MidSFN Local Scheduling Algorithm for Heterogeneous Grid Environment

Dr.G.Sumathi, R.Santhosh Kumar, and S.Sathyanarayanan
1Department of Information Technology, Sri Venkateshwara College of Engineering Sriperumbudur,TamilNadu,India

2Department of Information Technology, Sri Venkateshwara College of Engineering Sriperumbudur,TamilNadu,India

3Department of Information Technology, Sri Venkateshwara College of Engineering Sriperumbudur,TamilNadu,India

Abstract

A grid is a heterogeneous computing and communication system that allows a group of nodes to compass a task. The process of assigning the jobs or subtasks to the nodes present in the heterogeneous grid is known as scheduling. The type of scheduling in which the subtasks are assigned to the nodes is known as Local Scheduling.A significant research problem is how to assign resources to the subtasks and order the execution of the subtasks that are matched to maximize performance criterion of a local grid system. This procedure of scheduling is called mapping or resource allocation. MidSFN adverts to Medium Subtask Fastest Node algorithm which classifies the subtasks into three tier categories, High, Middle and Low based on their priority. In MidSFN algorithm priority is assigned based on the new parameters Computational Complexity and Processing Power. The value for processing power is assigned based on the Performance Factor. The value of the Performance Factor is the product of the number of operations per cycle per processor and the number of instructions processed per second. In MidSFN algorithm the subtask of medium computational complexity and resources exhibiting medium processing power are assigned with a high priority. The subtasks are then mapped to respective processors based on the assigned priority for execution. Compared to other local scheduling algorithms, MidSFN algorithm shows efficient load balancing and better computation with effective usage of resources. The effectiveness of this algorithm is evaluated through simulation results.

***Keywords*:** *Computational Grid, Local scheduling, Computational Complexity, Directed Acyclic Graph (DAG)*

1. Introduction

Computational Grid [1] is one that performs any form of calculation with a greater efficiency. It provides high end resources for executing parallel tasks. The resources available in the grid could be a single processor, a symmetric multiprocessor cluster, a distributed memory multiprocessor system, or a massively parallel supercomputer. The process of assigning jobs or subtasks to the resources of a grid is known as scheduling, and this is done by a grid scheduler .There are two classifications among the grid schedulers Global Scheduler and Local Scheduler. A local scheduler is different from global scheduler. A local scheduler is in charge of allocation of nodes in a resource, assigning of subtasks and subtask execution management. A proper scheduling and efficient load balancing can lead to improved overall system performance in heterogeneous grid environment with its multiple resources, also leading to a lower turn-around time for individual subtask. FirstComeFirstServe (FCFS) algorithm [1] neither considers any of the subtask parameters nor the resource parameters. The Shortest Subtask Fastest Node(SSFN) and Longest Subtask Fastest Node (LSFN) algorithms consider [2] computational complexity of subtask for scheduling and ignore the priority of a subtask.

Our Performance Factor based local scheduling algorithm assigns a priority to the subtask based on the parameters Computational Complexity and Performance Factor.The value of the Performance Factor is assigned based on the number of operations per cycle per processor and the number of instructions processed per second. In this a subtask which requires high processing power and which exhibits high computational complexity is given a high priority. A subtask, which exhibits high computational complexity and requires low processing power, is given a low priority. A subtask, which exhibits a medium computational complexity and requires medium processing power, is given a medium priority. The fastest free node available in the resource is allocated to the subtask which has high priority. The proposed MidSFN algorithm is a revisedversion of our Performance Factor based local scheduling algorithm. The MidSFN algorithm is based on the parameter ‘priority’ for the assignment of subtask to the resources. The subtask is prioritized on the basis of

The first author thanks the Defense Research & Development Organization (DRDO), Government of India, New Delhi, for its financial assistance through extramural research grant.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

computational complexity and processing power. The subtasks with medium computational complexity and requires medium processing power are given a high priority. The rest of the paper is organized as follows.

The grid framework is presented in Section 2, the proposed scheduling algorithm is discussed in Section 3, the performance study is carried out and results are discussed in Section 4 and finally, some concluding remarks in Section 5.

2. GENERAL FRAMEWORK OF A GRID

Fig.1 shows the framework of the grid. The grid is mainly formed by the following components: The Global Grid Resource Brokers (GGRB), Local Grid Resource Brokers (LGRB) and Grid Information Server (GIS) [2].The above listed components have their independent functionalities that help in management of the grid and subtask scheduling and thus serve the purpose of a grid.

