Application of Simulated Annealing Algorithm to Grid Computing Scheduling based on GridSim

Yuhua Guo, Xiaolin Wang

Department of Computer Science & Technology

Anhui University of Technology

Ma'anshan 243002,Anhui Province, China

guoyuhua@ahut.edu.cn, wxl@ahut.edu.cn

Abstract: **Simulated annealing algorithm has lots of advantages, but so far there was little research on how to apply it to grid scheduling. The main cause had two problems. One was how to decrease its overhead; the other was how to obtain its parameters. After introducing characters on scheduling in grid, one scheduling algorithms was proposed. In order to get the appropriate algorithm's parameters and to test its efficiency, a newly tool kits called Ana-GridSim based on GridSim was developed. Results of experiments showed the algorithm nice and the Ana-GridSim effective, feasible and convenient to study grid scheduling.**

Keyword: grid computing; scheduling; simulated annealing; GridSim

I. GRID COMPUTING SCHEDULING AND GRIDSIM

Grid computing is a share and cooperation resource pool in the eyes of scheduling. In this pool, the owner put into its resources with some restrictions and user expressed his demands and requirements with some cost or donations. So, grid scheduling always meant to make a nice match between these resources and tasks or jobs, playing an important role in grid [1]. The characters of this scheduling could be outlined as follows: (1) Heterogeneous. Resources were resident in different environment, such as operating system, hardware, mid-ware etc. This required a uniform operation interface. According to this, some simulation tools regard all abstract resources as a layer; (2) Instability. Resource including data, load and the ability of the network etc, may be able or disable from time to time, which meant that it must be nicely checked before use or scheduling;(3) Relative stability. For some duration, resources and important parameters in grid kept stable, otherwise it might be never useful or scheduled; (4) Transparency. All the details about scheduling were in the charge of the grid, transparently to users, who only expressed their requirements and submitted applications.

As what were stated above and as [1,2,3,4,5,6] grid scheduling was much complicated, and it was not easy to be studied. There were three main methods. The first is to set up a real experimental grid environment. The second one is simulation, which used a simulating environment for experiments. Building up real grid to study scheduling had a long cycle and enormous cost. Because of experiments carried out by large numbers of people and long time spent, it's not easy to control and analyze. Moreover, it is difficult to maintain the consistency of status and data in experimental grid. By contrast, grid simulator is easier for grid scheduling study. On one hand, the simulator can greatly reduced unnecessary work and improve efficiency. On the other hand, it also provides a uniform platform for grid scheduling. Because environment and the experimental conditions can be reconstructed in the simulator, different scheduling models or algorithms can also be compared with each other. The third is Petri nets. According to to the study by Jin Hai, it seems attractively that it might be applied to study fault tolerant models of scheduling in grid [2].

So far, there were many scheduling algorithms in grid. Some of them could be suitable for dynamic loading, besides networks load[3]. Algorithms based on simulated annealing (SA, for short) in this environment had not been seen yet, although it has excellent performance of global optimization in many fields. The main cause had two. One was how to decrease its overhead; the other was how to obtain its parameters. As far as the former was concerned, we can construct the hierarchical structure of grid scheduling to reduce its size greatly. In this paper, we mainly pay our attention to both in one level of the hierarchical structure of grid scheduling.

In section II, scheduling algorithms in grid based on simulated annealing were proposed. Applying SA should carry out experiments to obtain the appropriate values of its parameters of T_0,L_0,L_k,L,i_0,etc..GridSim was one of the nicest grid simulators[4,5,6]. In order to get the parameters' value, in section III, a new tool kits based on GridSim was developed and introduced in details. In section IV, parameters of those algorithms and experimental result were discussed. In section V, we drew some conclusion based on our work.

II. SIMULATED ANNEALING SCHEDULING ALGORITHM IN GRID

SA is an algorithm that simulated the procedure of physical solidification. The most difference is its principle of reception, which was called "Metropolis" rule. It transmitted from one state to another not only according to the fact that the new state was better than the older one, but it also transmitted if the new and worse state was "hit" by a function of random. It made it jump out of local extremum.

978-1-4244- 7618-3 /10/$26.00 ©2010 IEEE

All the states made up of a link, called “Markov chain”. The advantage is its capability of global optimization. The procedure of searching of the state of optimization was controlled by many factors indicated by table 1. So, applying of SA should obtain those factors’ appropriate value. Those parameters interacted tightly and controlled the optimization procedure.

In order to reduce SA’s overhead, hierarchical structure of grid scheduling should be adopted. But scheduling algorithms in each level may be the same. It should not only take load of machines(cpus) into account ,but also network’s load should be considered, especially for data intensive application. We constructed the target function of time combined with both load factors based on as formula (1), which could be adjusted by different weight α,β,α□[0,1],β□[0,1]. Practically, T_{rj}、M_{rj} were calculated by formula(2),(3)respectively.

