DOI: http://dx.doi.org/10.26483/ijarcs.v9i1.5269

Volume 9, No. 1, January-February 2018

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

STUDY OF TIMELINE AND PROFILE BASED SCHEDULING IN GRID ENVIRONMENT – A SCOPE TO IMPROVE CLOUD SCHEDULING.

Dr. Bimal V O
Associate Professor
Chinmaya Institute Of Technology
Kannur, Kerala, India

Mr. Bosco Paul Alapatt
Research Scholar
Bharathiyar University
Thamil Nadu, India

Dr. M Anand Kumar
Professor
Adigrat University Ethiopia

Abstract: In scheduling algorithms, some are cost-effective and some others are performance based. CPU intensive problems are very much increased and the research on effective scheduling is going deeply and widely. In this scenario, the scheduling problem is very interesting and this is the perfect time to develop a new scheduling strategy. Here we are introducing a new approach to reduce the total waiting time of any job submitted to the grid. The effective use of the idle computing powers scattered across the world is implemented and analyzed in this paper. We also portrait the possibilities of the same in the cloud environment.

Keywords: grid, cloud, scheduling, cluster, timeline, profile, distributed,

I. INTRODUCTION

Grid computing is the sharing of computational resources to execute a big task especially a scientific problem like DNS analysis or climate forecasting. In the real scenario, when a grid is implemented and configured, it should satisfy several requirements. These requirements are mostly user perspective. To effectively use the grid computing capabilities to the requirements, it is important to keep in mind the reasons for using grid computing. A good infrastructure is required for solving big scientific problems. The enhancement of grid computing infrastructure will indirectly improve the research activities in all scientific branches because many scientific problems need huge computing power. Computer aided research is required in simulated studies and it is unavoidable because many real type of equipments used in studies are costly [1].

If we have an existing application in a machine and that application is migrated to another machine and run that application remotely, is the easiest use of grid computing. Sometimes the computer on which the application is used to run might be busy due to a heavy traffic on the network or there is a high demand for processing power[2]. In such a case the job supposed to run on that computer should migrate to some other computer in the grid which is idle or having less network traffic. Obviously, now we have at least two requirements. The first requirement is the application should be executable remotely without any network or process overhead. Secondly, the remote machine should meet any special software, hardware or resource requested by the application. So an efficient algorithm and plan might be required to efficiently use the grid technology[3]. So some applications like word processor in a grid using a remote machine's computing power may cause a delay in responding due to network failure or conjunction occurred in the network. For all these reasons, job Scheduling is a hot topic among researchers. Job Scheduling is the process of selecting the CPUs suitable for executing a specific job. In a simple grid environment, the process of scheduling involves users manually selecting the machines suitable for running jobs and proper commands that send these jobs to the CPUs. A more advanced grid technology may consist of complex job scheduling algorithms which are capable of automatically performing these scheduling tasks on behalf of the user[4].

In this context, the research on scheduling problem to schedule a stream of applications from different users to a set of computing resources to maximize the system utilization is an emerging trend[5]. The usage in grid technology in the business world is increasing. So the economically feasible algorithms will increase the demand and popularity.

II. MOTIVATIONS

When we consider the actual grid environment, the resource demand and the availability jointly determine the cost of computing power. As per our observations, the timeline of the country is also involved in this cost because the computing power required for the day time will be peak and it is called the peak time in that country. The job is transferred to other region based on the threshold value, which is priorly calculated by the admin and the time line of that region. Moreover identifying the suitable algorithm based on the profile of the job improves the scheduling and it is a major challenge[6].

In recent years, various algorithms are introduced in scheduling but the efficiency of these algorithms varied based on the different factors like computing environment and computational requirement but the objective of timeline algorithm is to reduce the waiting time of jobs by migrating jobs to other countries. The simple migration techniques may not be an efficient strategy because many counties suffering from resource pool saturation. Another problem is migrating low resource consuming jobs. So it is better to execute such jobs locally[7] because there is no advantage in migrating such jobs when we consider cost and network delay.

As discussed above, another objective is to develop a job-profile based selection of scheduling algorithms that consist of TimeLine strategy too. So some important parameters of the job will be selected and create a profile for the job. The job profile plays an important role for the selection of appropriate scheduling algorithm[8].

