Volume 3, Issue 11, November 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Task Assignment in Heterogeneous Environment with Advanced Scheduling Algorithms

Mrs. Minal Shahakar, Prof. Rajesh Bharati, Prof. Rupesh Mahajan
Dept. of Computer Engg.
DYPIET, Pune, India

Abstract - This paper presents an innovative idea of scheduling the tasks to their best processor to reduce the execution time of each task by using multiple heuristics techniques. This paper presents hybrid heuristics techniques which provide better solution of scheduling task. Heuristics such as MinMin+, MaxMin+ and Sufferage+ overcomes the drawback of previously used heuristics methods such as MinMin, MaxMin and Sufferage as well as heuristics in this paper provides better complexity as compare to previous heuristics methods. Later the hybrid heuristics means combination of different heuristics provides better performance without degrading the result quality. This heuristics can also be used in heterogeneous environment more effectively to execute different set of task on different processors with different configurations.

Key Terms - MinMin, MaxMin, Sufferage, Task Scheduling, Standard Deviation, Load Balancing.

I. INTRODUCTION

Scheduling is one of the important process in the area of distributed computing. It performs an important step of mappings task to different machines based on the expected execution time. Normally an application is used to define the piece of work of higher level in heterogeneous environment. Since this application can generate several number of jobs that can be divided into subtasks and provided to different processors that should get completed within minimum time so that the processor use can be made to assign different task. Makespan is one of the most important term in case of mapping task to their processors using different heuristics. Makespan is nothing but turnaround time that is maximum of completion time. An optimal schedule will be the one that minmizes the makespan [1, 2].

The existing heuristics provides the various techniques for assigning different task to different processors with minimum completion time. This existing heuristics can be divided into two classes that are Online mode and Batch mode heuristics. In online mode, a task is assigned to processor as soon as it arrives at the scheduler [1]. Wherein Batch mode heuristics tasks are not assigned to processor immediately instead they are collected in to set of tasks also called as Metatask that are examined for assigning at prescheduled times to different processors also called as mapping events. Since in this work, batch mode is used in very efficient way for mapping different independent tasks to processors. Also this existing heuristics can be applied for heterogeneous environment effectively. This paper presents an efficient heuristic method that is hybrid method also known as RASA algorithm. Overall the algorithm aims to minimize the idle time and makespan of tasks. This paper also involves the concept of Load Balancing [2, 6], wherein, once scheduling of task is done using some heuristics the load balancing algorithm will take place to reschedule the task to utilize all the resources in the heterogeneous environment [5].

Also the proposed system in this work contains various heuristics method along with hybrid technology such as Minmin+, Maxmin+, and Sufferage+. Each of this heuristics provides better performance and also increases complexity without degrading the solution quality.

II. HEURISTICS DESCRIPTION

Many algorithms have been designed for mapping of task on different processors. Since existing heuristics methods aims at providing best quality of solution without degrading system performance. At the same time it aims to minimize the makespan, as it provides the task assignment alternatively depending on expected execution time and task assignment. In earlier distributed computing systems comprised of homogeneous and dedicated resources. This heuristics will work well even for heterogeneous resources also. Heterogeneous systems provides with the facility of utilization of all available resources as load balancing concept that aims at keeping resources busy [5].

A. Heuristics

Minimum Completion Time (MCT) assigns tasks to different processors in arbitrary order, with minimum expected completion time for that task. Since this causes some of the task to be assigned to the processor that do not have minimum execution time. For this purpose this minimum completion time is defined in a way that combines the benefit of both opportunistic load balancing (OLB) and minimum execution time (MET) to provide better performance of task mapping [3].

Opportunistic Load Balancing (OLB) is specially used to keep all the processors busy that is to make utilization of all the available resources by assigning task in arbitrary order, to the next available processor, without considering task expected execution time on that particular processor but this results in poor makespan [6].

Minimum Execution Time (MET) assigns tasks to processor in arbitrary order with best expected execution time for that task, without taking into consideration processors availability. Since assigning the task to its best processor provides better performance but causes severe load imbalancing and does not provide support for heterogeneous environment [4, 5].