TABLE1. PARAMETERS OF SA

T_{k+1}	temperature parameter
T_0	initial temperature
L_k	length of the inner kth Markov chain
i_0	initial state
L	ΣL_k
k	number of chasing optimization
TC	Terminate Condition

$$f(i) = \alpha \sum_r \sum_j \chi_r(j) T_{rj} + \beta \sum_r \sum_j \chi_r(j) M_{rj}$$

$$= \sum_r \sum_j \chi_r(j)(aT_{rj} + \beta M_{rj}) \quad (1)$$

$$M_{rj}(Data) = \frac{Input + Output}{L} * t \quad (2)$$

$$T_{rj} = \text{Task length} / \text{remaining capability} \quad (3)$$

Here, when resources r complete j, $\chi_r(j)=1$, otherwise $\chi_r(j)=0$.

The algorithm run as follows:

(1) Request and discovery resources and filter them, according to characters of resource and demand from task, performance;

(2) Extract tasks to be scheduled; the condition here is that the number of tasks extracted should not be greater than the number of resources;

(3) Retrieve dynamic information of each resource, then estimated completion time of the task and its communication;

(4) Call simulated annealing algorithms in grid computing;

(5) Distribute the task scheduled to target resources and update resources’ information;

(6) Compute the number of returning tasks;

(7) If there is more work to be scheduled, return step (2) ;

(8) Repeat (1)-(7)until all tasks completed.

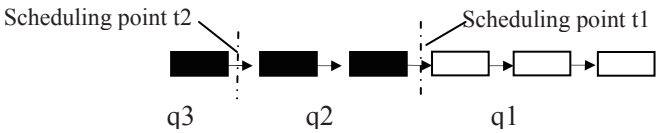

Figure1. Queue of tasks

As Fig.1 showed, tasks formed a queue q1. According to above, when scheduling point t1 came, all tasks in the queue q1 formed a queue which was scheduling(or ready to schedule), and all other new tasks arrived after this scheduling point t1 formed another queue q2 to be scheduled. Similarly, when another scheduling point t2 came, all tasks in the queue q2 formed a queue which was scheduling(or ready to schedule), and all other new tasks arrived after this scheduling point t2 formed another queue q3 to be scheduled.

III. NEWLY-DEVELOPED TOOL KITS BASED ON GRIDSIM AND ITS APPLICATION

GridSim was firstly developed by Rajkumar Buyya , a Pro in University of Melbourne, Australia. It can run on Windows or Linux OS. In GridSim, there were lots of entities, such as heterogeneous resources, users, applications and resource agency, scheduling, networks, routers, linkers etc. GridSim regarded grid as a market in which through "buying" and "selling"[4].

Generally speaking, GridSim had a strong simulation capability. It had a quantity of development kits, including Auction, data scheduling, network package etc. But it had not special analysis tools for particular application. Especially, it had not tools to help obtain parameters of SA. So, we extended GridSim core and developed some tools. It was Ana-GridSim as Fig.2. It can be regarded as a series of analysis tools, the core of GrimSim and input, outputor, receptor models, etc.. As shown in Fig. 2, the core GridSim mainly composed of the Broker and User Entity. User entities on behalf of grid users submitted tasks in form of a instance of Experiment, which was the wrapper of tasks. Scheduling algorithm can reconstructed according to application, for instance, SA for grid. After submit, it waited for signal of return. Broker interacted with the core of grid, including retrieving information of resources from global or local GIS, scheduling etc.

Outputor output the information of scheduling in details. Analyzer received those output info and generated evaluating index. In order to obtain parameters of SA and evaluate the scheduling, the analyzer included many tools for especial research as follows:

(a) Scheduling length L. It is average the number of tasks scheduled each time. Its fluctuating changes reflected its capability to dynamically adapt. Obviously, $\overline{L} = \sum_{i=1}^{n} L_i$ /n reflected the average scale of scheduling. According to Fig.1 and SA for grid, it practically had two levels of $\overline{L}$. One is the average of q1,q2,q3 described in Fig.1; the other is average for step (2) in SA for grid described in section 2, which is not greater than the number of resources;

(b)The amount of tasks rescheduled. The experimental results showed the interval between distribution and reception of tasks related with it greatly;

(c)Average experimental error of scheduling algorithm, d= $\sum_{i=1}^{n} d_i$ /n. It was the average accuracy of n times of simulations. Here, $d_i = (f - f_{\min}) / f_{\min}$ *100%, and $f_{\min}$ was the minimum;

(d) The total length of task was completed on each resource. To a certain degree, it indicated the results of scheduling;

(e) Reception rate of each scheduling, a special tool for simulated annealing, was used to produce temperature charts. It indicated the convergence of SA;

(f) The most busy resource, and its complete task ratio (%) ;it can be concluded from(d);

(g) Rate of failure (%), which indicates failures caused by all kinds of situation.

