



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY
CLOUD AND DISTRIBUTED

Volume 13 Issue 1 Version 1.0 Year 2013

Type: Double Blind Peer Reviewed International Research Journal

Publisher: Global Journals Inc. (USA)

Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Performance Evaluation of Dynamic Scheduling for Grid Systems

By Radha.B & Dr.V.Sumathy

Sri Ramakrishna Engineering College, Coimbatore

Abstract - Schedulers are applications responsible for job management including resource allocation for a specific job, splitting them to ensure parallel task execution, data management, event correlation, and service-level management capabilities. When Grids allotted a number of jobs, such applications have to consider the overhead time, cost regarding to and from Grid resources, job transmission and at job processing, Grid resources for allocation of the jobs. In this paper, it is proposed to investigate the performance of dynamic scheduling algorithm of schedulers for executing different number of tasks is evaluated.

Keywords : *grid computing, resource management, grid schedulers, dynamic scheduling.*

GJCST-B Classification : *C.2.5*



Strictly as per the compliance and regulations of:



Performance Evaluation of Dynamic Scheduling for Grid Systems

Radha.B ^α & Dr.V.Sumathy^σ

Abstract - Schedulers are applications responsible for job management including resource allocation for a specific job, splitting them to ensure parallel task execution, data management, event correlation, and service-level management capabilities. When Grids allotted a number of jobs, such applications have to consider the overhead time, cost regarding to and from Grid resources, job transmission and at job processing, Grid resources for allocation of the jobs. In this paper, it is proposed to investigate the performance of dynamic scheduling algorithm of schedulers for executing different number of tasks is evaluated.

Keywords : grid computing, resource management, grid schedulers, dynamic scheduling.

1. INTRODUCTION

Grid portals ensure access to Grid resources the same as web portals. Grid portals provision of capabilities include grid computing authentication of resources, remote resource access, scheduling capabilities, and status information monitoring [1]. Such portals reduce task management complexity through customized/personalized user's graphical interfaces which alleviates users need for additional domain knowledge and not on specific grid resource management details [2].

Integrated solutions combine current advanced middleware and application functionalities, to ensure coherence and high performance results on a Grid Computing environment. Integrated Grid Computing solutions should include updated features to aid complex grids utilization like coordinated/optimized resource sharing, enhanced security management, cost optimizations, and other yet to be explored areas. Such integrated solutions sustain high values and provide significant cost reduction in commercial and non-commercial worlds. Various flexibility levels using infrastructures provided by application and middleware frameworks [3] can be achieved by Grid applications.

Employment of Grid technologies leads to better exploitation of a company's IT resources. Administrative overheads can also be lowered. As resources do not belong to various providers they are

not part administrative domains. This scenario has a centralized scheduling architecture; i.e. a central broker, forming a single access point to the entire infrastructure, managing resource manager interfaces interacting directly with local resource managers. All users submit jobs to this centralized entity.

Different computing sites, like scientific research labs collaborate for research, a scenario which is represented by high performing computing grids, where compute- and/or data-intensive applications are used on participating HPC computing resources which are generally parallel computers/cluster systems. Resources form part of several administrative domains in such cases, with own policies and rules. Jobs are submitted by users to the institute broker or VO level with brokers splitting a scheduling problem into many sub issues. Otherwise the same is forwarded to many brokers in the same VO.

Global Grids include various resources types from a single desktop machine to large-scale HPC machines all linked through a global Grid network where every peer-to-peer broker can accept jobs yet to be scheduled [4].

A job is a computational activity in a grid and consists of many tasks needing varied processing capabilities with different resource requirements. Tasks, jobs and applications are scheduled, allocated and processed as they are basic computational entities and resources. Resources have self-characteristics like CPU characteristics, memory and software. Many parameters like processing speed and workload which change over time are associated with a resource. Resources can also belong to different administrative domains due to different usage and access [5] policies.