2.1. Local Grid Resource Broker (LGRB) and Global Grid Resource Broker (GGRB)

The local grid resource broker is a synonym for a grid resource. Each grid resource is categorized [4] based on the node's processing speed as follows:

Type 1 – TFLOPS machines
Type 2 – GFLOPS machines
Type 3 – MFLOPS machines

This categorization adds to the heterogeneous nature of a grid. Each LGRB in the grid can be any one of the above three types in a resource. The number of processing elements in an LGRB is the actual resource of the grid. A subtask submitted to the grid may be migrated to any of the LGRBs [5] in the grid for execution based on the global scheduling algorithm. Once a subtask has been migrated to a particular LGRB, the LGRB ensures execution of the subtask on the specified number of processors. A single LGRB takes care of scheduling subtasks in the grid based on the nodes available in the resource as per the local scheduling algorithm.

2.2. Grid Information Server (GIS)

It is the database of the grid. It keeps track of the resources available in the grid. New LGRBs should register itself with the GIS. The GIS provides information regarding free resources to the GGRB based on which the GGRB schedules the job.

2.3. Grid Working and Registration

A new LGRB should register itself with the GIS by sending a request [2]. The response of the GIS comes with an acknowledgement, which means that it is ready to accept a new resource as a grid member. Now, the LGRBs send the details its type, number of processing elements and speed of each processing element.

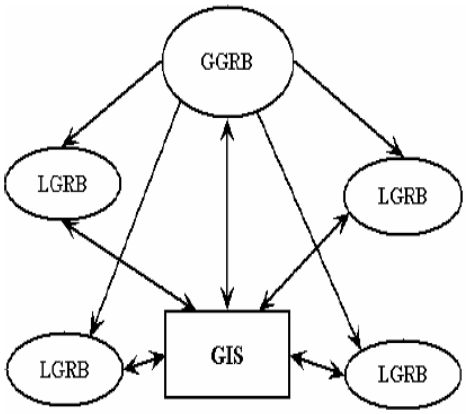


Figure 1. Grid Framework

3. THE LOCAL SCHEDULING ALGORITHM

Since a Grid has heterogeneous resources it is often complex to design an efficient scheduling algorithm.

3.1.MidSFN Local Scheduling Algorithm

As the proposed MidSFN local scheduling algorithm is a revised version of our Performance Factor based local scheduling algorithm, a parameter named "priority" has been taken into consideration. The priority assignment is done based on the computational complexity of the subtask and performance factor of the resource. In the Performance Factor based local scheduling algorithm the subtasks which exhibit higher computational complexity and require nodes that possess low performance factor are assigned with a high priority. Similarly the MidSFN algorithm classifies the subtask into high, medium and low categories based on the priority. The value of Processing Power is based on the Performance Factor. The value of the Performance Factor is the product of the number of operations per cycle per processor and the number of instructions processed per second. In the MidSFN algorithm a higher priority [1] is assigned to subtask of medium computational complexity and the node of medium processing power available in the resource .The fastest node available in the resource is assigned to the

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

subtask of high priority. This MidSFN algorithm optimizes the computational speed of the grid and reduces the usage of nodes and also shows a consistent performance during execution of the assigned subtask.

3.2. Computational Complexity

Task partitioning algorithm takes care of efficiently dividing an application into tasks of appropriate grain size and an abstract model of such a partitioned application is represented by a [2] Directed Acyclic Graph (DAG). Each task of a DAG corresponds to a sequence of operations and a directed arc represents the precedence constrains between the tasks. Each task can be executed on a processor and the directed arc shows transfer of relevant data from one processor to another. Each node in DAG represents sequence of operations. All the operations are represented in terms of additions. Node weight represents the amount of computations [6] (in terms of additions) involved in the particular node. This denotes the Computational Complexity of the subtasks.

3.3. Priority Assignment

In MidSFN algorithm higher priority is assigned to a subtask which requires medium processing power and which exhibits medium computational complexity. A subtask, which exhibits [2] low computational complexity and needs low processing power for execution is given a low priority. A subtask, which exhibits a high computational complexity and needs high processing power, is given a medium priority. The fastest free resource available in the grid is allocated to the subtask which has high priority. The procedures are given below.