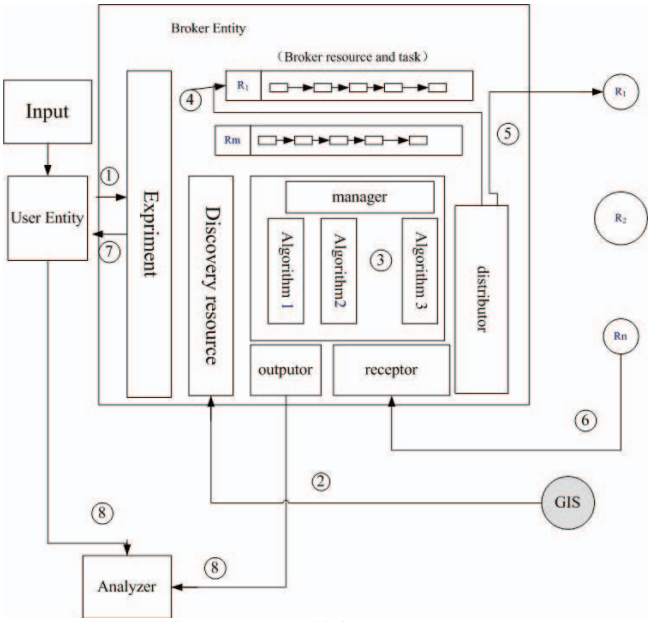

Figure2. Ana-GridSim Structure

IV. Simulation Experiments and Results

In order to experiment, we constructed the network topology as Fig. 3. Among it, r1 to r6 were routers, R0 to R6 were resources, which corresponded to the 1st to 7th resources shown in the Table 2. Routers were first-in first-out scheduling model (FIFO Schedule). The details of these resources were from real grid WWG shown in table 2. During simulation experiments, there were two users, and each user submitted seven tasks, which had random length of input and output, except random length of tasks' calculation. Due to the capacity of experiment facilities, experimental data showed that if the number of resources was greater than seven, it resulted in greater overhead. And

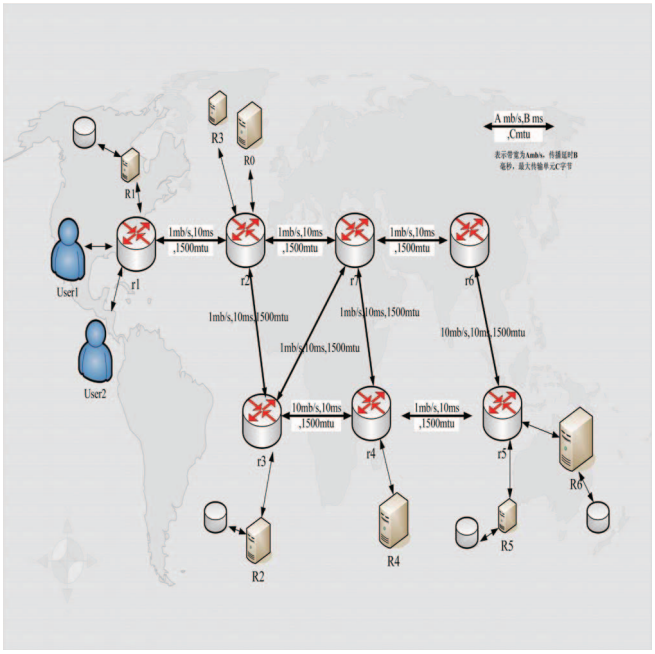

Figure3. the simulation network and resources deployed

seven was suitable. Maybe, it can be deployed more resources in real grid, but for the sake of SA's overhead, number of resource in each levels should be carefully considered. It always required experiments.

After the number of resources in each level was decided, lots of simulations should be carried out to determine and verify the parameters of SA. To observe the influence of those parameters in table 1 to scheduling performance and accuracy, when one of them was changed, the rest were kept unchanged. When one of them was determined according to experimental data, one of the rest undetermined parameters was selected to be determined. At first, we chose the initial state of T_0=380 , θ=0.92and L_k=150*m(m was the number of tasks to be scheduled, the same as below)and let L change freely. The sequence of the parameters determined were L,θ(corresponding to T_{k+1}, for T_{k+1}=θ*T_k),T_0 and L_k, each of which was selected the value of 11000*m,0.85,280 and 100*m in sequence, according to the average error of 20 times scheduled as showed in table 3.

