Evaluation of backfilling strategies based on workload models

Aditi Sharma,Department of Information Technology, UIET, PU,

RajkumariBhatia ,Assistant Professor, Department of Information Technology,

Abstract: **Grid computing was initially proposed to harness huge amounts computational resources. But the expected potential could not be fully exploited because the Grid applications are from a wide range of users each having own special requirements. In order to serve the needs, scheduling requirements of grid application first need to be understood. In this paper we will compare the performance aggressive, and conservative backfilling strategies based on average weighted response time(AWRT) and average slowdown of the jobs as the performance metrics**.

Keywords: **Grid computing, backfilling, task scheduling, average slowdown, AWRT.**

Introduction

The term "grid" refers to systems and applications that integrate resources and services distributed across multiple control domains[1].Grid computing is the distributed form of parallel computing, and makes use of various computers present on the internet for solution of a given problem. Grid provides its users with the required information and services to realize the resource sharing and collaboration in this virtual environment. Grid [2] has grown beyond the initial context since Ian Forsterfirst proposed it. The applications in which grid system is used and characteristics of grid have been attracting the researchers from years. A grid is able to process the resource requests of users ranging from a few to millions simultaneously. Computational grids provide large-scale resource sharing, such as personal computers, clusters, MPP's, database, and onlineinstructions, which may be cross-domain, dynamic and heterogeneous [3] The resources in grid system are geographically distributed and heterogeneous in nature. However the resources must be coordinated to provide aggregated computing capabilities. In order to hide the heterogeneity of resources the grid systems must follow the standard protocols, services and interfaces. This helps the grid computing to adapt to the dynamic environment and extract the maximum output from the available resources.

.Scheduling one of the most important technologies in grid computing and has been proved to be NP-hard problem due to the characteristics of grid computing. Main goal

Proc. of the Second International Conference on Advances in Computing, Control and Communication (CCN)
Editor In Chief Dr. R. K. Singh.

ISBN: 978-981-07-2579-2

of scheduling algorithms is to minimize the total execution time of an application. Though various scheduling algorithms based on execution time, QOS parameters, cost constrained etc have been used to optimize the scheduling problem but it is still hard to select the best one. Due to the underlying assumptions which the researcher makes while scheduling it becomes difficult to say that whether those assumptions will work on the other environment, because of the heterogeneous nature of the grid.

The paper is organized as follows. We provide backgroundabout job scheduling strategies in Section 2. In Section 3,we compare conservative and aggressive backfilling strategies on Lublin99 and Das2 workload model. In Section 4, we provide conclusions.

2 Background

Scheduling of parallel jobs is usually viewed in terms of time and the number of processors. The most basic and widely used scheduling algorithm is First come first serve(FCFS) algorithm and it suffers from low system utilization .Backfilling works by identifying “holes” and moving forward smaller jobs that fit those holes. The order of job execution is handled differently by two types of backfilling: conservativeand aggressive (EASY)[12, 13]. In conservative backfill,every job is given a reservation when it enters the system.A smaller job is allowed to move forward in the queue as long asit does not delay any previously queued job. In aggressivebackfilling, only the job at the head of the queue that is the pivot has a reservation. It maintains only one backfill at a time. A smaller job can leap forward as long as it does not delay the job at the head of the queue during scheduling.

Workload models used : lublin99 and Das2

Das2 stands for Distributed ASCI Supercomputer 2 (DAS-2). It is a 200-node wide-area distributed system built out of five Myrinet-based Dual Pentium-III clusters.Lublin99 is a very detailed model for rigid jobs, that includes an arrival pattern with a daily cycle, runtimes that are correlated with the number of nodes, and a distinction between interactive and batch jobs. Both the models are generating the set of jobs which the user provides and then these models generate the gridlets and submit them. We will run conservative as well as aggressive backfill strategy on both the models and accordingly compare the performance according to the performance of user metric.

Some of the common metrics used to evaluate theperformance of scheduling schemes are the average weighted response time and average slowdown. Weuse these metrics for our studies.

