ISSN No. 0976-5697

Volume 5, No. 5, May-June 2014

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

An Approach Towards Grid Computing to Include Priority Based Min-Min Task-Scheduling Algorithm

Arunima Singh
SRM University, Modinagar
India

Ashish Gautam
IIIT – Allahabad
India

Abstract: Grid computing provides the opportunity to use the resources in network to resolve the large scale tasks. The grid is globally distributed & consists of heterogenous loosely coupled data and resources. Grids use the resources of connected computers on the network and uses the results of these resources to easily perform complex calculation. Effective & efficient scheduling algorithm is needed for capabilities of large distributed systems. In this paper a new task scheduling algorithm "Min-Min task scheduling algorithm" has been introduced.The proposed algorithm tries to adopt the assets of this main algorithm and avoid its drawbacks by considering the priority concept . A proposed algorithm like Min-Min, estimates the completion time of the tasks on each resource and then selects the convenient resource for scheduling.The experimental outcome represents that the proposed algorithm enhanced overall minimum completion time of scheduling as compared to Min-Min algorithm.

Keywords: Grid computing, task scheduling algorithm, min-min task scheduling, completion time.

I. INTRODUCTION

In this paper, we introduce the method for reducing the overall completion time for the job for optimizing task scheduling algorithm in the distributing system based on priority.

Our paper proposed a new task scheduling algorithm to resolve the problem with applying min-min algorithm based on priority to scheduling .Grids use the connected computer resources to the network and used the result of these resources for performing complex calculation easily.

In grid computing, resources are allocated in 2 steps:-

a. ***Resource Discovery***:- It is the process of searching for all the resources available on the network.
b. ***Resource Selection***:- To collect the information of the resources and to choose the best set based on requirement.

The idea is to arrange all the tasks based on their priority which will be predefined and to sort all the jobs (tasks) in ascending order of their execution time so that we can reduce overall completion time for the jobs. Meta task is the independent set of tasks which is considered for mapping at mapping events.

II. RELATED WORK

Scheduling algorithm has been proposed for optimization of the available resources on the network.These algorithms assign tasks to resources and gives the best quality of services based on some criteria.

These algorithms use some parameter for arranging the task to be executed based on minimum execution time, but these algorithms did not consider any priority for the scheduling of the jobs or tasks.Hence, if we execute the jobs based on their priority then it will enhance the performance of the system.

III. EXISTING MIN-MIN TASK SCHEDULING ALGORITHM

In Min-Min Task scheduling Algorithm, all the set of tasks are unmapped.The machine selects the jobs that have the minimum completion time.Then we map selected job with overall minimum completion time to that resource. After that ready time of the resource is updated. Until all the un-mapped tasks are assigned, we repeat this process [1].

When the number of short tasks is more than the long task, Min-Min Algorithm gives poor results. For example, if there is one long task,then Min-Min algorithm assigns the short tasks before the long tasks, due to this makespan is increased[2] .Makespan means entire completion time for the fix number of jobs.In this certain situation,mapping the longest task to the fastest resource provides a better opportunity for concurrent execution of small task on different resources. The time complexity of existing Min-Min algorithm is $\mathbf{O\ (T^2R)}$ where **R** is resource and **T** is task [3].

The pseudo code of Min-Min Algorithm is depicted in Figure1.First of all, the tasks should be in ascending order , It means tasks with minimum completion time is in front of the queue and task with maximum completion time should be in the rear of the queue. Now Min-Min Algorithm computes the minimum completion time of all tasks on available resources [4]. The resource should be chosen according to the appropriate condition. In fig. 1 first of all it computes the amount of task completion time CT_{ij} for all tasks in MT on all resources by the following equation:

$$CT_{ij} = ET_{ij} + r_j \qquad (1)$$

Here CT_{ij} is the completion time of task t_i on resource t_j and ET_{ij} is completion of the task i_{th} on resource j_{th} and r_j is ready time for resource j_{th} . **or R_j is available time of resource j after completing previously assigned jobs**.