In order to test the (2) step in SA for grid computing

scheduling described in section 2. We constructed another SA for grid computing scheduling. The difference was that the new one was no condition described in (2) step in section 2, i.e., more than one tasks might be scheduled to one resource. By contrast, the new one was called NO.2 SA. As far as those two kinds of scheduling algorithms as concerned, after parameters were studied well, simulations were carried out to observe their performance. The experimental results of SA were list in Table 3, table 4. The experimental results of NO.2 SA were list in Table 5. All of those data were corresponding to the tools introduced in section 3. Here, Avr in table 3 was the average error of 20 times scheduled, corresponding to the tools (c) introduced in section 3.

V. ANALYZING AND CONCLUSION

To reduce size in each level, according step (2) described in section 2, it is necessary to control the number of resources and that of tasks in each level . Table 3 showed the results and the procedure of determining parameters of SA. Firstly, L was selected the value of 11000*m, even with the result of 3.72% average errors, for that we wanted to reduce the number of L to reduce overhead in each level of the hierarchical structure of grid scheduling. The L,θ(corresponding to T_k, for $T_{k+1}=\theta*T_k$),T_0 and L_k was selected the value of 11000*m,0.85,280 and 100*m. The errors seemed to be eliminated by the rest parameters for they were interacted tightly each other. Table 5 indicated that it should be necessary one tasks to one resource for each scheduling calculation. Otherwise, it would lead to great overhead and worse results.

Table 4 showed how the networks capacity affected scheduling in gird based on SA greatly. From it , it could been seen that R6 resources was the most load one in non-network environment; but in network environment, situation changed for there was another factors, i.e., data and its transportation time we should consider. As a result, R0 resource became the most load one. This was the mesh network scheduling, as well as consistent with the actual situation.

TABLE2. DETAILS OF THE SIMULATION RESOURCES [4]

Resource name	***Features Simulation of resources***	***SPEC/MIPS***
R0	Compaq,AlphaServer,CPU,OSF,4	515
R1	Sun,Ultra,Solaris,4	377
R2	Sun,Ultra,Solaris,4	377
R3	Sun,Ultra,Solaris,2	377
R4	Intel,Pentium/VCb20,linux,2	380
R5	SGI,Origin 3200,IRIX,6	410
R6	SGI,Origin 3200,IRIX,16	410

From above, we also can make a decision that Ana-GridSim is a nice tool kits to study grid scheduling and has its advantage of effectiveness, feasibility and convenience.

TABLE3. AVERAGE ERROR AND THE PARAMETERS OF SA

θ	Ave%	L/m	Ave %	L_k/m	Ave %	T_0	Ave %
0.6	0	100000	0	80	0	100	0
0.8	0	20000	0	100	0	280	0
0.85	0	12000	1.64	150	0	500	0
0.92	3.73	11000	3.72	200	0.10	1000	0.00
0.95	5.52	9000	4.47	250	1.33	1500	0.03

TABLE4. NETWORK VS. NON-NETWORK SCHEDULING SIMULATION

α	β	(g)	(c)	(a)	(f)
1	0	0	0	7	R 6,80%
1	1	0	0	7	R 0,45%

TABLE5. NETWORK SCHEDULING SIMULATION BY NO.2 SA

α	β	(g)	(b)	(c)	(a)
1	1	0	0	15.7%	7
1	1	0	1	11.38%	7
1	1	0	2	9.14%	7
1	1	0	3	8.05%	7

REFERENCES

[1] Ian Foster, Carl Kesselman (write), Jin Hai, Yuan Pingpeng, and Shi Ke (translate). Grid Computation (the 2nd edition). Beijing: Electronic Industry Publishing House, 2004.10,1-39, 124-149,372.

[2] Jin Hai, Chen Gang, Zhao Meiping. Research on a Job Scheduling Model for Fault Tolerant Computational Grid. Journal of Computer Research and Development, Aug. 2004, Vol. 41 (No. 8):1382-1388.

[3] Donna Griffin & Dirk Pesch. A Dynamic Service Provisioning Platform for Next Generation Networks – The Agent Grid Service Marketplace.http://ww.aws.cit.ie/Personnel/Papers/Paper284.pdf . (November 26,2006)

[4] Rajkumar Buyya1, and Manzur Murshed GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE Concurrency Computat.: Pract. Exper. 2002; 14:1175–1220 (DOI: 10.1002/cpe.710)

[5] http://www.gridbus.org/gridsim/.gridsimtoolkit-3.2/gridsimtoolkit-3.2 /doc/index.html,10/5/2008

[6] Liu, Xiangrui, Zhu Jianyong, Fan Xiaozhong , Grid Scheduling Simulations Based on GridSim, Computer Engineering, 2006, Vol. 32(2): 42-44