3.4. Algorithm

Assign Performance Factor (ResourceListRs_List)

While (Rs_List! =NULL)

For each resource

/*OC = No. of operations per cycle per processor

SP = Speed of the processor

Performance FactorList contains the performance factor for each node available and sorted in descending order in the resource*/

Performance Factor_List[i] = OC*SP

End While

Find the Max, Min and Mid ranges in Factor_list

For each subtask in Factor_List

If Performance Factor_List[i] >= Maximum

Final_List[i] = High

/*Final_List contains the value of performance factor calculatedand sorted in descending order for each node available in the resource*/

Else If Factor_List[i] >= Middle

Final_List[i] = Medium

Else Final_List[i] = Low

EndIf

End Assign Performance Factor

Assign Priority Procedure

AssignPriority (Local_List)

While (Local_List!=NULL)

/*Local_List contains the list of nodes available in the Grid resource*/

For each subtask

/* CompC_List contains the Computational Complexity of subtasks*/

If (CompC_List[i] =Medium AND Final_List[i] = Medium)

Priority[i] = 1

Else If (CompC_List[i] = High AND Final_List[i] = Medium)

Priority[i] = 2

Else If (CompC_List[i] = Low AND Final_List[i] =Medium)

Priority[i] = 3

Else If (CompC_List[i] = Medium AND Final_List[i] = High)

Priority[i] = 4

Else If (CompC_List[i] = High AND Final_List[i] = High)

Priority[i] = 5

Else If (CompC_List[i] = Low AND Final_List[i] = High)

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

Priority[i] = 6

Else If (CompC_List[i] = Medium AND Final_List[i] = Low)

Priority[i] = 7

Else If (CompC_List[i] = High AND Final_List[i] = Low)

Priority[i] = 8

Else If (CompC_List[i] = Low AND Final_List[i] = Low)

Priority[i] = 9

EndIf

End AssignPriority

Let m represent number of free nodes available in the resource and n represent the number of subtasks of a job present in the queue. The worst case time complexity of the algorithm is O (n log n) when m <= n and O (m log m) when m > n.

4. PERFORMANCE STUDY

We compare the performance of our algorithm with First Come First Serve, Shortest Subtask Fastest Node, Longest Subtask Fastest Node and Performance Factor based algorithm.

4.1. First Come First Serve (FCFS)

This scheduling algorithm schedules the subtask on a "First come First serve" basis. From the Fig 2, we get that FCFS algorithm is basic and is not based on the factors like computational complexity and performance factor .It shows very low computation results compared to other scheduling algorithms.

4.2. Shortest Subtask Fastest Node (SSFN)

The Shortest Subtask Fastest Node algorithm [1] assigns the subtask with a smaller value of computational complexity to the fastest node available in the resource. Shortest Subtask Fastest Node is a scheduling algorithm, which tries to reduce the overall turnaround time of the subtask. From the Fig.2 we can decipher that SSFN is more stable in handling subtask and hence outperforms FCFS scheduling algorithm.

4.3. Longest Subtask Fastest Node (LSFN)

The scheduling algorithm, commonly used for the assigning of complex subtasks to high efficiency resources is the Longest Subtask Fastest Node (LSFN) algorithm. It tries to reduce the overall [1] execution time of the subtask. From Fig.2 of the LSFN algorithm we can infer that LSFN outperforms FCFS and the SSFN as the subtasks are assigned to faster nodes in the resource which leads to shorter execution time.

4.4. Performance Factor based Local Scheduling Algorithm (PF)

Our Performance Factor algorithm is based on a concept "priority". In the Performance Factor based algorithm [2] the priority is assigned by considering the new parameters, computational complexityof the subtask and the value of Performance Factor of the node in the resource. The value of the Performance Factor is assigned based on the number of operations per cycle per processor and the number of instructions processed per second. In this a subtask which requires high processing power and which exhibits high computational complexity is given a high priority. A subtask, which exhibits high computational complexity and requires low processing power, is given a low priority.

A subtask, which exhibits a medium computational complexity and requires medium processing power, is given a medium priority. The fastest free node available in the resource is allocated to the subtask which has high priority. From this we can state that Performance Factor based local scheduling algorithm outperforms FCFS, SSFN and LSFN due to its enhanced usage of resources.

