ISSN (online): 2250-141X
Vol. 6 Issue 2, April 2016

GRID COMPUTING:SOFTWARE ENVIRONMENTS AND ALGORITHMIC TECHNIQUES

Dr.Salim Amdani*, Mr.S.R.Jadhao

*Head, Department of Computer Science and Engineering
Babasaheb Naik College of Engineering,Pusad

ABSTRACT:- *Distributed computing environments have been very much in spectacle for the last one and a half decade. But While Widely used high performance large scale Parallel computing platforms for scheduling of jobs problems which exist in existing scheduling strategies. In the Meanwhile, these existing strategies are rarely balance fairness and performance. Aiming at these issues, we reviewed lot of literature lead to unfairness. However, fairness and performance in the large scale parallel system is rarely taken into consideration simultaneously.*

I. INTRODUCTION

Grid computing environment behaves like a virtual organization consisting of distributed resources. It is a set of individuals defined by a definite set of sharing rules like what is shared, who is allowed to share, and the conditions under which the sharing takes place [1]. The needs of the grid range from client-server to peer-to-peer architecture, from single user to multi-user systems, from sharing files to sharing resources, etc. and all these in a dynamic, controlled and secured manner. As grid computing focuses on dynamic and cross-organizational sharing, it enhances the existing distributed computing technologies [2]. The focus of parallel processing research is fast execution of jobs. However, for expensive parallel supercomputers, a job often spends more time in the waiting queue than actually executing. Since the turnaround time of a job is the only figure of merit that matters to a user, it is important to not only improve the execution time of individual jobs but also the waiting time and the overall system performance. The scheduler is a major component for managing the resources of large-scale parallel environments. A policy in a scheduler is used to assign jobs to resources in specific time intervals such that the capacity of resources should meet jobs' needs [3] [4].

II. BASIC SYSTEM OF GRID

Following issues related to Grid Computing having scope to improve: Grid computing is a promising paradigm with the following potential.

A. Exploiting underutilized resources

Studies have shown that most low-end machines (PCs and workstations) are often idle: utilization is as low as 20 percent. And even for servers only 50 percent of their capacity is utilized . A simple case is that we can run a local job on a remote machine elsewhere in the Grid if the local machine is busy.

B. Distributed supercomputing capability

The parallel execution of parallel applications is one of the most attractive features of computational Grids. In Grid systems, there are a large number of computational resources available for one parallel application, such that different jobs within the application can be executed simultaneously on a

ISSN (online): 2250-141X
Vol. 6 Issue 2, April 2016

suite of Grid resources.

C. Virtual organizations for collaboration

Another important contribution of Grid computing is to enable the collaboration among wider-area members. Grid computing provides the infrastructure to integrate heterogeneous systems to form a virtual organization. Under the virtual organization, sharing is not limited to computational resources, but also includes various resources, such as storages, software, databases, special equipments, and so on.

D. Resource balancing

After joining a Grid, users will have a dramatically larger pool of resources available for their applications. When the local system is busy with a heavy load, part of the workloads can be scheduled to other resources in the Grid. Thus the function of resource balancing is achieved. This feature proves to be invaluable for handling occasional peak loads on a single system.

E. Reliability

High-end conventional computing systems use expensive hardware to increase reliability. In the future, Grid computing provides a complementary approach to achieving high-reliability nevertheless with little additional investment. The resources in a Grid can be relatively inexpensive, autonomous and geographically dispersed. Thus, even if some of the resources within a Grid encounter a severe disaster, the other parts of the Grid are unlikely to be affected and remain working well [5].