Software components form part of Grid schedulers which take of computing/mapping tasks to Grid resources under multiple criteria/Grid environment configurations. There are varying levels in a Grid scheduler as stated in Grid computing literature. They include super-schedulers, meta-schedulers, local/cluster schedulers and enterprise schedulers. As a main component of any Grid system, The Grid scheduler interacts with the grid system's other components of the Grid system as it is the system's main component. Other grid system components include Grid information system, local resource management systems and network management systems. All such schedulers must coexist in Grid environments, even though they

Author α : B.Sc Degree in Chemistry in 2001, MCA Department of Computer Applications, Sri Ramakrishna Engineering College, Coimbatore. E-mail : radhakbr10@gmail.com

Author σ : B.E Degree in Electronics and Communication Engineering from Government College of Technology (GCT), Coimbatore, in the year 1988. Department of ECE, GCT, Coimbatore. E-mail : sumi_gct2001@yahoo.co.in

might have conflicting goals; this implies that interaction and coordination between various schedulers is a must if tasks are to be executed.

Grid scheduler follows many steps to perform scheduling which are categorized into 5 blocks (1) Preparation/information gathering on grid submitted tasks; (2) Resource selection; (3) Planning tasks computation to selected resources; (4) Planning based task (job/application) allocation (task mapping to selected resources); and (5) Task completion monitoring [6, 7].

Schedulers are responsible for jobs management of jobs including resources allocation for specific jobs, splitting jobs to ensure parallel task execution, data management, event correlation, and service-level management capabilities [7]. Schedulers are hierarchically structured with meta-schedulers, the root and other lower level schedulers, simultaneously providing specific scheduling capabilities which then morph into leaves. Schedulers could be built with local scheduler implementation approach for a particular job execution, or another meta-scheduler/cluster scheduler for parallel executions.

Grid service provider's performance is proportional to collective workload undertaken by many processors scattered globally on participating grid sites. It is challenging to predict time required for workload completion [8, 9]. When grids are presented many jobs, applications take overall processing including high overhead time and cost regarding to and from Grid resources, job transmission and job processing at Grid resources. This paper proposes to investigate dynamic scheduling algorithm performance to execute different tasks. The rest of the paper is organized as follows: Section II reviews related works in the literature, Section III describes the experimental setup; section IV discusses results and Section V conclude the paper.

II. RELATED WORKS

Typical scheduling structures that occur in computational grids were investigated by Volker Hamscher et al., [10]. Scheduling algorithms and selection strategies are presented and classified to these structures. Discrete-event simulation was performed to estimate features regarding various job combinations and Machine Models. The obtained results are discussed quantitatively and qualitatively. Simulation proves the importance of Backfill, for hierarchical scheduling. Unexpected results were obtained as FCFS confirms better performance than Backfill under a central job pool.

A complete investigation of a wide range of dynamic workflow scheduling policies performance in multi-cluster grids was presented by Sonmez et al [11]. Based on dynamic information amount used in the scheduling process, taxonomy of grid work flow

scheduling policies was presented with the taxonomy of seven such policies being mapped across the information use's full spectrum. Then scheduling policies performance was analyzed through simulations/ experiments in a real multi-cluster grid. The results revealed no single grid work flow scheduling policy had a good performance across investigated scenarios.

A novel grid-scheduling heuristic adaptively/ dynamically scheduling task without incoming tasks workload prior information was proposed by Bansal et al., [12]. The grid system is modelled in the form of a state-transition diagram, using a prioritized round-robin algorithm. This undertook task replication to optimally schedule tasks through the use of prediction information on individual nodes processor utilization. Simulations which compared the proposed approach with round robin heuristic revealed the latter to be better than the former in task scheduling.

The scheduling length minimization problem of a batch of jobs having varied arrival times was addressed by Barbosa et al [13]. Direct Acyclic Graph (DAG) of parallel tasks describes a job. A dynamic scheduling method is proposed by this paper which adapts the schedule on submission of new jobs and which change processors assigned during its execution. The scheduling method is split into scheduling Strategy and scheduling algorithm. An adaptation of Heterogeneous Earliest-Finish-Time (HEFT) algorithm, called P-HEFT is proposed to handle heterogeneous clusters parallel tasks with efficiently without a makespan compromise. When this algorithm was compared with another DAG scheduler through several machine configuration simulations and job types it revealed that P-HEFT ensures a shorter make span for single DAG. But scores are worse for multiple DAGs. In the end, results of dynamic scheduling of a jobs batch using the proposed scheduler showed significant improvements in heavily loaded machines when compared with an alternative resource reservation approach.

