International Journal of Power Control Signal Computation(IJPCSC)
Vol 7. No.1 2015 Pp. 41-46

available at : www.ijcns.com
ISSN: 0976-268X

Centralized and Decentralized Job Scheduling Algorithm for Scheduling Jobs on Computational Grids

Dr.G.K.Kamalam
Kongu Engineering College, India

Abstract- Scheduling of jobs to the distributed heterogeneous resources in grid is a complicated problem. In grid computing system, when all jobs are assigned to the same resource leads to the high work load for the resource and the computational time of the processed jobs is also high. Efficient job scheduling algorithm is essential to achieve high system throughput and for the effective resource utilization. This paper presents a grid architecture model as a collection of clusters. The proposed hybrid job scheduling algorithm which performs the intra cluster and inter cluster job scheduling. The hybrid job scheduling algorithm is developed by integrating Divisible Load Theory (DLT) and Least Cost Method (LCM). The proposed hybrid job scheduling algorithm will reduce the processing time, processing cost and improves the resource utilization. Experimental results obtained showed that the hybrid job scheduling algorithm is a promising algorithm for scheduling of jobs in heterogeneous grid environment.

Keywords: Centralized and Decentralized, Job Scheduling, Algorithm, Scheduling Jobs, Computational Grids

I. INTRODUCTION

The goal of a grid system is to solve computing problems through many geographically distributed heterogeneous resources belonging to different administrative domains [1,6,12,19].The coordination of multiple administrative domains leads to the heterogeneity in grid environment[10].

In dynamic grid environment, network connections may become unreliable; resources may join or leave the grid in an unpredictable way, the access policies of the resources could change over time. In grid computing environment, there exists more than one resource to compute to process a job in term of minimizing the processing time of the job [16].

Job scheduling in grid is an NP-hard problem [14, 20, 21, 22]. Scheduling at the grid level is a challenging problem because of the dynamics and heterogeneity of the system [11,13,15, 24].

Grid job scheduling is of two types: static and dynamic scheduling [1]. In static scheduling, there is no job failure and the resources are assumed to be available all the time. The jobs are assigned to the suitable resources before their execution begin. Once, execution begins, the job keeps running on the same resource without preemption. The two main important factors that decide the dynamics of the job scheduling is namely (i) dynamics of the job execution which refers to the situation of job failure or job execution could be preempted due to the arrival of high priority job.(ii) dynamics of resources, which refers to the situation where the resources from various administrative domains can enter or leave the grid at any time, some resources are completely dedicated to the grid, some provide service only when they are idle and so on.

The main objective of the scheduling algorithm is to reduce the make span. Optimizing job execution time is done by partitioning the jobs into sub jobs and efficiently scheduling sub jobs to different clusters in a grid environment[2,17,25].

Independent jobs are those which do not have any interprocess communication lead to the divisible of large jobs. The divisible load theory is one solution for scheduling of divisible sub jobs to the resources available in the grid [7,8,9].The divisible job is one that can be divided into sub jobs of equal size and it can be processed independently on the available heterogeneous resources[3,5].

In grid environment, the service provider, provide the resources in a cost effective manner. The scheduling of the jobs to the resources is done using LCM [3].

The aim of this research paper is mainly focused on improving the performance of a grid environment by the use of effective scheduling algorithm. The proposed algorithm, the efficient centralized and decentralized job scheduling algorithm for a grid environment modeled as a collection of clusters. Users are associated to various clusters. In a decentralized job scheduling algorithm, the jobs submitted to a cluster can be migrated to another cluster with minimum processing time. In a centralized job scheduling algorithm, the coordinator node of each cluster is responsible for scheduling of jobs and allocates the job to the worker node of the cluster where the user is associated with minimum processing time of the job. In this paper, the proposed algorithms are, the centralized hybrid job scheduling algorithm and decentralized hybrid job scheduling algorithm that apply Divisible Load Theory (DLT) and Least Cost Method (LCM) for scheduling the jobs to improve the makespan of the jobs.

The organization of rest of the paper is as follows: The section 2 presents the literature review. The section 3 focuses on the grid architecture model. The section 4 and 5, in turn, explains the various centralized and decentralized job scheduling algorithm. While section 6 compares the results of various centralized job scheduling algorithm and decentralized job scheduling algorithm and section 7 talks about conclusion and future work.

II. LITERATURE REVIEW

Divisible load theory provides a realistic approach to scheduling of computational jobs in parallel and distributed computing systems. DLT originated to cater the needs of recent applications that involve parallel and distributed computing. DLT is a powerful tool for modeling data-intensive computational problems [8, 26,27].