III. TIMELINE ALGORITHM IN GRID

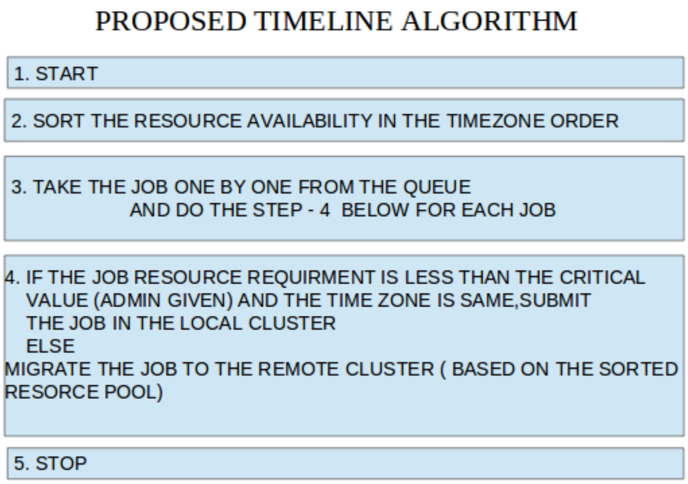

Figure 1. Steps involved in Timeline Algorithm

Figure 1 shows the algorithm of Timeline. Here the sorting of the resource availability will be done initially. The sorting criteria is based on the timezone region in the world map. Now the selection of job one after another will occur. There is a critical value fixed by the admin based n the environment. If the resource requirement is less than this critical value d the time zone is same, the job will be submitted in the local cluster. So it avoids all network overheads and guarantees the fastest execution. If this condition is failed the job will be migrated to the remote cluster. As mentioned above the resource will be taken from the sorted pool.

There are many effective scheduling algorithms developed by the researches worldwide in the area of grid computing. The main disadvantage of such algorithms is that they are not considering the job profile. The profile of a job consists of the average processing time of N batch jobs, time stamp of job arrival and number of jobs in the queue. Figure-1.1 shows the parameters inside the Job profile. Average processing time is calculated using equation -1

$$Tav = \frac{\sum_{i=1}^{N} PTi}{N} \quad \textbf{(1)}$$

In the above equation

N = the total number of jobs

PT = Processing time.

Here jobs can be submitted as a single job or a group of jobs, say batch. In the case os a batch job, each job has its own processing time and job number. The average processing time required is calculated from the total resource capacity of particular grid[9].

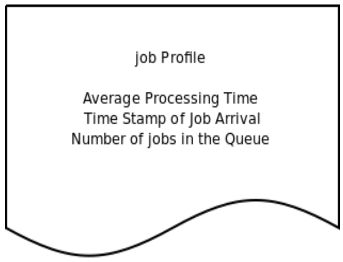

Figure 1.2. Job profile

Table I. Job Profile based selection

Algorithm – Job Profile based selection
1: start
2: compute average processing time ,avp
3: Set threshold average processing time=tavp
4. Set threshold number of jobs=tj
5: Local time stamp = t , number of jobs=n
6: if timestamp is greater than 8.00 and less than 23.00
select TimeLine() policy
go to step 9
endif
7: if avp<tavp and j<tj
select FCFS() policy
go to step 9
endif
8: select EDF() policy
9: stop

If the average processing time and the number of jobs are under the assigned threshold value, the proposed grid scheduler will select conventional FCFS algorithm, otherwise, it will select other policies. Table - 1.1 shows the algorithm for Profile based selection of Scheduling.

IV. RESULT AND DISCUSSION – ANALYSIS OF TIMELINE ALGORITHM

When analyzing the experimental result of jobs equal to 12000, there is a slight waiting time for TimeLine jobs (Figure- 1.5). When compared to other two algorithms (Figure 1.3 and Figure 1.4) this waiting time is negligible. It is noted that there is a subsequent increase in waiting time for jobs greater than 12000, but the TimeLine algorithm reduces the waiting time effectively.

Figure 1.3. Performance analysis of FCFS when 12000 jobs

It is also noted that when the total no of jobs submitted is equal to 11000, all jobs are completed round 67 days except EDF algorithm. EDF took around 75 days to complete the job. In the case of 12000 jobs, it was around 67, 75 and 67 days respectively. So there is no increase in the total execution time for TimeLine strategy when compared to other existing algorithms.

Figure 1.4. Performance analysis of EDF when 12000 jobs

Figure 1.5. Performance analysis of Timeline when 12000 jobs

The above experiment is repeated for jobs 13000 and 14000 and found that the waiting time is much better in TimeLine algorithm compared to others.

