International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 3– No.8, August 2012 – www.ijais.org

A Comparative Study on Grid Scheduling Algorithms

J.Shanthini
Research Scholar & Assistant Professor
INFO Institute of Engineering
Tamilnadu,India

K.R.Shankarkumar,
Phd, Professor
Sri Ramakrishna Engineering College
Tamilnadu, India

ABSTRACT
The grid is known to be a next generation computing technology; enables coordinated sharing of computing resources, which are geographically distributed. The scheduling is known to be an NP-hard problem for the last decades. Here we have compared different grid scheduling algorithms like ATC, WMDD, WMS, WMSPT, with the objective to minimize the total tardiness of the jobs. The simulation results show that the WMSPT, WMDD and ATC algorithm yields better perform tardiness minimization and WMSPT gives good minimization and lower wait time.

General Terms
Distributed computing, Grid computing, Scheduling algorithms

Keywords
Grid scheduler, Tardiness, dispatching rules, Weight, Due date.

1. INTRODUCTION
The Grid Computing has paved a new way for economic computing, in which wide range of distributed resources shared among the scientific community. The Grid paradigm aggregates high performance computing and high throughput computing as well. Grid distributes the jobs across the administrative domains. Grids are complex multivariate environments, which are made up of numerous Grid entities that need to be managed. Ian Foster [1] Defines *Grid computing is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organization*. The grid environment seems to be a promising trend for three reasons [2]: (1) It is capable of use give resources in a cost effective way,(2) able to solve more complex and teious problems, (3) it combins varity of resources that are synergistically harnessed, all these reasons actually work together inout set to achive acommon goal.

In grid computing security, scheduling and data transfer are the key areas of research. The grid scheduling is a process of scheduling grid resources for a grid application over administrative domains. Assignments of jobs to the best suitable resources are most challenging jobs in grid computing, these are realized through well defined standards and interfaces. Grid scheduling is intrinsically more complicated than local resource scheduling because it should manipulate large-scale resources across management boundaries.

The grid scheduler is a software component in charge of computing and mapping of task to grid resources under multiple criteria and grid environment configurations. The Grid scheduler consist of major components such as Grid Information service, Globus Resource Allocation Manager, Monitoring and Discovery service, Local Resource Management service. All these components must work coordinately in order to achieve better scheduling order; these components were shown in fig1.

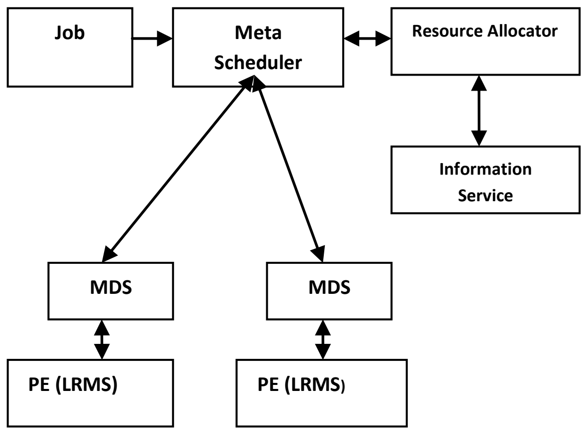

Fig.1. Grid Scheduling Architecture

The Grid Information services are like an Index service in the service oriented architecture, whenever the user contributes a service or resource to the grid community, it gets registered in the GIS service. In every instance the user submits the job to the Grid Meta scheduler; the Meta Scheduler invites this GIS via GRAM [3] to suggest the best suitable resources to execute the given job. The Gram component selects the suitable resources based on the user policy as well as provider policy. Now the job is sent to the local processing element for execution. The LRMS is now scheduling the job using local scheduling policies. The MDS monitors the grid job running in the local machine element and report to the GRAM component resides in the Meta scheduler.

The rest of the paper is organized as follows. In section 2, the current studies on grid task scheduling methods are briefly reviewed. The characteristics of ATC, WMDD, WMS, WMSPT, are described in detail in section 3. In section 4, the experimental results are shown. Finally, a conclusion is given in section 5.