3Evaluation of Backfilling strategies

The simulation studies were performed with the Grisim scheduler on workload logs generated by workload models lublin99 and Das2. From the collection of jobs generated from these workload models we will be evaluating the performance of the backfilling strategies.

We are using conservative and aggressive backfilling strategies which will be run on the scheduler.

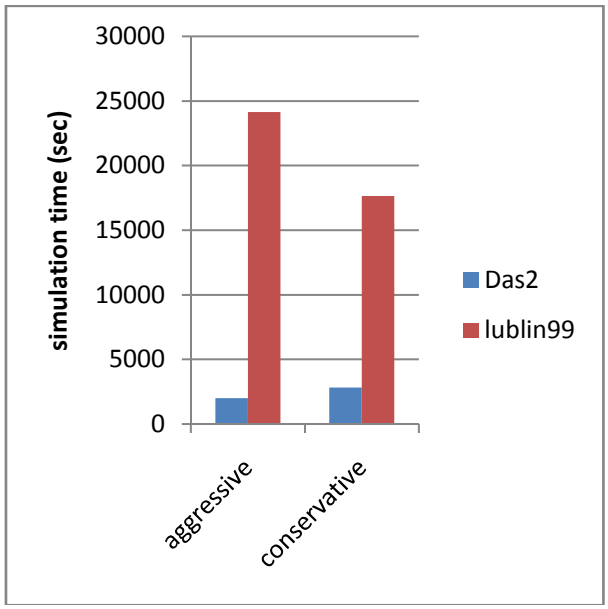

Table 1 AWRT for backfilling strategies

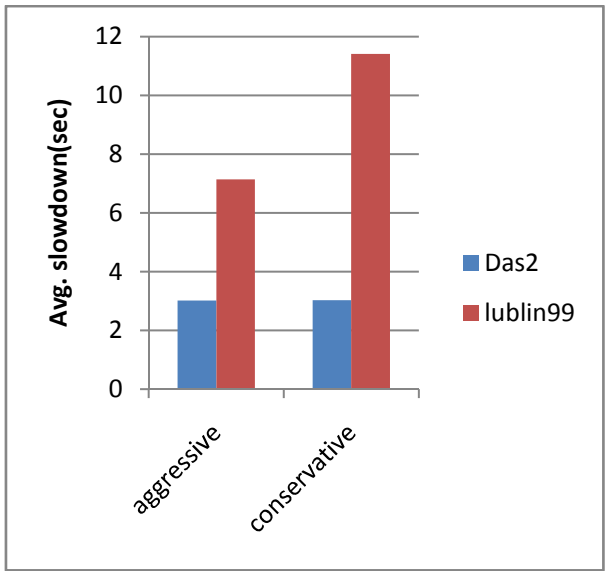

Table 2 Average slowdown

In the above table 1 and table 2 we have done comparison among the conservative and aggressive backfilling strategies on a set of 200 jobs run according to Das2 and lublin99 workload model. Both the backfilling strategies are running under the FCFS scheme.

The average slowdown of jobs is done for both the workload models over a set of 200 jobs. Slowdown for a job is defined as follows:

Slowdown = (Wait Time + Run Time) / Run Time

The average weighted response time is defined by:

AWRT= cputime*(fin-sub)/cputime

Where cputime= Gets the total execution time of a job from the latest gridresource

Fin=finish time of job

Sub=the arrival or the submission time of a job

We have plotted the graph for the average slowdown and AWRT for aggressive and conservative backfill strategies for whole set of jobs. On comparing the performance on the basis of average weighted response time we get that aggressive backfill outperforms conservative backfill in the Das2. However the average slowdown for the aggressive and conservative is almost similar in Das2 and no clear trend can be established. Conservative performs better and has a better response time in the lublin99 workload model.

The slowdown gives the idea to the user about the usage of the resources which are being heavily used and gives the metric to analyze that how the performance of the backfilling strategy changes whenever we are using a different workload model.Since the conservative backfill provides reservation in advance so the number of

backfill opportunities get reduced to some extent. When we compare the performance of conservative backfill for the slowdown metric, the Das2 model keeps it to a lower level as compared to lublin99 model. By doing so it improves the total runtime of the job set. The response time for lublin99 is better for the conservative backfill because the reservations have already been made in advance for every job. So the response time is greatly improved leading to reduced average weighted response time(AWRT) for conservative backfill.