We have conducted more experiments using Timeline algorithm by comparing other algorithms like PBS-PRO, CONS and bestgap. We have changed various parameters such as the total number of jobs and analyzed the performance of all algorithms. From the Figure 1.6 to 1.9 it is evident that timeline algorithm creates less waiting time. The waiting time is less when the number of jobs are less and there is a subsequence increase in the waiting time when we increase the number of jobs. Anyway compared to other algorithms, there is a significant advantage for timeline algorithm.

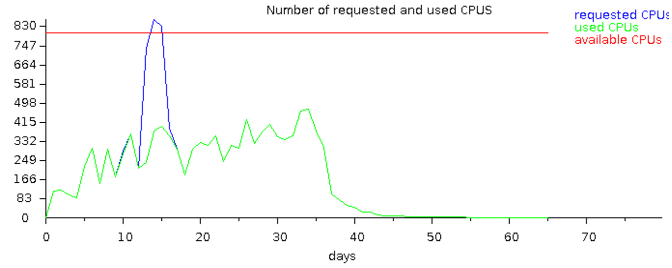

Figure 1.6. Performance analysis of CONS when 10000 jobs

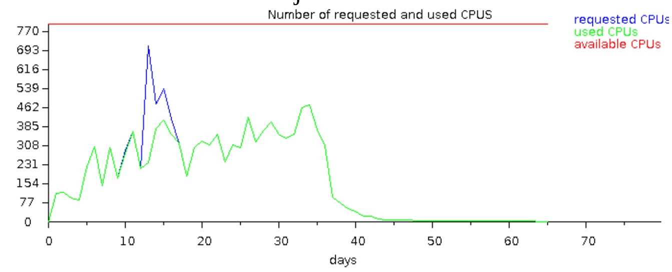

Figure 1.7. Performance analysis of PBS_PRO when 10000 jobs

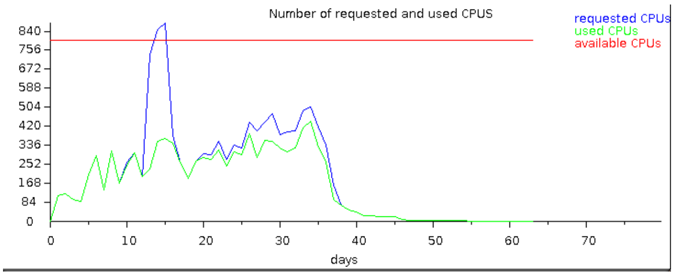

Figure 1.8. Performance analysis of Bestgap when 10000 jobs

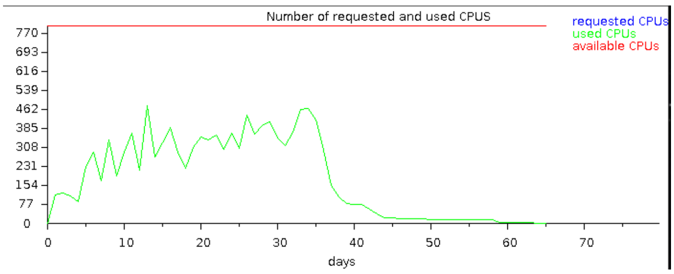

Figure 1.9. Performance analysis of Timeline when 10000 jobs

When we look the above figures, it is obvious that the total execution time is equal in all cases but there is an advantage of significant improve in job waiting time.