2. RELATED WORK

Many Systematic investigations have been carried out in the field of resource and job scheduling. Maheswaran et al [4] have compared 11 heuristic algorithms, that are applied in the heterogeneous computing environment . Their study includes Opportunistic Load Balancing (OLB), K-Percent best(KPB), MinMin, MaxMin, MET, MCT and Genetic Algorithm. Fijumoto et al [5] proposed RR with TPCC and compared with MCT and MET. Kim et al [6] has combined MET and MCT proposed new algorithm named MECT. Wu [7] purposed segmented MinMin which concentrates more on load balancing sector. In [8] He et al, presented Qos guided Minmin, which gurantees Qos requirements of the user at the same time minimizes the makespan. Tseng [9] used machine scheduling algorithm ATCS and combined with MCT and achieved cost reduction. There are many researches , carried out toward minimizing the tardiness and makespan of the process in machine scheduling. In [10] N. Bahaji and M. E. Kuhl has presented dispatching rules, with comparison results with various other rules.

3. PROBLEM FORMULATION

In this section, we firstly describe the characteristic of algorithms that we compare and secondly we describe the adaptation of the algorithms in the Grid environment. We considered an independent task for this model, and we assume no communication between the processes were performed during the execution. We have also assumed to have non primitive jobs, in which job once submitted to the processor cannot be taken back until it is completed or aborted. We assumed there are n jobs, $N=\{1,..n\}$, each job $i \varepsilon N$, has processing time pi, and due date di priority or weight is models as wi.. We also assumed all jobs are arriving on time $t=0$. In order accommodate these dispatching rules with grid computing environment; we calculated the index values whenever the user jobs a new job to the grid environment.

3.1 Apparent Tardiness Cost (ATC)

It is a dynamic, priority based dispatching rule; job with highest priority index is scheduled at the next instance. This ATC rule combines well known heuristics WSPT (Weighted Shortest Processing Time) and MS (Minimum Slack) rules. In MS rule the slack of job j at time t, max $(dj-pj-t,\ 0)$, is computed and the job with the minimum slack is scheduled. Under the ATC rule jobs are scheduled one at a time; that is, every time the machine becomes free a ranking index is computed for each remaining job. The job with the highest ranking index is then selected to be processed next. This ranking index is a function of the time t at which the machine became free as well as of the pj , the wj and the dj of the remaining jobs. The index is defined as

$$Ij(t) = \frac{Wi}{Pj} * \exp\left(\frac{-\max(dj - pj - t, 0)}{K * pavg}\right) \qquad (1)$$

Here the *pavg* denotes the average processing time of remaining jobs and *k* is a look ahead parameter

3.2 Weighted Modified Due Date (WMDD)

The WMDD rule is a composite dispatching heuristic; the job with a tight due date with weight function is scheduled in the next sequence. The ranking index calculated here is a function of process time and due date with priority of the job. The job with lowest due date (i.e. Job with tight due date) is considered to be scheduled next

$$Ij(t) = \frac{\max(pj, dj - t)}{Wj} \qquad (2)$$

3.3 Weighted minimum Slack (WMS)

The WMS includes the weight function to the Minimum slack method, wherein the minimum slack process time on the job, due date and current time instance are taken into account.

$$Ij(t) = \frac{\max(dj - pj - t, 0)}{Wj} \qquad (3)$$

3.4 Weighted Minimum Slack Shortest Processing Time

The WMSPT is a composite dispatching rule proposed by Ibrahim H. Osman et al. [13]. This rule uses the components of ATC rule, however WMSPT is a parameter-free rule, uses WSPT and WMS rules. WMS rule is optimal when due dates are sufficiently spread out, Pinedo []. Under this WMSPT rule jobs are scheduled one at a time; each time when a processing element of any one of the cluster becomes free. The priority index in calculated for all the unscheduled jobs, and job with minimum index is selected for processing next. The ranking index is calculated as follows

$$Ij(t) = \frac{\max(dj - pj - t, pj)}{Wj} \qquad (4)$$

4. EXPERIMENTAL RESULTS

In the Grid environment, we have designed a centralized scheduling architecture [11]; where the centralized Meta scheduler computes the indexes for the jobs to be scheduled. All the unscheduled jobs *U* are considered for index calculation. We have formed a cluster of machines, and GIS produces each matching machine information. Whenever a user submits job to the grid portal the Meta scheduler starts a new scheduling instance, considering all unmapped jobs at once. For easy implementation we have assumed all jobs were arrived at time zero. We have used Alea 3 [12] for our simulation, with 4GB RAM and core i3 processor.