DLT is based on the divisibility property of the workload. It is classified into Arbitrarily Divisible Loads (ADL) and Modularly Divisible Loads (MDL). In ADL, the workloads can be arbitrarily partitioned into any number of load fractions. In MDL, workload can be subdivided into smaller modules based on some characteristics of the load of the application [7]. DLT assumes that computational loads can be partitioned arbitrarily among multiple processors and links.

Grid resource allocation is an important linear programming transportation problem. A grid resource allocation problem deals with the assignment of jobs to a number of resources available in the grid at minimum cost. The aim is to make optimum allocation of resources among the grid users to achieve minimum computational cost [18].

III. GRID ARCHITECTURE MODEL

By $C = \{C_1, C_2, \ldots, C_m\}$ denotes the set of clusters forming the grid. Each user U_k owns a cluster C_k. The set $U = \{U_1, U_2, \ldots, U_o\}$denotes the set of users. The set of all jobs produced by C_k of user U_k is denoted by J_i, where the job set is denoted by $J = \{J_1, J_2, \ldots, J_k\}$and the sub jobs of each job is denoted by the set $SJ = \{SJ_{i1}, SJ_{i2}, \ldots, SJ_{il}\}$. Each cluster has a coordinator node denoted as CN. Each cluster has a set of worker nodes denoted as $WN = \{WN_{i1}, WN_{i2}, \ldots, WN_{in}\}$. Each worker node has different processing powers.

In a decentralized job scheduling, the sub jobs SJ_{ik} of cluster C_i is assigned to the worker node of the originating cluster or it will be migrated to any clusters depending on the processing time of the job.

In a centralized job scheduling, the sub jobs SJ_{ik} of cluster C_i is allocated to the worker node of the same originating cluster with minimum processing time.

IV. CENTRALIZED JOB SCHEDULING ALGORITHMS (CJSA)

A. ***Centralized Local Job Scheduling Algorithm (CLJSA)*****:** The centralized local job scheduling algorithm allocates the job to the worker node of the cluster where the job has been originated. The selection of worker node is based on the minimum completion time of the job.

Step1: Job submission by the user to the CN of a cluster.

Step2: The scheduler allocates the job to the WN of the cluster where the user is associated with minimum completion time of the job.

Step3: If no more jobs to be scheduled, the processing time and the total cost of completing the job is calculated.

Step4: If jobs are still to be scheduled go to step 2.

Step5: END.

B. ***Centralized Divisible Job Scheduling Algorithm (CDJSA)*****:** By applying DLT, the jobs submitted by the user are divided into sub jobs of equal size to the maximum of five partitions. The scheduler then schedules the sub jobs to the WN of the cluster with minimum processing time where the user is associated.

Step1: Submission of job to the CN of a cluster.

Step2: Divide the job into sub jobs.

Step3: The scheduler then schedules the sub jobs to the WN of the cluster with minimum processing time.

Step4: Repeat step 3 until all sub jobs is scheduled.

Step5: Repeat step 2 to step 4 until job set J is empty.

Step6: Calculate the processing time and processing cost of the jobs.

Step7: END.

C. ***Centralized Hybrid Job Scheduling Algorithm (CHJSA)*****:** By applying DLT and LCM, the centralized hybrid job scheduling algorithm divides the job into sub jobs of equal partitions of maximum five partitions and allocates to the WN of the cluster with minimum processing time and minimum processing cost.

Step1: Submission of job to the CN of a cluster.

Step2: Divide the job into sub jobs and add it to the sub job set SJ.

Step3: The scheduler then schedules the sub jobs to the WN of the cluster with minimum processing time and processing cost.

Step4: Remove sub job from job set SJ.

Step4: Repeat step 3 until all sub jobs is scheduled.

Step5: Repeat step 2 to step 4 until job set J is empty.

Step6: Calculate the processing time and processing cost of the jobs.

Step7: END.

V. DECENTRALIZED JOB SCHEDULING ALGORITHMS (DJSA)

D. *Decentralized Job Scheduling Algorithm (DJSA)*: The decentralized job scheduling algorithm allocates the job to the worker node of any cluster. The selection of worker node is based on the minimum completion time of the job.

Step1: The completion time information of WN is updated periodically at GIC and CN of each cluster.

Step2: Job submission by the user to the CN of a cluster.

Step3: Upon receiving the completion time information from GIC the scheduler allocates the job to the WN of any cluster with minimum completion time of the job.

Step4: If no more jobs to be scheduled, the processing time and the total cost of completing the job is calculated.