III. EXISTING METHODS

Year	Policy	Drawbacks
2002	Priority,reservation,Non-FCFS,backfilling,multiple queue [6]	Degrade performance for long jobs.
2004	backfilling and gang scheduling [7]	Delay incurred.
2002	EASY Backfill-ing,Conservative Backfilling using selective reservation [8]	advance reservation will increase Mean Waiting Time and Request Rejection Rate
2006	event-based simulation of EASY scheduling [9]	A backfill scheduler will prematurely kill such jobs
2008	flexible backfilling ,Pri-ority ,FCFS using back-filling [10]	Load increases perfor-mance decreases.
2008	flexible Advance Reser-vation , backfilling [11]	increasing inaccuracy interestingly has op-posing effect on back-filling methods
2010	a relaxed backfill scheduling , multiple reservations [12]	reservation can lead to starvation if more reservations.
2013	a new hierarchical scheduling strategy ,LB and FCFS with backfilling , dynamically groups ARs [13]	possibly delaying other jobs, a job will be killed immediately if it cannot finish within its estimated running time
2012	a hybrid job scheduling mechanism [14]	less requirement then free processors will remain idle. small jobs requiting fewer resources may queue for long and wait longer till large job finishes.
2013	two-dimensional matrix [15]	But still has the wastage and have chance for improvement
2014	a new preemptive venture EASY backfilling (PVEASY),novel shadow load preemption (SLP) and venture backfilling (VB) [16]	Around 20 percent of the blocked jobs in each workload suffered from unfairness.
2014	reservation based First-Fit priority (R-FirstFit) [17]	average response time of R-FirstrFit method is a little longer than FirstFit method
2015	reservation based First-Fit (RF) and feedback based distribution for tasks (FD) [18]	execution of jobs with less recourses demand may not delay the jobs with high recourses de-mand and high prior-ity, which contributes to the less total finish time of jobs

ISSN (online): 2250-141X
Vol. 6 Issue 2, April 2016

IV. GRID SIMULATION TOOLS

Grid Simulation Tools There are many simulation tools for grid computing such as simgrid, gridsim, optorsim, bricks etc available for evaluating scheduling algorithms and network services for grid systems.

A. SimGrid

The SimGrid project was started in 1999 which provides core functionalities for simulation of distributed applications in heterogeneous distributed environments. The main aim of the project was to facilitate research in the area of distributed and parallel application scheduling on distributed computing platforms that were ranging from simple network of workstations to computational grids. Henri Casanova in has used Simgrid for the study of scheduling algorithms for distributed application.

B. GridSim

Rajkumar Buyya et al has developed java based discrete event grid simulation toolkit. The toolkit allows modeling and simulation of entities in parallel and distributed computing systems users, applications, resources and resource broker for design as well as evaluation of scheduling algorithms. Some of the functionalities of gridsim include incorporating failures of grid resources during runtime, supporting advance reservation of a grid system, incorporating auction model, incorporating extension of datagrid into GridSim, incorporating network extension into GridSim etc.

C. OptorSim

A grid simulator designed to test dynamic replication strategies and appropriate scheduling of jobs was developed as a part of European Data Grid project. OptorSim.

D. Bricks

It is a java based performance evaluation system for scheduling algorithms and frameworks of high performance global computing systems. It consists of a scheduling unit that allows simulation of various behaviors of resource scheduling algorithms, programming modules for scheduling, processing schemes for networks and servers etc [19].

V. DISCUSSION

Backfilling [20] optimization technique help to improve system utilization and had been implemented in most production schedulers . Nowadays, a research direction focuses on providing fair scheduling between users in clusters. Backfilling [21] algorithm leverages fairness and performance in a simple and efficient manner. One of the Simulation Environment or Testbed used For this purpose a mathematical model will be formulated and algorithm(s) will be designed and developed. The developed models and methodologies will be integrated within the general Grid scheduling architecture and will be demonstrated under different workload conditions through a simulator- based Grid environment. Evaluate the performance of developed models and methodologies by simulation that can allow the test of a wide range of scenarios and compare the performance of identified and propose approaches on the basis of evaluation parameters namely total completion time, Average waiting time/Average Response time and Throughput.

VI. CONCLUSION

We analyzed various job scheduling policies and also identified the software requirement for simulation of scheduling different jobs strategies. Hence it is required a efficient method to address these issues more closely and fairly so that we can utilize Grid Computing resource in an efficient and fair manner to improve the performance of the system. Our initial research shows that it is worth investigating the potential impact on the performance of the Grid when efficient optimization techniques are applied to scheduling policies using available simulations tools or testbeds to implement our ideas and test, compare the results.

ISSN (online): 2250-141X
Vol. 6 Issue 2, April 2016

REFERENCES

I. Bakri Yahaya, Rohaya Latip, Mohamed Othman, and Azizol Abdullah. Dynamic load balancing policy in grid computing with multiagent system integration. In Software Engineering and Computer Systems, pages 416–424. Springer, 2011.

II. B. Bansal and S. Bawa. Design and development of grid portals. In TENCON 2005 - 2005 IEEE Region 10 Conference, pages 1–5, Nov 2005.

III. Ranieri Baraglia, Gabriele Capannini, Patrizio Dazzi, and Giancarlo Pagano. A multi-criteria job scheduling framework for large computing farms. Journal of Computer and System Sciences, 79(2):230 244, 2013. 10th {IEEE} International Conference on Computer and Information Technology, 2010.