After that we calculate average completion time (ACT) and standard deviation (SD) from the following equations:

$$\mathbf{ACT} = \frac{\sum_{i=1}^{r} CT_{ij}}{r} \qquad (2)$$

$$\mathbf{SD} = \sqrt{\frac{\sum_{i=1}^{r}(CT_{ij}-ACT)^2}{r}} \qquad (3)$$

Where **'r'** represent the number of resources

This existing algorithm compares the values ACT and SD.

From equation (2) and (3),two cases arise:

a. If **ACT >SD** , it means if all tasks in MT are in small range then we will select tasks from the front of the queue to assign the next task.
b. If **ACT< SD ,**we will select task rear of the queue to assign the next task.

a) **Sort** all tasks in **MT** ascending //**MT**= Meta Task.
b) **while** there are tasks in **MT.**
c) **for** all tasks $\mathbf{t_i}$ in **MT.**
d) **for** all machines $\mathbf{m_j}$.
e) $CT_{ij} = ET_{ij} + r_j$ // r_j =Ready Time
f) **for** all tasks t_i in MT.
g) Find the minimum CT_{ij} and Resource m_j .
h) **if** there is more than a resource that obtains it.
i) Select a resource with **minimum Completion time.**
j) Calculate **average completion Time & standard deviation** for **Time & standard deviation** for Tasks in **MT.**
k) **if** ACT>SD **then** // ACT =average of Completion Time ,SD = standard Deviation.
l) Assign t_f to resource m_x that Obtains $\mathbf{CT_{fx}}$ // $\mathbf{t_f = t_{front}}$.
m) **else**
n) Assign t_r to resource m_x that Obtains $\mathbf{CT_{rx}}$ // $t_{rx} = t_{rear}$.
o) **end if**
p) delete assigned task from **MT.**
q) **end while**

Figure.1.The pseudo code of an existing algorithm.

IV. PROPOSED MIN-MIN TASK ALGORITHM

The proposed method is to use the concept of priority in the existing Min-Min scheduling algorithm so as to reduce the waiting time of the task in Min-Min scheduling algorithm so as to reduce the waiting time of the task with larger completion time [5,6].

In the existing Min-Min algorithm, the priority of the incoming tasks is not considered due to which the waiting time of the task with large completion increased.The idea of this algorithm is to reduce the waiting time of the larger jobs. Time complexity of proposed algorithm is **O (T^2 R)** where **R** is resource and **T** is a task.

Our proposed scheduling algorithm is presented in Fig.2. Firstly arranges all the tasks based on the priority.Secondly, all the tasks should be sorted in ascending order [7,8].It means tasks with minimum completion time are in the front of the queue and tasks with maximum completion time are in the rear of the queue.After that loads the resources to start execution based on the minimum completion time then assign highest priority tasks from resources[7]. To choose a task for scheduling,

We compute average completion time (ACT) and standard deviation (SD) from equation (2) and (3).

After, we compare the values of ACT and SD by applying two cases (a) and (b) from existing algorithm:-

.

c. If ACT>SD, we will select task from the front of the Queue to assign the next Task
d. If ACT<SD, we will select task from the rear of the queue to assign the next task.

a) **Sort** all tasks in MT ascending order// MT=Meta Task.
b) **while** there are tasks in MT.
c) **for** all tasks t_i in MT.
d) **for** all machines m_j.
e) $CT_{ij} = ET_{ij} + r_j$ // r_j= Ready Time.
f) **for** all tasks t_i in MT.
g) **Enqeue** t_i in priority queue based on priority t_i.
h) **for** highest priority queue t_i.
i) **Find** the minimum CT_{ij} and resource m_j .
j) **if** there is more than one resource that obtain it.
k) Select the resource with least Usage (minimum completion Time).
l) Calculate the **Average Completion Time & standard Deviation** of all tasks in MT.
m) **if** ACT>SD **then** // ACT= average of Completion Time , SD= Standard deviation**.**
n) Assign t_f to resource m_x that Obtains CT_{fx} // $t_f = t_{front}$.
o) **else**
p) Assign t_r to resource m_x that obtainsn CT_{rx} // $t_r = t_{rear}$
q) **end if**
r) Delete assigned task from MT.
s) **end while**

Figure.2. The pseudo code proposed algorithm.