Step5: If jobs are still to be scheduled go to step 3.

Step6: END.

E. *Decentralized Divisible Job Scheduling Algorithm (DDJSA)*: By applying DLT, the jobs submitted by the user are divided into sub jobs of equal size to the maximum of five partitions. The scheduler then schedules the sub jobs to the WN of any cluster with minimum processing time where the user is associated.

Step1: The completion time information of WN is updated periodically at GIC and CN of each cluster.

Step2: Submission of job to the CN of a cluster.

Step3: Divide the job into sub jobs.

Step4: The scheduler then schedules the sub jobs to the WN of any cluster with minimum processing time.

Step5:Repeat step 4 until all sub jobs is scheduled.

Step6: Repeat step 3 to step 5 until job set J is empty.

Step7:Calculate the processing time and processing cost of the jobs.

Step8: END.

F. *Decentralized Hybrid Job Scheduling Algorithm (DHJSA)*: By applying DLT and LCM, the centralized hybrid job scheduling algorithm divides the job into sub jobs of equal partitions of maximum five partitions and allocates to the WN of any cluster with minimum processing time and minimum processing cost.

Step1: The completion time information of WN is updated periodically at GIC and CN of each cluster.

Step2: Submission of job to the CN of a cluster.

Step3: Divide the job into sub jobs and add it to the sub job set SJ.

Step4: The scheduler then schedules the sub jobs to the WN of the cluster with minimum processing time and processing cost.

Step5: Remove sub job from job set SJ.

Step6: Repeat step 4 until all sub jobs is scheduled.

Step7: Repeat step 3 to step 6 until job set J is empty.

Step8: Calculate the processing time and processing cost of the jobs.

Step9: END.

VI. RESULTS AND DISCUSSION

In this paper, the proposed hybrid job scheduling algorithm is compared with local job scheduling algorithm and divisible job scheduling algorithm.

Simulation parameters taken from [4]. In this scenario, the grid system consists of five users and the parameters are listed in Table I.

Table I. Simulation Parameters

Parameters	Value
No. of Clusters	10
No. of worker nodes per cluster	10
Processing power of worker node	50 – 1000 MIPS
Job length	2,50,000-6,50,000 MI
Cost	1-5 G$ unit
No. of users	5
No. of Jobs	50 - 500

The performance of centralized job scheduling algorithms and decentralized job scheduling algorithms is based on the three metrics: Total processing time, Total cost and Number of jobs. The performance of the centralized job scheduling algorithms and decentralized job scheduling algorithms is compared by varying the number of submitted jobs.

The total processing time of completing the jobs using CJSA and DJSA algorithm is shown in Table1 and Table2. From Table1 and Table2 it is clear that the total processing time of completing the jobs of DJSA algorithm is minimum compared to that of the CJSA algorithm. The total processing cost of completing the jobs using CJSA and DJSA algorithm is shown in Table3 and Table4. From Table3 and Table4 it is clear that the total processing cost of completing the jobs of DJSA algorithm is minimum compared to that of the CJSA algorithm.

Graphical representation in Figure 1 shows that the DJSA provides a minimum makespan than the CJSA and also DHJSA provides minimum makespan than all the other CJSA and DJSA algorithms. Graphical representation in Figure 2 shows that the DJSA provides a minimum processing cost than the CJSA and also DHJSA provides minimum processing cost than all the other CJSA and DJSA algorithms.

Table 1: Total Processing Time of CJSA Algorithm

No. of Jobs	CLJSA	CDJSA	CHJSA
50	359	409	242
100	620	1113	445
150	824	1745	582
200	660	1114	513
250	1208	1983	821
300	1097	1471	698
350	1439	2166	912
400	2279	2874	1421
450	2353	4268	1636
500	2646	2851	1619

Table 2: Total Processing Time of DJSA Algorithm

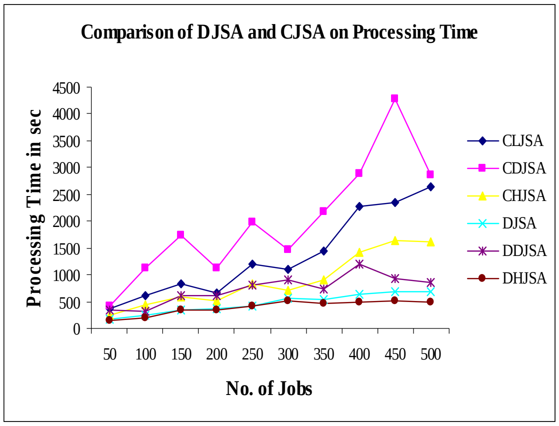
No. of Jobs	DJSA	DDJSA	DHJSA
50	169	342	151
100	243	329	192
150	348	622	336
200	365	608	333
250	410	801	408
300	569	907	506
350	541	736	456
400	630	1196	495
450	675	930	524
500	686	846	488

