ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 2, February 2019

Reputation Based Dead Line Scheduling in Grid Computing Environment

R. Ananthi Lakshmi[1], R.RaviChandran[2]

Ph. D Scholar, Dept of Computer Science, KG College of Arts and Science, Coimbatore, India[1]

Director, KG College of Arts and Science, Coimbatore, India[2]

Abstract: Grid is a kind of distributed and parallel computing. Mainly it is used to solve complex problems such as weather forecasting, earth observation and financial modeling (etc). Hence we propose the reputation based deadline scheduling in grid, which is to be consider important factor for resource selection. In this paper, reputation based deadline scheduling is based on select for solving a complex task within particular schedule. Most existing reputation models used for reliability evaluation ignore the time influence. Based on Reputation Prioritized Based Deadline Scheduling Algorithm (PDSA) using average turnaround time. PDSA has shown optimal performance as compared to EDF and RR scheduling algorithm.

Keywords: Reputation, Deadline, Grid, Resource

I. INTRODUCTION

Grid is a collection of different nodes where in all of them contribute any combination of resources. The basic idea of Grid Computing is to create a large and powerful virtual computer which is a collection of heterogeneous distributed environment. Job Scheduling is used to choose the most suitable resource for a job to complete its execution based on the waiting time, turnaround time. The basic idea of Grid is one such technology providing solution to the industry expectation by the way of resource sharing and allocation .The reputation is an important part in evaluating the deadline scheduling. It collects a particular resource from the group of users.

Reputation Based Deadline Scheduling is more effective than normal way of scheduling to enable reputation, two important issues need to be considered.(i)How to evaluate the reputation and (ii)How to perform reputation based on deadline scheduling. Reputation systems are commonly used to evaluate reliability. Firstly, most reputation according to its ratio of successfully completed tasks. Secondly, from the task perspective to all the tasks on a resource based on the resource reputation. Several list of algorithm have been proposed for this problem. Usually Prioritized Based Deadline Scheduling Algorithm (PDSA) can provide a better quality solution to list of algorithms. Although PDSA is more time consuming. It is acceptable for applications with long runtime. Furthermore Grid Computing stands out as the principle occurring for several years of time. Simply by concentrating on virtual organizations to be able to share large scale resources innovating applications and perhaps acquiring high performance orientation. Most of the scheduling algorithms have not considered deadline perspective for job execution. PDSA has been proposed to meet deadline constraints.

This motivated us to design reputation service for Grids to assist in the selection process for resources .Reputation based service and product selection has proved to be a great asset for online sites such as eBay and Amazon. We believe that such reputation service framework is importance for Grid Computing to increase reliability, utilization and popularity. Grid environment that agglomerates expensive and specialized resources including high-performance servers, storage databases, advanced scientific etc.

II. REPUTATION AND DEADLINE

In this section we define the basic terminology that will be used throughout the rest of the paper.

A. REPUTATION

The reputation of an entity is an expectation of its behavior based on other entities. It includes agents, services and persons in the grid. Resource reputation provides a way of assigning quality or value in regards to a resource. If a resource is known to provide certain qualities over a period of time irrespective of its limitations, then it is assumed to have good limitations.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 2, February 2019

B. DEADLINE

In deadline the processes are dispatched based on minimum deadline on the ready queue. When a process has completed its task it will be terminated and then the next job with minimum deadline will be dispatched from the ready queue.

III. RELATED WORK

Algorithm 1 RD Reputation Calculation Algorithm for each resource r_i do

```
1   rdri □ rdri0 □ rdrinitial
2   ti ← 1
3   si □ fi □ current _ time
4   runtimei ← 0; ci ← 0
5   end for
7   while there is a reputation record testimony ij do
8   if ( f i □ s □ T ) then //current interval
    j     i     windows
9   ci ← ci □ cij
10  runtimei ← runtimei □ ( f ji − sij )
11  fi ← max( f ji , fi )
12  Remove the record testimonyij
13  Compute λstatistici by Equation 2
14  Compute rdr i by Equation 3
15  else//next interval
16  rdriti ← rdri
17  ti ← ti □ 1
18  si □ fi □ si □ Twindows
19  runtimei ← 0; ci ← 0
20  end if
21  end while
```

The real time resource reliability can be monitored by the resource's reputation, which can be defined as the probability that the resource can deliver the expected utility service [2]. Neither the normalized number nor the ratio of correct responses considered the time influence [3].PDSA used to evaluate the reputation. Although task runtime is included in their model, they did not specify how the task runtime affects the reputation. The time related performance can also be evaluated by the resource availability [4].