Conclusion

We have studied differentscheduling techniques on a system consisting of two different workload models and local jobs compete for thesame resources. Two different scheduling approaches were considered and their performance was evaluated. Backfillingwas implemented in order to avoid fragmentation.Two sets of experiments were conducted using both simulation models.

In terms of average slowdown for Das2 no clear trend could be established for both backfilling strategies. While in case of lublin99 the average slowdown for the job set is greater for conservative backfill but the difference is not significantly large when we compare the values of the both the backfill strategies.

For future work we can improve scheduling algorithms by using the partitioning of single queue backfilling to fully extract the performance of the available resources and improve the total runtime of the system.

References

1) Uri Lublin and Dror G. Feitelson, The Workload on Parallel Supercomputers: Modeling the Characteristics of Rigid Jobs. J. Parallel & Distributed Comput.**63(11)**, pp. 1105-1122, Nov 2003.
2) WeifengSun,YudanZhu,Zhiyuan Su, Dong Jiao, Mingchu li, A priority-based task scheduling algorit+hm in grid, IEEE 2010.
3)] Zhenxia Yu , Fang Meng, Haining Chen,An Efficient List Scheduling Algorithm of Dependent Task in Grid, IEEE 2010
4) Wu YANG, YuanshuSUN,An improved shuffled frog leaping algorithm for grid task scheduling, IEEE 2011.
5) Ian Foster, what is the grid? A three point checklist, Grid Today, vol.1, pp6-12,2002.
6) HaolinFeng,Guanghua Song, Yao Zhag, Jun Xia, A deadline and budget constrained cost-optimization algorithm for scheduling dependent tasks in grid computing, pp 113-120, Springer-verlag Berlin Heidelberg 2004.
7) Fangpeng Dong and Selim G. Akl,Scheduling Algorithms for Grid Computing:State of the Art and Open Problems, Technical Report No. 2006-504
8) Sofia K. Dimitriadou, Helen D. Karatza, Job Scheduling in a Distributed System Using

Backfilling with Inaccurate Runtime Computations, IEEE 2010.

9) T. Kokilavani, Dr. D.I. George Amalarethinam, Applying non-traditional optimization techniques to task scheduling in grid computing-An overview,IJRRCS,Dec2010.

10) Wang Meihong, ZengWenhua, A comparison of four popular heuristics for taskscheduling problem in computational grid,IEEE 2010.

11) YaoshenLuan,JianlunSheng, Shuffled frog leaping algorithm based on particle swarm optimization. Computer and modernization 2009, (11):39-42

12) A. W. Mu'alem and D. G. Feitelson. Utilization, predictability,workloads, and user runtime estimates inscheduling the ibm sp2 with backfilling. In *IEEE Transactionson Parallel and Distributed Computing*, volume 12,pages 529–543, 2001.

13) J. Skovira, W. Chan, H. Zhou, and D. Lifka. The easy - loadleveler API project. In *JSSPP*, pages 41–47, 1996.

14) SrividyaSrinivasanRajkumar Kettimuthu Vijay Subramani P. Sadayappan.Characterization of Backfilling Strategies for Parallel Job Scheduling. IEEE 2002,pages 514.

15) RajkumarBuyya, "Economic-based Distributed Resource Management and Scheduling for Grid Computing", PhD Thesis, Monash University, Melbourne, Australia, April 12, 2002

16) Hui Li, David Groep, and LexWolters, "Workload Characteristics of a Multi-cluster Supercomputer"Lecture Notes in Computer Science, 2005, Volume 3277/2005, 176-193.

17) Ahuva W. Mu'alem and Dror G. Feitelson, Utilization, Predictability,Workloads, and User Runtime Estimates in Scheduling the IBM SP2 with Backfilling. IEEE Transactions on Parallel and Distributed Systems, 12:(6), pp. 529-543, 2001.