Parallel Network Simulation Scheduling Algorithm Oriented Towards Multi-Task

Xiangzhan Yu[1], Qinghai Yang[1], Dongbin Wang[2]

[1] Harbin Institute of Technology, 150001, China
yxz@hit.edu.cn qinghaiyang@hit.edu.cn
[2] School of Software Engineering, Beijing University of Posts and Telecommunicatons,100876, China
dbwang@bupt.edu.cn

Abstract. To meet the demands of the large-scale network simulation platform technology, we proposed a novel multi-task network simulation scheduling algorithm based on multi-valued mapping and the new algorithm is called MUNS-Min-min. Experimental results show that our algorithm is suitable for complex computing environment.

Keywords: distribute computing; grid; network simulation; task scheduling

1 Introduction

Currently, there has been a lot of researches on the parallel network simulation algorithm for single simulation task. However, a large-scale network simulation platform may receive multiple simulation tasks at a time or successively. If the server just executes each task in order of its arrival time, it cannot guarantee the shortest total execution time. Current parallel network simulation algorithm did not consider the issue mentioned above. So in the complex computing environment, studying how to divide these simulation tasks, allocate computing resources and schedule tasks has great significance for reducing the runtime of all the simulation tasks and improving performance of the simulation platform. We learned from the research results of task scheduling algorithm in grid computing, and proposed a novel multi-task oriented parallel network simulation algorithm based on the complex computing environment.

In the field of traditional parallel and distributed scheduling, considerable researches have been done centering on scheduling algorithm. They have put forward some sophisticated scheduling algorithms[1-6], such as OLB(Opportunistic Load Balancing) algorithm,.UDA(User-Directed Assignment) algorithm,.Fast Greedy algorithm, Min-min algorithm, Max-min algorithm, Greedy algorithm, GA(Genetic Algorithm) algorithm, SA(Simulated Annealing) algorithm, GSA(the Genetic Simulated Annealing) algorithm, Tabu algorithm, A* algorithm.

Experiments of the literature showed that OLB algorithm, UDA algorithm, Max-min algorithm, SA algorithm, GSA algorithm and Tabu algorithm are worse than Min-min algorithm, GA algorithm and A* algorithm normally. The last three algorithms' differences in completion time do not exceed 10%. GA is a little better than Min-min because we use Min-min algorithm to seed in GA algorithm. In the last three algorithms, Min-min algorithm is the fastest algorithm, GA is the second, and A* is the slowest.

2 Parallel network simulation scheduling algorithm oriented towards multitask

On the large-scale network simulation platform, the task's input time and scale is uncertain, and one or more tasks may arrive at a time. Therefore, we choose batch mode scheduling strategy in dynamic scheduling algorithm as our scheduling strategy. According to the characteristics of network simulation, we choose to improve on the basis of Min-min algorithm. Through improving the single valued mapping between tasks and the computing nodes in the Min-min algorithm into multivalued mapping, we proposed a novel multi-task network simulation algorithm based on multivalued mapping which is called MTNS-Min-min(Multi-task Network Simulation Min-min).

2.1. The algorithm idea

Our algorithm is based on some assumptions. Firstly, we should have mastered m parallel network simulation partition algorithms oriented towards single task. Secondly, the runtime of each simulation task is far greater than parallel partition time of simulation task. Finally, for each task, we can predict possible consumption time of m algorithms separately in the complex computing environment. Prediction method can be found in literature [7] that we wrote.

During the parallel network simulation, the consumption of each task consists of two parts, the runtime and resource occupied. The main idea of this algorithm is assigning priorities to the simulation task with the least runtime and resource degradation when scheduling the queue of simulation tasks. In batch mode, it collects the tasks and form a task set instead of mapping the simulation task to the machine immediately, and the task set will be mapped concentratedly after the arrival of mapping event. The independent task set is called meta-task.

Under the influence of m partition algorithms $P=\{s_0, s_1, \cdots, s_{m-1}\}$, the mission of our algorithm is mapping the S tasks $M_v=\{t_0, t_1, \cdots, t_{S-1}\}$ of meta-task M_v to n hosts $H=\{m_0, m_1, \cdots, m_{n-1}\}$ in a reasonable way.