The problem of workflow applications scheduling in a grid environment was suggested by Mika et al [14]. Computational resources and network resources were the two divisions of grid resources. Computational workflow tasks and transmission tasks were distinguished. The problem was further divided into two sub problems: (i) how to find a distributed grid resources feasible resource for workflow task allocation so that all tasks resource demands (both computational, and transmission) were satisfied, and (ii) how to schedule local grid schedulers managed local resources computational tasks. The aim is to lessen total workflow (makespan) completion time. Computational experiments justified resource allocation stage importance and to examine local scheduling policy influence. Experiments revealed that even minor resource allocation improvement compared to random

allocation lead to greatly improved schedules. SJF algorithm produced slightly improved results compared to LJF algorithm as regards local scheduling policies. But the second experiment's main aim was to prove local scheduling policy influence with regard to the resource allocation stage.

a) Methodology

Processes are executed by the CPU scheduler when scheduled. When many ready processes are in the ready queue the scheduling algorithm decides the execution order of execution of these processes. Different CPU scheduling algorithms are First Come First Serve (FCFS), Shortest Job First (SJF) and Priority scheduling [15], all of which are by nature not pre emptive and hence unsuitable for time sharing systems. Shortest Remaining Time First (SRTF) and Round Robin (RR) are pre emptive in nature.

The Dynamic Level Scheduling (DLS) algorithm [16] uses the dynamic level (DL) attribute, the difference between a node's static level and its earliest execution

start time. At every scheduling step, a pair of node processor providing the highest DL value is selected; a process which is similar to that used by Earliest Time First (ETF) algorithm. But, there is a subtle difference between ETF and DLS: ETF algorithm schedules the node with minimum early execution start time with the static level being used to break ties. But the DLS algorithm in contrast, schedules nodes in their static levels descending order when the process first starts, but schedules nodes in EEST ascending order of EEST when the process nears the end .

b) Experimental Setup and Results

Simulations were carried out in Sim grid framework. The simulations were conducted using 8 clusters of resources. The resources are located at different locations connected using switches. The resources are scheduled using Random and Dynamic scheduling algorithm. The number of jobs of uniform size is varied from 1 to 1000. The DLS algorithm is as follows:

1. Compute the *SL* (static level) of each node in the graph
2. Initially, the ready nodes list includes only the entrynode
3. **while** the ready list is not empty **do**
 - compute the earliest execution start time for every ready node on each processor
 - compute the *DL* of every node-processor pair by subtracting the earliest execution start time from the node's static level (*SL*)
 - select the node-processor pair that gives the largest *DL*
 - schedule the node to the corresponding selected processor
 - add the newly ready nodes to the ready list

Table 1 : Grid Parameters

Number of node clusters	8
Number of jobs used in the simulation	200,400,600,800,100 jobs of uniform size.
Job workload	Uniform size
Job failure probability- ρ	0.15
Scheduling schemes used	Random, Dynamic

Table 2 : Tabulates the parameters used in the simulation

Table 2 tabulates part of the simulation results of the time taken to execute the varying number of tasks for Random and Dynamic scheduling. Figure 1 shows the same.

Table 3 : Simulation Results for time taken to execute varying number of tasks

No of tasks	Dynamic system	Random scheduling
1	0.194475	1.78215
50	17.1837	18.4649
100	33.4314	37.0473
150	51.7019	56.4195
200	68.2239	75.3997
250	87.6894	95.5514
300	101.95	114.922
350	131.146	139.513
400	141.48	151.381
450	166.75	189.209
500	185.042	188.58
550	201.204	217.56

600	217.87	227.713
650	235.683	247.082
700	255.054	266.066
750	269.141	284.296
800	290.652	301.764
850	317.501	320.741
900	314.168	339.721
950	337.405	359.483
1000	358.63	378.852