Moreover, most existing works did not give methods or algorithms to predict the real time task failure rate for a resource, which is needed for task scheduling. However our reliability-driven reputation is specially defined to be time dependent, and our reputation calculation algorithm can provide from one of the remote host and it will record the job information in the job table and update its accumulative successful execution rate. If a job scheduler receives a job completion message from one of the remote host, it will record the job information in the job table and update its information.

IV. PROPOSED JOB SCHEDULING

Prioritized Deadline Based Scheduling Algorithm (PDSA): This algorithm executes the job with the closest deadline time delay in the cyclic manner using a dynamic time quantum. Based on our algorithm perform the allocation for a single processor based on the deadline criteria dependent on the minimum time delay of job execution, turnaround time, waiting time and maximum tardiness.

Basic definition of the aforementioned criteria:
Let us assume J_i : ith Job;
n: the number of jobs;
TQ_i : time quantum of job i;
T_i : arrival time of job i;
d_i: deadline of job i;
α_i: burst time of job i;
$LCM(\alpha_{1\ to}\ \alpha_n)$: lowest common multiple of overall burst time of job i;

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 2, February 2019

C_i: Job completion time of job i;
T_{TRi}: turnaround time of job i;
T_{WTi}: waiting time of job i;
T_{TDi}: time delay of job i;
T_{TRDi}: tardiness of job i;
T_{Max_TRD}: maximum tardiness;
S-list: Sorted list;

1.Time delay: Referred to the time difference between burst time and deadline time.

$T_{WTi} = T_{TRi} - {}_i$.....(1)

α (1)

2.Time quantum: referred to a fixed time for each job to be executed in cyclic manner meaning when a job has completed its task, i.e. Before the expiry of the time quantum, it terminates and is deleted from the system. The next job is then dispatched from the head of the ready queue.

Time quantum, take = LCM ($\alpha_{1 \text{ to } n}$).....(2) (2)

3.Turnaround time: Referred to the total time taken between the submission of job for execution and the return of the completed result.

Turnaround time $T_{TRi} = C_i$ .....(3)

Average turnaround time,

$$T_{Avg\_TR} = \frac{\sum_{i=1}^{n} T_{TRi}}{N}.....(3)$$

4. Waiting time: Referred to the total waiting time of job before its final execution.

$T_{WTi} = T_{TRi} - {}_i$.....(4) (5)

Assign time quantum, by computing *LCM* of all burst time, and then compute the value of time delay for each job by sorting out the jobs on the basis of time delay in ascending order, then selecting the jobs with minimum time delay for execution. If multiple jobs have same time delay value then, it will break the tie by selecting a job from job set on the basis of FCFS .The algorithm dispatches jobs from the head of the ready queue for execution the CPU. Jobs being executed are preempted based on a time quantum. A Preempted Process Control Block (PCB) is linked to the tail of the ready queue. When a job has completed its task, i.e. before the expiry of the time quantum, it terminates and is deleted from the system. The next job is then dispatched from the head of the ready queue. After each cycle a new time quantum will be assigned, by computing *LCM* of all remaining burst time of jobs. The value of turnaround time waiting time and tardiness for each job are computed. The algorithm computes the average turnaround time each user's job, average waiting time each user's job and finally compute the maximum tardiness value for jobs to identify the maximum time delay of jobs execution.

V. RESULT AND DISCUSSION

(A) Average Turnaround Time

Our simulator has been used to carry out extensive experimentation using the Windows 7 operating system on an Intel Core4 Duo. We used Monte Carlo method for process set generation in our experiments. The simulations of the algorithms have generated useful data that has been analyzed. To check the performance of the proposed algorithms, i.e. PDSA scheduling algorithm, FCFS scheduling algorithm and RR scheduling algorithm.

We assume 50 numbers for CPU time for various job, we have taken this CPU time values in 10, 100 and 1000 showing the heterogeneous demands of the user›s jobs, each with different characteristics, and ran them through the simulator. Each job is specified by its CPU burst length, arrival time and priority number. Each job set has been given a time quantum for simulation. Performance metrics for the CPU scheduling algorithms are based on the following

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 2, February 2019

factors - Average Turnaround Time, Average Waiting Time and Maximum Tardiness. The traditional reputation based task failure probability gets close to the standard task failure probability only when the test tasks in the system also have the medium task size. Otherwise, the failure probability also increases as the size of the test tasks increases. And when the resources have a faster speed or lower failure.