Table 3: Total Processing Cost of CJSA Algorithm

No. of Jobs	CLJSA	CDJSA	CHJSA
50	14556	9093	8976
100	33546	22767	20655
150	52122	40074	36549
200	67986	50601	45336
250	86439	55959	53952
300	120645	76794	72684
350	128382	79851	79341
400	153525	97428	94500
450	163512	121209	130502
500	177612	120953	122686

Table 4: Total Processing Cost of DJSA Algorithm

No. of Jobs	DJSA	DDJSA	DHJSA
50	13020	7530	7476
100	27969	16524	16053
150	48711	35619	35010
200	66927	44520	44334
250	81678	54822	53622
300	103149	70965	68622
350	115449	77304	76554
400	139380	92367	90885
450	158340	112746	110122
500	165780	116401	106839



. **Fig 1: Impact of CJSA and DJSA on Makespan**

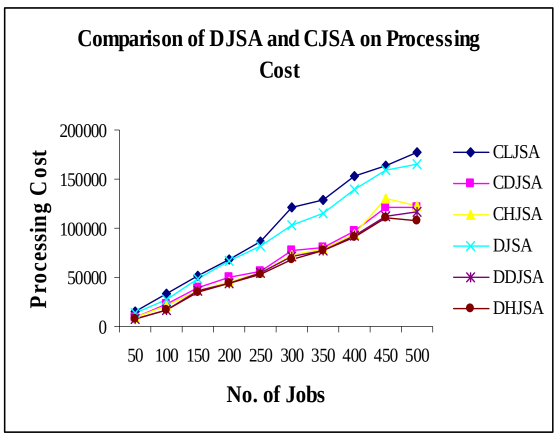


Fig 2: Impact of CJSA and DJSA on Total Cost

VII. CONCLUSION AND FUTURE WORK

In this paper, an efficient centralized job scheduling algorithm and decentralized job scheduling algorithm for allocating the jobs to the worker node of the originating cluster is proposed. The centralized hybrid job scheduling algorithm produces better results as compared to the centralized local job scheduling algorithm and centralized divisible job scheduling algorithm. The decentralized hybrid job scheduling algorithm produces better results as compared to the decentralized job scheduling algorithm and decentralized divisible job scheduling algorithm. The centralized hybrid job scheduling algorithm is a potential solution for scheduling of jobs in a centralized grid environment. The decentralized hybrid job scheduling algorithm is a potential solution for scheduling of jobs in a decentralized grid environment. Finally, intention is to use the proposed algorithm in an actual environment for practical evaluation.

VIII. REFERENCES

[1] Ku Ruhana ku-Mahamud, Husna Jamal Abdul Nasir,"Ant Colony Algorithm for Job Scheduling in Grid Computing", in fourth Asia International Conference on Mathematical/Analytical Modeling and Computer Simulation",2010,pp.40-45. .

[2] Valrmathi V.Korkhor, Jakub T.Moscicki, and Valeria V.Krzhizhanovskaya,"The User-level Scheduling of Divisible Load Parallel Applications with Resource Selection and Adaptive Workload Balancing on the Grid",IEEE Systems Journal,Vol.3.No.1, March 2009.pp121-130.

[3] Syed Nazir mehmood shah, Ahmad Kamil Bin Mahmood, and Alan Oxley, " Hybrid Resource Allocation Method for Grid Computing", in IEEE second International Conference on Computer Researh and Development, 2010, PP.121-130.

[4] P.K suri, and Monpret Singh, "An Efficient Decentralized Load Balancing Algorithm for Grid", in IEEE Second International Advance Computing Conference, 2010, pp.10-13.

[5] Wim Depoorter, Ruben Van Den Bossche, Kurt Vanmechelen, and Jan Broeckhove, " Evaluating the Divisible Load Assumption in the Context of Economic Grid Scheduling with Deadline Based QoS Guarantees", 9th IEEE/ACM International Symposium on Cluster Computing and the Grid,2009, pp. 452 – 459.

[6] Sandeep Singh Rawat, and Dr.Lakshmi Rajamani, "Experiments with CPU Scheduling Algorithm on a Computational Grid", in IEEE

Iternational Advance Computing Conference, March 2009, pp.71-75.

