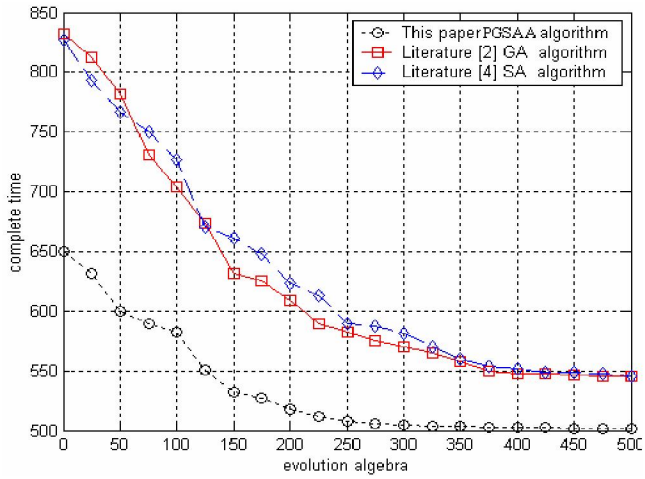

Figure 2.Static performance curve of algorithm (m=5, n=50)

V. Conclusions

Theoretically, both GA and SA is optimal algorithm based on the probability distribution scheme. The optimal scheme of SA is to achieve the global optimal by endowing the search process with a time change and an abrupt probability change which eventually approaches zero to avoid being stuck in the local minimum; GA is to realize the optimal through the genetic operation of the population based on the idea of survival of the fittest in the sense of probability .As the combination of the two optimal scheme ,PGSAHA proposed in this paper can not only enrich and hospitalize the whole process, but also enhance the ability and efficiency of the search in the global and local sense.

References

[1] Rajkumar Buyya, David Abramson, Jonathan Giddy. Grid Resource Management, Scheduling, and Computational Economy [A].WGCC 2000[C].Japan, March 15-17, 2000.

[2] Vincenzo Di Martino.Schduling in a grid computing enviroment using genetic algorithm. Marco Mililoti the 16th Int'1 Parallel and Distributed Processing Symposium (IPDPS2002), USA.2002
[3] Vincenzo Di Martino, M Mililotti. Sub-optimal scheduling in a grid using genetic algorithm. Parallel Computing, 2004, 30(5/6):553~565.
[4] Ajith Abraham, Rajkumar Buyya. Nature's heuristics for scheduling jobs on computational grids. The 8th Int'1 Conf on Advanced Computing and Communications (ADCOM 2000), Cochin, India, 2000
[5] Shijue Zheng，Wanneng Shu and Guangdong Chen:A Load Balanced Method Based on Campus Grid, 2005 International Symposium on Communications and Information Technologies (ISCIT 2005). October 12- 14, Beijing.
[6] ZHANG Jiang-She, XU Zong-Ben, LIANG Yi.Global Annealing Genetic Algorithm and its convergence well necessary condition [J].SCIENCE IN CHINA(Series E)，1997，27（2）：154~164.
[7] WANG Xia, ZHOU Guo-Biao. Strong Convergence (a.s.) of Global Annealing Genetic Algorithm. MATHEMATICA APPLICATA，2003，16 (3):1~7.