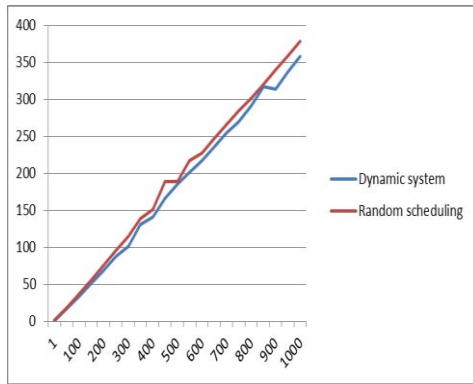


Figure 2 : Time Taken to execute varying Number of Tasks

It is observed from Figure 1, the time required to carry out the scheduled tasks is comparatively lower for the dynamic scheduling.

III. CONCLUSION

Software components make up Grid schedulers to compute task mapping to Grid resources through multiple criteria and Grid environment configurations. Scheduling's goals include achieving high performance computing and high throughput. The former is through execution time reduction for each job. It is usually utilized for parallel processing. In this paper, the performance of dynamic scheduling algorithm of schedulers for executing different number of tasks is evaluated. Simulation results demonstrate that the dynamic scheduling does improve the performance of the grid.

REFERENCES RÉFÉRENCES REFERENCIAS

1. Foster, I., Kesselman, C., Nick, J. and Tuecke, S. "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", Technical Report, Open Grid Service Infrastructure WG, Global Grid Forum, 2002.
2. Subramaniam, N. "Grid computing: An overview", Idea Group, 2003.
3. Joshy Joseph, Craig Fellenstein, "Grid Computing", IBM Press, ISBN-13: 978-0-13-145660-0, 2004.
4. N. Tonellotto, R. Yahyapour, Ph. Wieder, "A Proposal for a Generic Grid Scheduling Architecture", Core GRID Technical Report, Number TR-0015, January 11, 2006.
5. Xhafa, F., & Abraham, A. (2010). Computational models and heuristic methods for Grid scheduling problems. *Future Generation Computer Systems*, 26(4), 608-621.
6. Batista, D., & da Fonseca, N. (2010). A survey of self-adaptive grids. *Communications Magazine*, IEEE, 48(7), 94-100.
7. LizheWang, Gregor von Laszewski and Marcel Kunze "Grid Virtualization Engine: Design, Implementation, and Evaluation" *IEEE SYSTEMS JOURNAL*, VOL. 3, NO. 4, DECEMBER 2009.
8. J. Schopf and F. Berman, "Performance Prediction in Production Environments," *Proc. 12th Int'l Parallel Processing Symp.*, pp. 647-653, Apr. 1998.
9. Auver Grid Workload Report, http://gwa.ewi.tudelft.nl/pmwiki/reports/gwa-t-4/trace_analysis_report.html, 2009.
10. Hamscher, V., Schwiegelshohn, U., Streit, A. and Yahyapour, R., Evaluation of Job-Scheduling Strategies for Grid Computing. in *7th International Conference of High Performance Computing*, (Bangalore, India, 2000)
11. Sonmez, O., Yigitbasi, N., Abrishami, S., Iosup, A., & Epema, D. (2010, June). Performance analysis of dynamic workflow scheduling in multicluster grids. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing* (pp. 49-60). ACM.
12. Bansal, S., Kothari, B., & Hota, C. (2011). Dynamic Task-Scheduling in Grid Computing using Prioritized Round Robin Algorithm. *International Journal of Computer Science*, 8.
13. Barbosa, J. G., & Moreira, B. (2011). Dynamic scheduling of a batch of parallel task jobs on heterogeneous clusters. *Parallel Computing*, 37(8), 428-438.
14. Mika, M., Piatek, W., Waligóra, G., & Weglarz, J. (2011). Computational Experiments for Scheduling Workflow Applications in Grid Environment. *Computational Methods in Science and Technology*, 17(1-2), 53-62.
15. Pinedo, M. L. (2012). *Scheduling: theory, algorithms, and systems*. Springer.
16. Sih G., Lee E.: A Compile-Time Scheduling Heuristic for Interconnection-Constrained Heterogeneous Processor Architectures. *IEEE Trans. In Parallel and Distributed Systems*, 1993, Vol. 4, No. 2, p. 75-87.