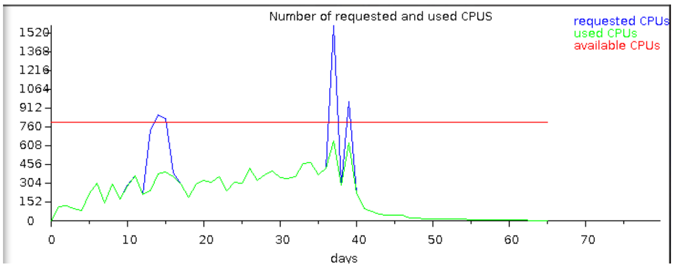

Figure 1.10. Performance analysis of CONS when 15000 jobs

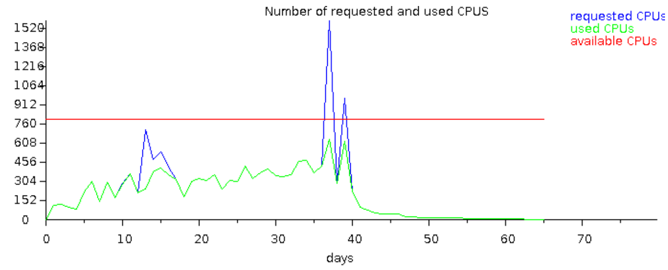

Figure 1.11. Performance analysis of PBS_PRO when 15000 jobs

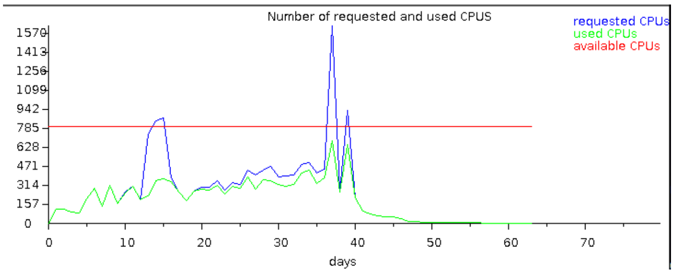

Figure 1.12. Performance analysis of Bestgap when 15000 jobs

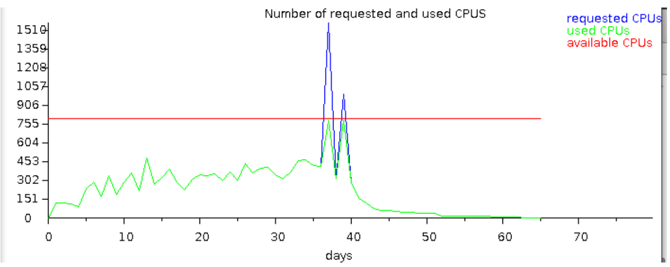

Figure 1.13. Performance analysis of Timeline when 15000 jobs

V. ANALYSIS OF JOB PROFILE BASED SELECTION

When we analyze the job profile based selection of scheduling algorithm. The grid scheduler will schedule the job for TimeLine when the average processing time and the number of jobs are above a threshold value. This threshold value can be fixed by the grid administrator manually. Timeline algorithm will work only during day time. If the Timestamp gives the night time, this algorithm never executes. If the number of jobs and average processing time are under a threshold value the scheduler will select FCFS. If the conditions of FCFS and TimeLine are not met, the scheduler will select EDF algorithm.

The performance of the algorithms was analysed by varying the different parameters such as total number of jobs, average processing time and the timestamp of the job. The Figures-1.14, 1.15 and 1.16 show the waiting time for FCFS algorithm, EDF algorithm and TimeLine algorithm respectively. In the first experiment we set total number of jobs equal to 500 and the time stamp of the job at night.