4.1 Algorithm Implementation

Most of the heuristics implemented here deals with jobs, weight, due date, and process time. We expect user to submit

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 3– No.8, August 2012 – www.ijais.org

their jobs weight and expected processing time and due dates. We have used metacentrum workload traces and Blue_12000 workload traces for running our simulation, around 5000 jobs and 1200 jobs were submitted for each traces and performances were recorded. The table 1 shows the result of first simulation and observed that ATC and WMSPT give better result than another. The total simulation was executed 30 times, and shown the performance

Table 1. The Performance Values Of Each Algorithm

Algorithm	Tardiness	Run time	Response Time	Average Run time
WMDD	181.32	1.14	68239.92	316579.5
WMS	3151.46	1.23	77441.81	315902.4
WMSPT	181.32	1.19	68239.92	316579.5
WRA	3942.32	1.02	65265.19	331291
ATC	181.32	1.25	68239.92	316579.5

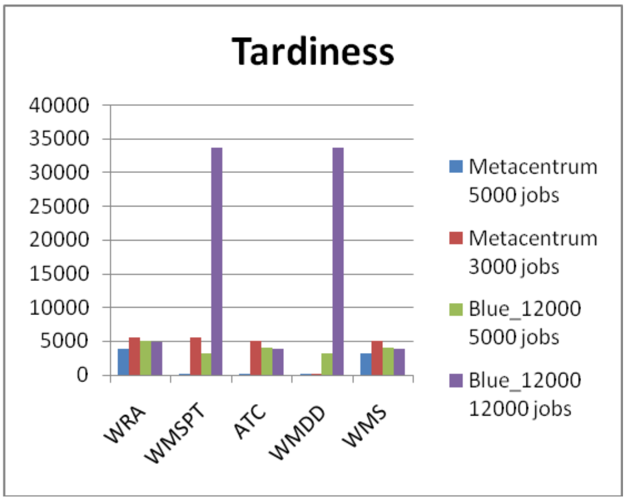

Fig. 2. Tardiness with various rules and workload traces

It was observed that algorithms WMDD, WMSPT and ATC equally performs and obtained same tardiness values for the simulation. The run time of the algorithms also recorded and each time the value changes, due to the machine failure model of the simulator, but then most likely values of runtime were recorded here. Algorithm WRA gives lowest run time, but does not give the same performance in tardiness minimization. The slowdown time surprisingly gives near results for ATC, WMDD and WMSPT.

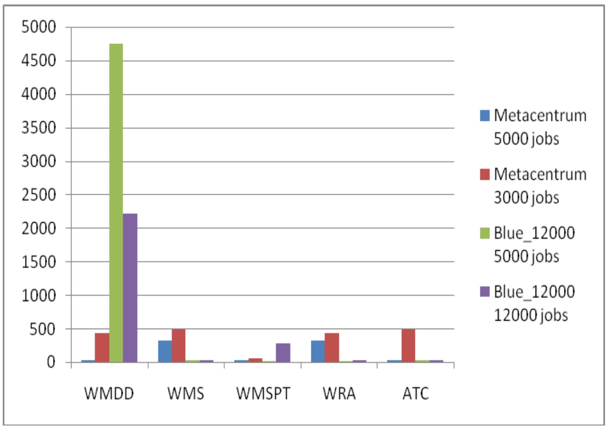

Fig. 4. Slowdown

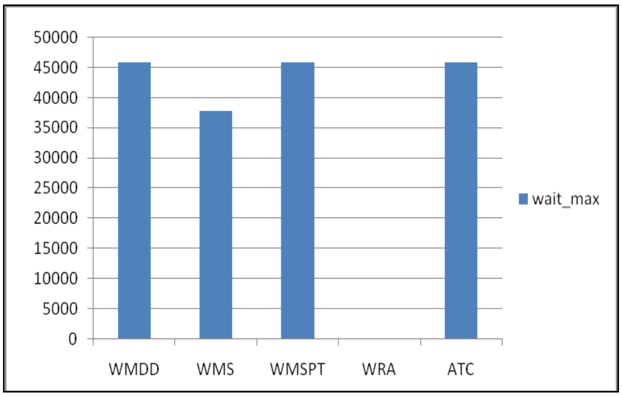

Fig. 3. Fairness (max. Wait time)