B. Minmin heuristics

Minmin heuristic starts with set of metatask that contains all unassigned task that has minimum completion time. Since for this purpose it requires to calculate minimum completion time i.e. MCT of each task in MT that is metatask. Once minimum completion time is found in the first step this heuristic find minimum expected completion time of each task in metatask and finally the task with overall minimum expected completion time is found and assigned to the respective available resource [6]. When assigned to the resource it is removed from the metatask that is set of task and this process is repeated until all the tasks are assigned to resources. Since this method provides easiest way to assign task to processor but has one drawback due to selection of task having minimum expected completion time, the task with largest expected completion time remains unassigned for longer time and also load is not balanced across the systems, due to which some resources remains idle and this also results in increase in makespan.

C. Maxmin heuristics

Maxmin heuristic is similar to minmin heuristic wherein it also starts with mapping independent task to different machines. The set of tasks is found that having minimum completion time from metatask. And the task with overall maximum completion time also known as makespan is selected and assigned to the available resource. This process is continued until all the tasks are assigned to respective available machines. Since this heuristics provides the way of mapping task to its best machine with longer execution time first allows this task to be executed concurrently with the task that having shorter execution time. Since this mapping of task to resources is better than minmin heuristic wherein the task with smaller execution time is selected and assigned to the available resource for execution and then task with longer execution time are executed while several machines sit idle. Since maxmin heuristic provides better load balancing across machines as well as better makespan [6, 7].

D. Sufferage Heuristics

Sufferage heuristic differs with previous heuristics in the sense of task selection process. Like minmin and maxmin it also begins with set of unassigned task that has minimum completion time i.e. sufferage heuristic is also based on the concept of minimum completion time since it differ from the previous heuristics in the sense it selects and assigns the task to the processor on the basis of sufferage value and not minimum or maximum completion time. Since it computes second MCT value instead of computing MCT value for each task and calculates sufferage value which is defined as difference between MCT and second MCT values of a task is taken into account. This heuristic selects the task with largest sufferage value and assigns it to available resource. Thus sufferage heuristic differs from minmin and maxmin heuristics in the task selection policy [4, 5].

III. ADVANCE HEURISTICS

This type of heuristics provides better performance than the previous heuristics without degrading solution quality. That is previously mentioned heuristics increases time complexity by using number of iterations for computing minimum completion time of each task. Wherein, the advance heuristics these MCT values of each task are maintained separately that reduces the number of iterations as well as reduces time complexity along with makespan and makes processing faster.

A. Minmin+ heuristics

Minmin+ heuristics uses different methods for initialization of necessary variables and for selection of task with minimum completion time. Also it maintains a separate queue wherein all the MCT values are arranged in sorted order. Thus minmin+ heuristics first makes necessary initializations, select tasks with minimum completion time for mapping to processor and once task is selected and assigned to processor it is deleted from queue.

For the implementing this priority queue two alternatives are being considered that are binary heap and sorted linear array, and also some operations are being used like sorting operation, deletion operation, and necessary check is made on the queue to know which task has not being yet assigned to available resource that is with minimum completion time. Hence the overall running time complexity is reduced.

B. Maxmin+ heuristics

This heuristics is similar to minmin+ heuristics except it differs in the way of selecting task. Since it requires MCT values of task but assignment of task to respective available resource is done on the basis of makespan that is maximum of completion time. Thus it does first selection of task to processor assignment is done using minmin+ heuristic only. That is necessary initializations of variables and task selection is done using the same methods that are used in minmin+ heuristics.

The computed assignment is realized only if it does not lead to increase in makespan of previous iteration. Since if computed assignment increases in makespan of previous iteration then task assignment to processor is recomputed

according to maxmin heuristics. This heuristic overcome drawback of maxmin heuristic of task assignment problem to same processor by doing the combination of maxmin with minmin+ under a hybrid heuristic that is maxmin+.