We define the following parameters for a certain task t_i in meta-task.

b_i : the starting execution time of task t_i;

d_j : the expectant time that machine m_j has finished all the tasks assigned to it and has begun to execute another task;

ET_{ik} (expectant execution time): the expectant execution time of task t_i after the partition of partition algorithm s_k;

ER_{ik} (expectant execution resource): the expectant resource occupied by task t_i after the partition of partition algorithm s_k;

CT_{ik} (expectant completion time): the expectant completion time of task t_i after the partition of partition algorithm s_k, and $CT_{ik} = b_i + ET_{ik}$;

The MTNS-Min-min scheduling algorithm is shown as follows:

```
do until all tasks in M_v are mapped or all machines are marked
for each task t_i in meta-task M_v
   for each single task partition algorithm s_k in partition algorithm set P
      compute ET_ik and ER_ik in unmarked machines;
for each task t_i in M_v
   ET_ik set the predicted run-time through BPnet in all machines of ER_ik;
   b_max set the maximum of d_j in all machine m_j in ER_ik;
   CT_ik = b_max + ET_ik
find the task t_l with the shortest completion time and the least resources;
assign task t_l to the machines in ER_lk;
delete task t_l from M_v;
for each machines m_j in ER_lk
     update d_j and mark machine m_j;
end do
```

1. For each task of meta-task, we separately partition it according to the algorithms of the partition algorithm set. We should know that the computer resources here are just those unallocated resources. If task t_i can not find suitable computer resources through partition algorithm s_k, ET_{ik} and ER_{ik} are assigned to NOP.

2. Analyse each task in turn and predict the completion time of the task. After considering the expectant completion time and dissipative resources of all the tasks, we need to choose the task with the least execution time and resource consumption and label the machines occupied by it. The algorithm won't stop until all meta-tasks are mapped to specified machines or all machines are assigned.

3. Parallelly simulate all assigned tasks, and wait for the next mapping event.

4. Repeat above processes until the task queue is empty.

Suppose the average time complexity of partition algorithm is O_{PART}. For partition algorithm MTNS-Min-min, only one task can be assigned in each loop, consequently the average time complexity of the algorithm is $SO_{PARKT}+(S-1)O_{PARKT}+(S-2)O_{PARKT}+\ldots+O_{PARKT}=O(S\times S\times O_{PARKT})$, which is S^2 times more than the time complexity of the partition algorithm. S is the number of simulation tasks in the meta-task.

To improve the execution efficiency of MTNS-Min-min, there will be another machine that will run constantly to execute the partition algorithm for each arrived task during the implementation of the algorithm.

3 Conclusion

In summary, we have improved the classical Min-min algorithm and proposed the multi-task simulation scheduling algorithm which is called MTNS-Min-min. Focusing on how to ensure the simulation time, the resource consumption, and trigger mapping events, we provide detailed solutions. It not only supports the single simulation task partition algorithm based on a homogeneous partition, but also supports the single simulation task partition algorithm based on a isomerism partition. Therefore, MTNS-Min-min is also suitable to complex computing environment.

Acknowledgment

This work has been supported by the National Natural Science Foundation of China under Grant No. 61173144.

References

1. R. Armstrong,D. Hensgen and T. Kidd. The relative performance of various mapping aigorithms is independent of sizable variances in run-time predictions. In 7th IEEE Heterogeneous Computing Workshop(HCW'98). Mar. 1998: 79~87
2. R. Freund, M. Gherrity. Scheduling resource in multi-user, heterogeneous, cornputing environments with SmartNet. In7th IEEE Heterogeneous Computing Workshop. March 1998: 184~199
3. Ding Qing, Chen Guoliang, Gu Jun. A Unified Resource Mapping Strategy in Computational Grid Environments. Journal of Software. 2002, 13(7): 1303~1308
4. Cao Huaihu, Yu Zhenwei, Xu Shoulin. Job Scheduling Algorithm Research on Grid Environment. Computer Engineering and Applications. 2004, 40(5): 87~90
5. Fujimoto N., Hagihara K.. Near-Optimal Dynamic Task Scheduling of Precedence Constrained Coarse-Grained Tasks onto a Computational Grid. In: Parallel and Distributed Computing, 2003. Proceedings. Second International Symposium, October 2003: 80~87
6. J. Luo, P. Ji, X. Wang, Y. Zhu, F. Li, T. Ma. Resource management and task scheduling in grid computing. Computer Supported Cooperative Work in Design, 2004. Proceedings. The 8th International Conference on Volume 2, May 2004: 431~436
7. Yu Jiahua. Yu Xianzhan. A mechanism of predicting network simulation performance based on BP nets. International Conference on Control, Automation and Systems Engineering, CASE 2009:78-82