IV. Preeti Gulia and Deepika Nee Miku. Analysis and review of load balancing in grid computing using artificial bee colony. International Journal of Computer Applications, 71(20), 2013.

V. Yanmin Zhu. A survey on grid scheduling systems. Department of Computer Science, Hong Kong University of science and Technology, page 32, 2003.

VI. Barry G Lawson and Evgenia Smirni. Multiple-queue backfilling scheduling with priorities and reservations for parallel systems. In Job Scheduling Strategies for Parallel Processing, pages 72–87. Springer, 2002.

VII. Dror G Feitelson, Larry Rudolph, and Uwe Schwiegelshohn. Parallel job scheduling—a status report. In Job Scheduling Strategies for Parallel Processing, pages 1–16. Springer, 2004.

VIII. Srividya Srinivasan, Rajkumar Kettimuthu, Vijay Subramani, and P Sadayappan. Characterization of backfilling strategies for parallel job scheduling. In Parallel Processing Workshops, 2002. Proceedings. International Conference on, pages 514–519. IEEE, 2002.

IX. Dan Tsafrir and Dror G Feitelson. The dynamics of backfilling: solving the mystery of why increased inaccuracy may help. In Workload Characterization, 2006 IEEE International Symposium on, pages 131–141. IEEE, 2006.

X. Ranieri Baraglia, Gabriele Capannini, Marco Pasquali, Diego Puppin, Laura Ricci, and Ariel D Techiouba. Backfilling strategies for scheduling streams of jobs on computational farms. In Making Grids Work, pages 103–115. Springer, 2008.

XI. Hamid Reza Moaddeli, Gh Dastghaibyfard, and Mohammad R Moosavi. Flexible advance reservation impact on backfilling scheduling strategies. In Grid and Cooperative Computing, 2008. GCC'08. Seventh International Conference on, pages 151–159. IEEE, 2008.

XII. Shengwei Yi, Zhichao Wang, Shilong Ma, Zhanbin Che, Feng Liang, and Yonggang Huang. Combinational backfilling for parallel job scheduling. In Education Technology and Computer (ICETC), 2010 2nd International Conference on, volume 2, pages V2–112. IEEE, 2010.

XIII. Krzysztof Kurowski, Ariel Oleksiak, Wojciech Piatek, and Jan Weglarz. Hierarchical scheduling strategies for parallel tasks and advance reservations in grids. Journal of Scheduling, 16(4):349– 368, 2013.

XIV. Kiejin Park, Changhoon Kang, and Sungsook Kim. Hybrid job scheduling mechanism using a backfill-based multi-queue strategy in distributed grid computing. Int J Comput Sci Netw Secur, 12(9):39– 48, 2012.

XV. Zeynab Moradpour Hafshejani, Seyedeh Leili Mirtaheri, Ehsan Mousavi Khaneghah, and Mohsen Sharifi. An efficient method for improving backfill job scheduling algorithm in cluster computing systems.

XVI. Yulai Yuan, Yongwei Wu, Weimin Zheng, and Keqin Li. Guarantee strict fairness and utilizeprediction better in parallel job scheduling. Parallel and Distributed Systems, IEEE Transactions on, 25(4):971– 981, 2014.

XVII. Qian Li, Weiguo Wu, Xiaxin Zhou, Zeyu Sun, and Jianhang Huang. R-firstfit: A reservation based first fit priority job scheduling strategy and its application for rendering. In Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on, pages 1078–1085. IEEE, 2014.

XVIII. Qian Li, Weiguo Wu, Zeyu Sun, Lei Wang, Jianhang

ISSN (online): 2250-141X
Vol. 6 Issue 2, April 2016

Huang, and Xiaxin Zhou. A novel hierarchal scheduling strategy for rendering system. In 2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI), pages 206–209. IEEE, 2015.

XIX. K Vivekanandan et al. A study on scheduling in grid environment. In International Journal on Computer Science and Engineering (IJCSE). Citeseer, 2011.

XX. Sandip Fakira Lokhande, Sachin D Chavhan, and SR Jadhao. Grid computing scheduling jobs based on priority using backfilling. IJEECSE, 2(2):68–72, 2015.

XXI. Mr.S.R.Jadhao Dr.S.Y.Amdani. Job scheduling using backfilling techniques in grid computing: a survey. In International Conference on Advances in Computing and Information Technology (ICACIT2014), pages 110–114. Elsevier, 2014.