C. Sufferage+ heuristics

Sufferage + working is similar to sufferage heuristic since to make applicable sufferage heuristic to large datasets it is combine with minmin+ heuristic under a new heuristic that is sufferage+. Here also task assignment is done according to minmin+ heuristic only that is initialization and task selection for assignment to processor. This heuristic differs from previous heuristic in the sense that when assignment is computed, it is realized only if it does not, lead to increase in makespan of previous iteration otherwise assignment is recomputed using sufferage heuristics.

D. Switcher Heuristics

As the name indicates it is the combination of different heuristics also known as hybrid heuristics. Switcher heuristics is based on concept of standard deviation value comparison with threshold value. Based on this it switches between heuristics that is if standard deviation [6] value is less than threshold value than tasks are considered to be with minimum execution time and minmin heuristics is applied to assign the tasks to available resources, otherwise maxmin heuristics is used to assign the tasks to available resources. This process is repeated until all the tasks are assigned to their respective available resources [3, 13]. There are many different types of hybrid algorithms that call alternatively different heuristics for mapping tasks to their best processors. This type of heuristics also maintains the proper load balance across the processors due to which all the available resources get fully utilized and no resource remains an idle.

IV. STANDARD DEVIATION

Standard deviation concept is specially used for hybrid heuristics that is combination of different heuristics. Wherein, the standard deviation value is compared with threshold value to check which heuristic to be applied for mapping of task to different resources [6]. Since the standard deviation value is calculated on the basis of average of completion time of all tasks, as mention below:

$$avgCT = \frac{\sum_{i-1}^{s} CT_{ij}}{s}$$

Where avgCT denotes average of completion time that is sum of all completion time of given tasks and s is nothing but index of task. Using this average value standard deviation is calculated as:

$$sd = \sqrt{\frac{\sum_{i-1}^{s}(CT_{ij} - avgCT)^2}{s}}$$

Based on above mentioned formulae standard deviation is calculated. Since this is compared with the threshold value in case of hybrid heuristics wherein the multiple heuristics are called by algorithm alternatively for mapping task to processors.

This hybrid heuristic will use standard deviation concept wherein if the calculated standard deviation is less than threshold value then that particular task is assigned using the minmin heuristic to available resource otherwise the task is assigned to available resource using maxmin heuristic. Since after the assignment of task to resource it will be deleted from set of task that is metatask and the hybrid heuristic will repeat all the process until all the task are assigned to processors.

This standard deviation mentioned above can also be represented in another way that is in the relation as mention below:

$$sd = \sqrt{E\left(CT_{ij}^2\right) - E(CT_{ij})^2}$$

Where $E(x_i)$ denotes the average of x_i .

V. LOAD BALANCING

Load Balancing concept takes place in distributed system to keep all the resources busy that is all the resources should get utilized so that task execution will become faster and time complexity will be reduced. Heuristics mentioned above like minmin and maxmin maps different tasks to different available resources efficiently but it does not maintain proper load balancing among the resources due to which some resources are utilized and some remain idle. This load balancing concept can be applied to this heuristics to get done execution faster [6]. Minmin heuristics selects tasks with minimum completion time and allocates it to available resource, due to which task with longer execution time remains unassigned although the resource is available that causes resources to remain idle. Similarly in maxmin heuristics tasks with maximum completion time is selected and assigned to processor, due to which smaller tasks are assigned after long time to available processors [7]. Solution for above is to apply load balancing concept with this types of heuristics. This can be done when tasks are assigned to their resources that is once minmin heuristics is applied to assign tasks to available resources using minimum completion time, the load balancing method is applied again on this assigned task for rescheduling it i.e. it may happen that minmin heuristic will use some resources to assign task and some remain idle then load balancing method will select the task with maximum completion time that will be less than makespan produced by Minmin heuristic and reschedule it to the resource that is available and not utilized yet so that execution of tasks will be more faster [11]. Other tasks maximum completion time is not less than makespan. So whichever task has maximum completion time less tan makespan is selected and rescheduled to available resource.