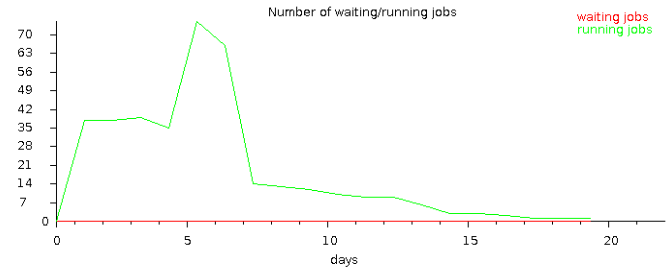

Figure 1.14. Waiting time graph of FCFS when 500 jobs

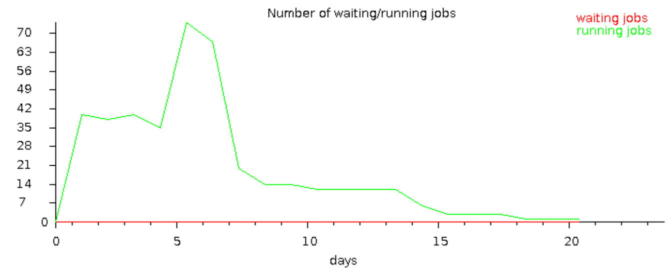

Figure 1.15. Waiting time graph of EDF when 500 jobs

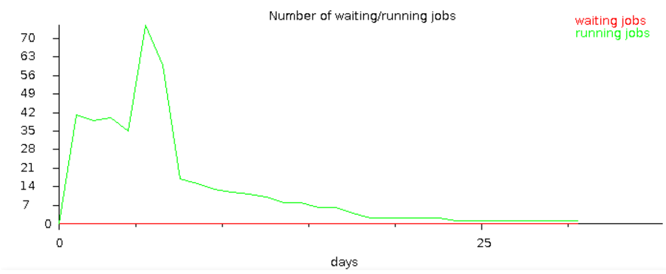

Figure 1.16. Waiting time graph of Timeline when 500 jobs

By analysing these graphs it is seen that FCFS algorithm is better when we submit less number of jobs, say 500. There is no waiting time for other algorithms also but when we consider the total execution time FCFS has advantage. In the above experiment with 500 jobs, FCFS finished the execution in 17 days whereas EDF took 22 days and TimeLine tooks 32 days. So in this particular scenario the best suitable algorithm is FCFS. In FCFS, jobs are executed first come first serve basis, so many instances of time, the clusters are not free and it creates a huge waiting time but in the case of EDF, It is noted that the high-cost local resources are busy and the low-cost resources are free[10].

Here the earliest deadline jobs get high priority. So when we implement the new approach it is cost effective because most of the computers are used during nignt, which is an off peak time.

VII. SCOPE IN CLOUD

Nowadays, cloud computing is a buzzword in business and industry. Lot of research works are going on in this field. In short words, the cloud is the business version of the grid. Cloud provides services such as infrastructure, platform, software, network, storage etc whereas grid is a collection of CPUs mainly dedicated for large scientific applications like DNA analysis or climate prediction. All the algorithms developed in the grid can be adopted for clouds with little or no modifications. So Timeline algorithm and profile based selection of algorithms can be implemented in cloud.

VIII. CONCLUSION

We have many effective scheduling algorithms developed by the researches but unfortunately, such algorithms are not considering the job profile and timeline. This paper discussed the simulated study of profiling, timeline strategy and combination of both to prove the importance of job profile based selection and timeline in scheduling algorithms. The same work can be adopted to the Cloud environment also. Further studies may require for the same.

IX. ACKNOWLEDGMENT

We would like to show our gratitude to the management and staff of Chinmaya Institute of Technology for sharing their pearls of wisdom with us during the course of this research, and we thank all reviewers for their so-called insights.

VIII. REFERENCES

[1]. A. Grimshaw and W. Wulf, 1997. The Legion Vision of a Worldwide Virtual Computer, Communications of the ACM, vol. 40.

[2]. Alexandre di Costanzo, Marcos Dias de Assuncao, 2009. Harnessing Cloud Technologies for a Virtualized Distributed Computing Infrastructure, IEEE Internet Computing, Volume 13, Number 5, Pages: 24-33.

[3]. Srikumar Venugopal, Rajkumar Buyya and Lyle Winton, 2006. A Grid Service Broker for Scheduling e-Science Applications on Global Data Grids, Concurrency and Computation: Practice and Experience, Volume 18, Issue 6, Pages: 685-699.

[4]. Anthony Sulistio, Uros Cibej, Sushil Prasad, and Rajkumar Buyya, 2009. GarQ: An Efficient Scheduling Data Structure for Advance Reservations of Grid Resources, International Journal of Parallel, Emergent and Distributed Systems, Volume 24, Number 1, Pages: 1-19.

[5]. Balaji, N.. Enhancing Computational Speed for Search Application Through High Performance Grid \Using Globus Tool Kit, International Journal of Computational Intelligence Research/09731873, 20090401.

[6]. Bimal VO, Dr. G.Raju, 2011 “Alea- The efficient job scheduling simulator”- Proceedings in 'National conference on Envisioning the future: Emerging Trends in Management and Information Technology-Aavishkar' Chintech, Kannur, ISBN 978-81-921983-9-2.

[7]. Bimal VO, G Raju, 2014. Performance Analysis of TimeLine Algorithm in Grid Environment using Alea.International Journal of Computer Science Systems Engineering and Information Technology (IJCSSEIT), Vol. 7, No. 2, December 2014, pp. 199-210.

[8]. Bimal VO, G Raju. Performance Analysis of TimeLine Algorithm in Grid Environment using Alea. International Journal of Computer Science Systems Engineering and Information Technology (IJCSSEIT), Vol. 7, No. 2, December

[9]. Bimal VO, G Raju. Job-Profile based selection of Scheduling Algorithms in Grid Environment. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 14 (2015) pp 34479-34484

[10] . C. Waldspurger, T. Hogg, B. Huberman, J. Kephart, and W. Stornetta, 1992. Spawn: A Distributed Computational Economy, IEEE Transactions on Software Engineering, Vol. 18, No. 2, pp 103.