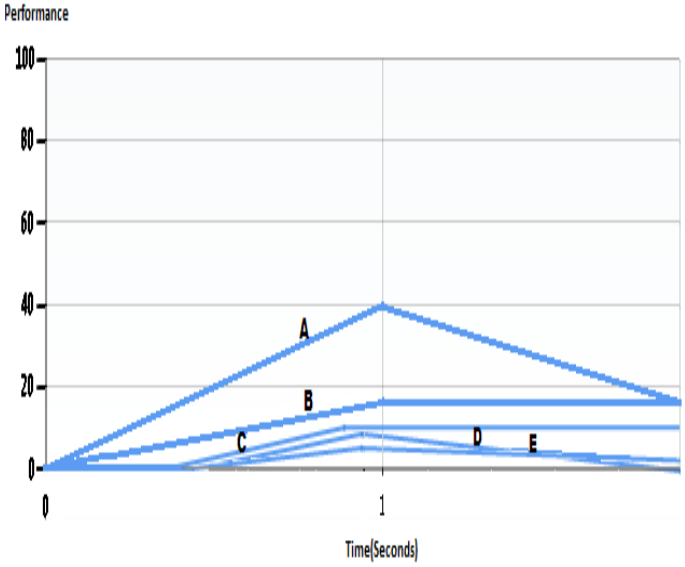
4.6. MidSFN Algorithm

The MidSFN algorithm is based on the parameter 'priority' for the assignment of subtask to the resources. The subtask is prioritized on the basis of computational complexity and processing power. The subtasks with medium computational complexity and medium processing power are given a first priority. This method of prioritizing enhances the rate of completion of subtask with a greater accuracy and with a proper usage of resources.

From the Fig.2 of the MidSFN algorithm, we can decipher that MidSFN algorithm outperforms the Performance

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

Factor based local scheduling algorithm due to its optimized resource allocation based on the computational complexity and processing power. So from this performance graph we can infer that it can perform execution of given subtasks in a shorter time when compared with FCFS, SSFN, LSFN and Performance Factor based algorithm.



A - MidSFN, B – PF, C - LSFN, D - SSFN, E- FCFS

Figure 2.Performance Graph for FCFS, SSFN, LSFN, Performance Factor, and MidSFN Local Scheduling algorithm

4.7. Comparative study between MidSFN Algorithm and other algorithms

MidSFN provides Maximum economic utilization of resources that are provided for the execution of subtasks compared to other task scheduling algorithms like FCFS, SSFN, LSFN, and Performance Factor based algorithm. This gives the best execution results among all the algorithms. It helps in scheduling tasks that exhibit medium computational complexity. MidSFN also optimizes the allocation of resources for completion of complex tasks with comparatively higher performance than LSFN, SSFN and Performance Factor based algorithm.

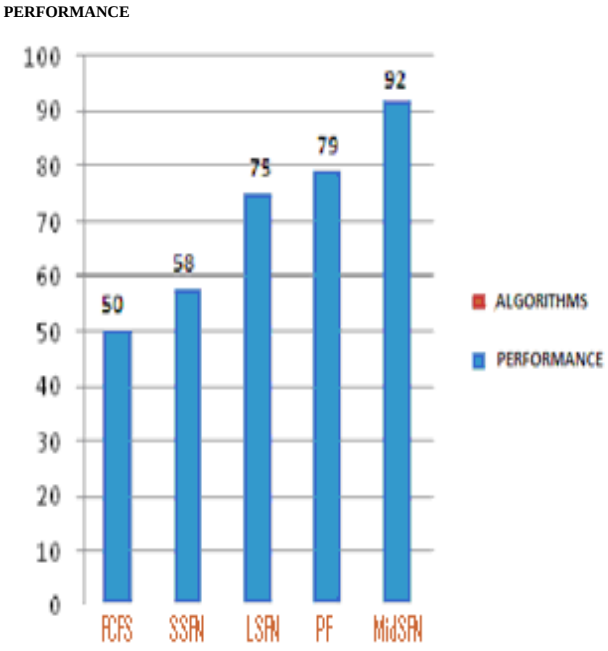


Figure 3.Performance Chart

Fig.3 explains about the performance of various local scheduling algorithms .Among all the above listed algorithms, MidSFN as seen from the graph is the best algorithm that provides efficient load balancing and better computation with effective usage of resources.

5. Conclusions

The process of designing a proper local scheduling algorithm with an aim to optimize the performance has been a complex subtask. The FCFS algorithm [5] schedules the subtask without considering the factors like computational complexity and performance factor of the node and hence it shows low efficiency. The two fundamental algorithms SSFN and LSFN schedule the subtask based on computational complexity of the [2] subtask and the speed of the node in the resource. In the Performance Factor based local scheduling algorithm parameter named “priority”is used in the analysis and the subtasks are classified into high, medium and low categories .In the Performance Factor algorithm the subtask that possesses a high computational complexity and requires the nodes that exhibit high performance factor is given a high priority. As the MidSFN Algorithm is revised version of the Performance Factor algorithm the subtask that exhibits medium Computational Complexity and requires medium processing power are given the high priority. This method of prioritizing the subtask leads to completion of the subtask with high efficiency, shorter execution time with the usage of lesser number of

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

resources and also shows consistency during the execution of the assigned tasks. The effectiveness of our algorithm is evaluated through simulation results and its superiority over other known algorithms is demonstrated.