The fig 2. Shows the tardiness comparison between jobs executed by various algorithms as noted above. Fig 4 shows the slowdown scenrio of the cluster workload, the WMDD algorithm give huge slowdown, which is directly proposition to the performance of the algorithm.The WMSPT amd ATC algorithms works fine when fully loaded, their performaces are excellent, with 12000 jobs from BLUE workload traces. Fig. 3. Shows the fairness of the algorithms, Ibrahim H. Osman et al [13] rule gives smallest Average wait time while comparing to others

5. CONCLUSIONS

In this endeavor we analyzed and implemented some renowned algorithms used in job shop scheduling, in grid environment. The results shown here shows that ATC , WMDD and WMSPT performs in the same manner in our work. We understood that, load balancing and replica should be incorporated in this part of the work, since while cluster fails, the load become heavy and could not meet the deadline. The failure prediction model can also be incorporated in order make intelligent scheduling decisions. The WMSPT give a good reduction in average wait time of jobs, it also gives considerable tardiness minimization

6. AKNOWLEDGEMENTS

I would like to thank Dr. ShankarKumar for guiding me to come out with this paper. I record my gratitude to , Dr.V.Palanisamy, Pricipal and Dr.ChitraManohar, secretary INFO Intitute of Engineering. I also thank Dr. Umamaheswari for her valuable suggestions and guidance.

7. REFERENCES

[1] Foster and C. Kesselman (editors), The Grid: Blueprint for a Future Computing

[2] *Infrastructure*, Morgan Kaufmann Publishers, USA, 1999.

[3] http://searchdatacenter.techtarget.com/definition/grid-computing

[4] www.globus.com

[5] R. Braun, H. Siegel, N. Beck, L. Boloni, M. Maheswaran, A. Reuther, J. Robertson,M. Theys, B. Yao, D. Hensgen and R. Freund, *A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems*, in J. of Parallel and Distributed Computing, vol.61, No. 6, pp. 810-837, 2001

[6] Noriyuki Fujimoto, Kenichi Hagihara, *A comparison among grid scheduling algorithms for independent coarse-grained tasks*, Proceedings of the 2004 International Symposium on Applications an the Internet Workshops (SAINTW'04), Vol. 26–30, January 2004, pp. 674–680

[7] Hak Du Kim, Jin Suk Kim, An online scheduling algorithm for grid computing

[8] *systems*, GCC (2), LNCS 3033, Springer, Shanghai, China, 2004, pp. 34–39.

[9] M. Wu, W. Shu and H. Zhang, *Segmented Min-Min: A Static Mapping Algorithm for Meta-Tasks on Heterogeneous Computing Systems*, in Proc. of the 9th Heterogeneous Computing Workshop (HCW'00), pp. 375--385, Cancun, Mexico, May 2000.

[10] X. He, X. Sun and G. Laszewski, *A QoS Guided Min-Min Heuristic for Grid Task Scheduling*, in J. of Computer Science and Technology, Special Issue on Grid Computing, Vol.18, No.4,pp.442--451, July 2003.

[11] L.Y. Tseng, Y.H. Chin , S.C. Wang, The anatomy study of high performance task scheduling algorithm for Grid computing system, Computer Standards & Interfaces 31 (2009) 713–722

[12] N. Bahaji , M. E. Kuhl, A simulation study of new multi-objective composite dis patching rules, CONWIP, and push lot release in semiconductor fabrication, Journal of Production Research, 46:14, 3801 — 3824

[13] N. Tonellotto, R. Yahyapour and Ph. Wieder , *Towards a standards-based Grid Scheduling Architecture,* coreGRID Technical Report No.TR-00015, Sep.13.2005

[14] Dalibor Klusáček and Hana Rudová. *Alea 2 - Job Scheduling Simulator*. In proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools 2010), ICST, 2010.

[15] Ibrahim H. Osman, Hocine Belouadah,Krzysztof Fleszar and Maysaa saffar, Hybrid of weighted minimum slack and shortest Processing time dispatching rule for the the total weighted tardiness single machine scheduling problem with availability constraints., Multidisciplinary International Conference on scheduling: Theory and Applications (MISTA 2009)