V. AN ILLUSTRATIVE EXAMPLE

Assume there is a grid environment. The completion time of the tasks is depicted in table 1.

Table 1. Completion Time and Priority of tasks

Task	Priority
2	1
4	1
6	2
10	3
5	1
8	2
11	1
14	2
16	3
20	2
30	3
32	3

Fig.2. Draw three queues based on the priority, It means tasks with the highest priority is in on the front of the queue.Also, all the tasks should be sorted in ascending

order, It means tasks with minimum completion time is in the front of the queue and tasks with maximum completion time is in the rear of the queue.

Queue 1 :sorted tasks based on priority 1

2	4	5	11

Queue 2 :sorted tasks based on priority 2

6	8	14	20

Queue 3 :sorted tasks based on priority 3

10	16	30	32

Figure 3. Sorted tasks based on the priority

We load three resources (R_1,R_2,R_3) to start the proposed algorithm. Then assigns three highest priorities from queue 1 and two highest priorities from queue 2 and one highest from queue 1 and repeat this process until all the tasks are mapped into resources.Then we calculate ACT and SD from equation 2 and 3 . After that, apply case (a) and case (b). From existing Min-Min algorithm.

Resource1:Makespan of proposed algorithm on R_1

2	11	16

0 2 13 29

Resource2: makespan of proposed algorithm on R_2

4	6	14	30

0 4 10 24 54

Resource3: make the span of the proposed algorithm on R_3.

11	8	10	32

0 11 19 29 61

Figure 4:Makespan of proposed algorithm on resources

VI. CONCLUSION AND FUTURE WORK

In this paper, we have developed priority based scheduling algorithm based on the Min-Min task scheduling algorithm.The proposed algorithm uses the concept of priority to reduce the waiting time of tasks with larger completion time.New algorithm is introduced to a selection of task of scheduling.The proposed algorithm uses the advantages of Min-Min algorithm and covers their drawbacks.[9,10].

The proposed algorithm gives the better makespan than the Min-Min algorithm. For the future, we can compute minimum completion time by applying other issues like deadlines on tasks and resources.

VII. REFERENCES

[1]. Soheil and Anousha (2013).”An Improved Min-Min task Sheduling Algorithm in Grid Computing”(Springer).

[2]. George Amalarethinam. D. I, Vaaheedha Kfatheen.S (2012). Max-min Average Algorithm for Scheduling Tasks in Grid Comping System “. (IJCSIT) International Journal of Computer Science and Information Technologies.

[3]. Liang Yu, Gang Zhou, Yifei Pu. (February 21,2012)." An Improved Task Scheduling Algorithm in Grid Computing Environment". (IJCNS*) Int.J.Communications, Networks Systems Sciences* .

[4]. Zhongyuan Lee1, Ying Wang[1], Wen Zhou[2](2011). “A dynamic priority scheduling algorithm on service request scheduling in cloud computing”.IEEE.

[5]. T.Kokilavani, Dr. D.I. George Amalarethinam.(April2011).” Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing”.International journal of Computer Applications.

[6]. D. Doreen Hephzibah Miriam and K. S. Easwarakumar. (July 2010). “A Double Min Min Algorithm for Task Metascheduler on Hypercubic P2P Grid Systems”.IJCSI,

[7]. Saeed Parsa and Reza Entezari-Maleki. (2009R ”RASA: A New Task Scheduling Algorithm in Grid Environment”.World Applied Sciences Journal 7.

[8]. Foster, I. (2005).” Globus Toolkit Version 4: Software for Service-Oriented Systems” Springer.

[9]. HE Xiaoshan[1], Xian-He Sun[1] Gregor von Laszewski (May, 2003).”QoS Guided Min-Min Heuristic for Grid Task Scheduling”. National Science Foundation of USA under NSF Grant Nos.EIA-0224377, ANI-0123930, EIA-0130673.

[10]. Fangpeng Dong and Selim G. Akl. (January 2006).” Scheduling Algorithms for Grid Computing: State of the Art and Open Problems”. Technical Report No. 2006-504.