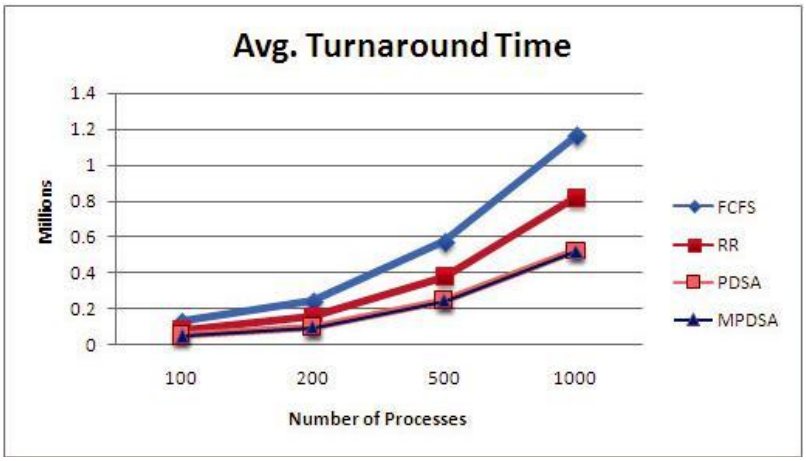

Fig.1 shows that, PDSA has the best performance as compared to FCFS and RR under variable and scalable workload.

(B) Average Waiting Time

CONCLUSIONS AND FUTURE WORK

In this paper, a scheduling algorithm for executing jobs on grid systems is proposed. Just like real-life scenarios, we have considered the dynamic arrival of jobs as well as the deadline requirement of each job to be processed. The experiment has been performed by varying workload, by increasing jobs from 100 to 1000 in a scalable manner. The result has shown maintained performance under dynamic environment. Based on the comparative performance analysis PDSA has shown the best performance as compared to FCFS and RR scheduling algorithm under variable and scalable workload. We have developed a new simulator using Java language to facilitate this research. This has been input simply by extensive experimentation. Various possible input patterns were experimented with all the CPU scheduling algorithms. We can say that PDSA is a scheduling policy from the system point of view; it satisfies the system requirements (i.e. short Average Waiting Time and short Turnaround Time) and also supports scalability under heavy workload. In the future, we will evaluate and propose a computational scheduling algorithm on the grid based on multiple processors and perform detailed comparative performance analysis with other scheduling approaches.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 2, February 2019

REFERENCES

[1]. Goswami& Das, 2015] Goswami S, Das A, "Deadline stringency based job scheduling in computational grid environment", Computing for Sustainable Global Development(INDIACom), 2015 2nd International Conference, ISBN: 978-9-3805-4415-1, pp- 531-536, IEEE March 2015.
[2]. I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell and J. Von Reich, The Open Grid Services Architecture, ersion 1.0, ttp://forge.gridforum.org/projects/ogsa-wg, January 2005.
[3]. E. Caron, P. K. Chouhan, and F. Desprez. DeadlineScheduling with Priority for Client-Server Systemson the Grid. ACM GRID, Nov 2004.
[4]. C. Germain, V. Breton, P. Clarysse, Y. Gaudeau,T. Glatard, E. Jeannot, Y. Legre, C. Loomis, J. Montagnat,J.-M. Moureaux, A. Osorio, X. Pennec, andR. Texier. Grid-enabling medical image analysis. CC-Grid, May 2005.
[5]. A. K. F. Khattab and K. M. F. Elsayed. Channel-Quality Dependent Earliest Deadline Due FairScheduling Schemes for Wireless Multimedia Networks.ACM MSWiM, 2004.
[6]. T. Lam and K. To. Performance guarantee for OnlineDeadline Scheduling in the Presence of Overload.ACM SODA, 2001.
[7]. C. L. Liu and J. W. Layland. Scheduling Algorithmsfor Multiprogramming in a Hard-Real-Time Environment. Journal of the Association for Computing Ma-chinery 20, 1:46–61, Jan 1973.
[8]. A. Sulistio, G. Poduval, R. Buyya, and C. Tham, On Incorporating Differentiated Levels of Network Service into GridSim, Future Generation Computer Systems (FGCS), 23(4):606-615, 2007.
[9]. X. Wang, R. Buyya and J. Su, Reliability-Oriented Genetic Algorithm for Workflow Applications Using Max-Min Strategy, 9th IEEE International Symposium on Cluster Computing and the Grid, 2009.
[10]. M. Wieczorek, S. Podlipnig, R. Prodan, and T. Fahringer. Bi-criteria Scheduling of Scientific Workflows for the Grid. IEEE Symposium on Cluster Computing and the Grid, May, 2008.
[11]. J. Yu, M. Kirley, and R. Buyya, Multi-objective Planning for Workflow Execution on Grids, IEEE/ACM Conference on Grid Computing, 2007.