6. References

[1]G.Sumathi,R.Santhosh Kumar,S.Sathyanarayanan,"MidJFR Global Scheduling Algorithm for Heterogeneous Grid Environment",IJRTET,November,2011

[2]G.Sumathi,S.Sathyanarayanan,R.Santhosh Kumar " Performance Factor based Local Scheduling for Heterogeneous Grid Environments", International Journal of Computer Applications (0975 – 8887),May 2012

[3] G.Sumathi, N.P. Gopalan, "Priority Based Scheduling For Heterogeneous Grid Environments", Proc. Of 10th IEEE InternationalConference on Communication Systems (ICCS 2006), October, 2006.

[4] AmitAgarwal, Padam Kumar, Economical Task Scheduling Algorithm for Grid Computing Systems, GJCST Classification(FOR) D.4.1, F.1.2

[5] T. Kokilavani, Dr. D.I. George Amalarethinam, Applying Non-Traditional Optimization Techniques to Task Scheduling in GridComputing an Overview International Journal of Research andReviews in Computer Science (IJRRCS) Vol. 1, No. 4, December2010

[6] Zhan Gao, SiweiLuo and Ding Ding, A Scheduling ApproachConsidering Local Tasks in the Computational GridInternational Journal of Multimedia and Ubiquitous EngineeringVol. 2, No. 4, October, 2007

[7]Kousalya.K and Balasubramanie.P, Ant Algorithm for Grid Scheduling Powered by Local Search Int. J. Open Problems Compt.Math., Vol. 1, No. 3, December 2008

[8] Fangpeng Dong and Selim G. Akl, Scheduling Algorithms for Grid Computing: State of the Art and Open Problems TechnicalReport No. 2006-504 .

[9] S.Padmavathi, S.MercyShalinie and R.Abhilaash, A MemeticAlgorithm Based Task Scheduling considering Communication Coston Cluster of Workstations Int. J. Advance. Soft Computing.Appl.,Vol. 2, No. 2, July 2010 ISSN 2074-8523.

[10] Topcuoglu,H.,S.Hariri and M.Y.Wu, "Performance Effective And Low Complexity Task Scheduling Algorithm scheduling forheterogeneous computing ", IEEE Transaction on Parallel andDistributed Systems, Vol.13,No.3,(2002).

[11] P.J. Huang, H. Peng, X.Z. Li, Macro adjustment based task schedulinginhierarchical Grid market, in: Proceedings of the 7th International Conference on Computational Science, ICCS 2007, Beijing, China, in: Lecture Notes inComputer Science, vol. 4487, Springer-Verlag, 2007, pp. 430_433.

[12] G. Stuer, K. Vanmechelena, J. Broeckhovea, A commodity market algorithm forpricing substitutable Grid resources, Future Generation Computer Systems 23(5) (2007) 688_701.

[13] C.L. Li, L.Y. Li, QoS based resource scheduling by computational economy in computational Grid, Information Processing Letters 98 (3) (2006) 119_126.

[14] R. Subrata, A.Y. Zomaya, B. Landfeldt, Artificial life techniques for load balancing in computational Grids, Journal of Computer and System Sciences73 (8) (2007) 1176_1190.

[15] C.H. Hsu, T.L. Chen, K.C. Li, Performance effective pre-scheduling strategy for heterogeneous Grid systems in the master slave paradigm, Future GenerationComputer Systems 23 (4) (2007) 569_579.

Dr. G. Sumathi obtained her B.E. degree in Electronics and Communication from Bharathidasan University, M.E. degree in Computer Science and Engineering from Regional Engineering College, Tiruchirappalli and Ph.D in Computer Science and Engineering from National Institute of Technology, Tiruchirappalli. She had been trained at Carnegie Mellon University, Pittsburgh, U.S.A. Presently, she is working as Professor in the Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur, Tamil Nadu, India. She is the life member of ISTE & CSI. Her research interest includes Cluster, Grid & Cloud Computing and Networks.

R. Santhosh Kumar is studying second year B.Tech Information Technology in Sri Venkateswara College of Engineering, Sriperumbudur, Tamil Nadu, India. He has published two papers in International Journals.

S. Sathyanarayanan is studying second year B.Tech Information Technology in Sri Venkateswara College of Engineering, Sriperumbudur, Tamil Nadu, India. He has published two papers in International Journals. He is the member of IEEE & CSI and Microsoft Student Partner.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.