[7] V.Bharadwaj, D.Ghose, and T.G.Robertazzi, "Divisible Laoad Theory: A New Paradigm for Load Scheduling in Distributed Systems", Cluster Computing on Divisible Load Scheduling, 2003,pp. 7 – 18.

[8] T.G.Robertazzi, "Ten Reasons to use Divisible Load Theory", Computer, Vol.36, No.5,pp. 63 – 68.

[9] D.Yu, T.G.Robertazzi 2003, "Divisible Load Scheduling for Grid Computing ", in Proceedings of the International Conference on Parallel and Distributed Computing Systems.

[10] N.Malarvizhi, and Dr. V.Rhymend Uthariaraj, " A Minimum Time to Release Job Scheduling Algorithm in Computational Grid Environment", in Fifth International Joint Conference on INC, IMS and IDC, 2009,pp.13-18.

[11] Daniel Vladusic, Ales Cernivec, and Bossjan Slivnik, "Improving Job Scheduling in Grid Environments with use of Simple Machine Learning Methods", in Sixth International Conference on Information Technology: New Generations, 2009, pp.177-182.

[12] Mustafizur Rahman, Rajiv Rajan, and Rajkumar Buyya,"Dependable Workflow Scheduling in Global Grids", in 10th IEEE/ACM International Conference on Grid Computing, 2009, pp.153-162.

[13] Weel Abdullal, Omar Al Jadaan, Ahmad Jabas, and S.Ramachandran, "Genetic Algorithm for Grid Scheduling using Best Rank Power", in 2009 World Congress on Nature and Biologically Inspired Commputing,pp.181-186.

[14] Leli Mohammad Khanli, Maryam Far, and Amir Masoud Rahmani, "RFOH: A New Fault Tolerant Job Scheduler in Grid Computing ",2010 2nd International Conference on Computer Engineering and Applications, pp.422-425.

[15] Ruay-Shiung Chang, Chih-Yuan Lin, and Chun_Fu lin,"Scheduling Jobs in Grid Adaptively",IEEE International Symposium on Paralel and Distributed Processing with Applications, 2009, pp.19-25.

[16] Kuo-Chan Huang,"On Effects of Resource Fragmentation on Job Scheduling Performance in Computing Grids", 10th IEEE International Symposium on Pervasive Systems, Algorithms, and Network,2009, pp.701-705 .

[17] KeginLi, "Job Scheduling for Grid Computing on Meta Computers", 19th IEEE Internatinal Conference on Parallel and Distributed Processing Symposium",2005.

[18] Syed Nazir mehmood shah, Ahmad Kamil Bin Mahmood, and Alan Oxley, " Modified Leasr Cost Method for Grid Resource Allocation", in IEEE International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2010, PP.218-225.

[19] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2010 a. A New Heuristic Approach:Min-Mean Algorithm For Scheduling Meta-Tasks On Heterogeneous Computing Systems. IJCSNS International Journal of Computer Science and Network Security, Vol.10 No.1, pp. 24-31.

[20] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2010 b. An Improved Min-Mean Heuristic Scheduling Algorithm for Mapping Independent Tasks on Heterogeneous Computing Environment. International Journal of Computational Cognition, Vol. 8, N0. 4, pp. 85-91.

[21] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2011. An Efficient Hybrid Job Scheduling for Computational Grids. International Journal of Computer Applications.

[22] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2012a. New Enhanced Heuristic Min-Mean Scheduling Algorithm for Scheduling Meta-Tasks on Heterogeneous Grid Environment. European Journal of Scientific Research, Vol.70 No.3, pp. 423-430.

[23] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2012b. Novel Adaptive Job Scheduling algorithm on Heterogeneous Grid Resources. American Journal of applied Sciences, pp. 1294-1299.

[24] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2014a. Resource Fitness Task Scheduling Algorithm for Scheduling Tasks on Heterogeneous Grid Environment. Australian Journal of Basic and Applied Sciences, Vol.8 No.18, pp. 128-135.

[25] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2014b. Best Cluster Decentralized Job Scheduling Algorithm for Scheduling Tasks on Heterogeneous Grid Environment. Australian Journal of Basic and Applied Sciences, Vol.8 No.18, pp. 171-177.

[26] Kamalam, G.K. and V.M. Bhaskaran, 2011a. An effective approach to job scheduling in decentralized grid environment. Int. J. Comput. Appli., 24: 26-30. DOI: 10.5120/2914-3838

[27] Kamalam, G.K. and V.M. Bhaskaran 2011b. An efficient hybrid job scheduling algorithm For computational grids. Int. J. Comput Appli.