VI. EXAMPLE OF LOAD BALANCING

Consider a heterogeneous environment with two resources R_1 and R_2 and metatask that contain four different tasks T_1, T_2, T_3 and T_4 as shown below in table1 that contains tasks, resources along with expected execution time for mapping tasks to their respective resources.

Table 1: Resources and Tasks with Expected Execution Time

Tasks	Resources	
	R_1	**R_2**
T_1	7	2
T_2	13	3
T_3	14	2
T_4	11	3

As shown in above table task assignment is done to different processors using minmin heuristics is done in following way.

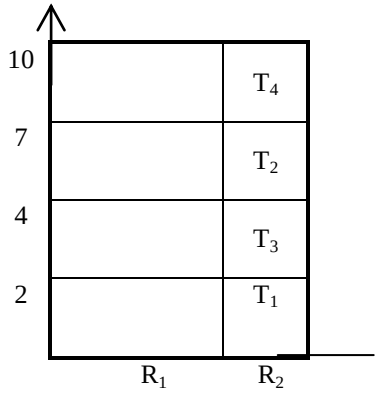

Figure 1: Mapping of Tasks to Resources with Minmin Algorithm

As shown in figure1 minmin heuristic will select tasks according to given execution time so task T_1 will be assigned first to Resource R_2, then task T_3 will be assigned again on resource R_2, then task T_2 will be assigned again on resource R_2 and finally the remaining task that is task T_4 will also be assigned to resource R_2 only according to given expected execution time in Table1. Since on resource R_2 task completion is faster than on resource R_1, so all the task will be assigned on resource R_2 only.

Once minmin heuristic is applied for mapping tasks to available resources, load balancing technique is applied for again rescheduling tasks to make utilization of idle resources that minimizes overall task completion time that is makespan.

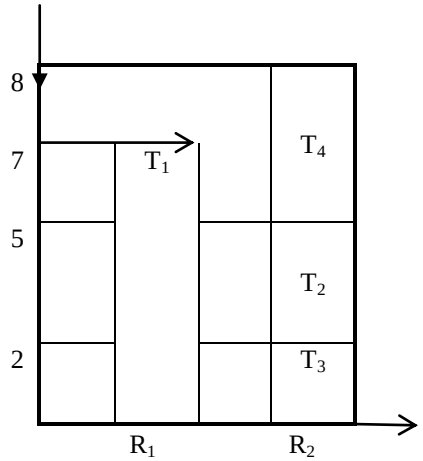

Figure 2: Rescheduling of tasks to Resources with Load Balancing method

As shown in figure2 task T_1 is rescheduled to balance the load as it provides maximum completion time on resource R_1 as shown in Table1 as well as it is less than makespan produced by minmin heuristics. While remaining tasks although have maximum completion time but are not less than makespan. So task T_1 is rescheduled on resource R_1 that results in better makespan as compared to minmin heuristic.

VII. CONCLUSION

The goal of this paper was to present various heuristics methods like minmin, maxmin, sufferage, hybrid, load balancing techniques in the field of distributed systems. The heuristics like minmin and maxmin are suitable for small scale distributed systems but when number of tasks increases than these heuristics cannot schedule task appropriately that affects on makespan which relatively become large. To overcome limitations of these heuristics and make them applicable for large scale distributed systems, a new task scheduling algorithm like minmin+, maxmin+ and sufferage+ along with hybrid heuristics are used that also maintains proper load balancing across the systems. This heuristics uses advantages of minmin and maxmin and covers there disadvantages. This study can be further extended by considering task heterogeneity and machine heterogeneity.

REFERENCES

[1] T. D. Braun,H. J. Siegel,N. Beck, L. L. Boloni, "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems", J. Parallel Distrib. Comput., vol. 61, no. 6, pp. 810837, 2001.

[2] P. Luo, K. Lu, and Z. Shi, "A revisit of fast greedy heuristics for mapping a class of independent tasks onto heterogeneous computing systems", J. Parallel Distrib. Comput., vol. 67, pp. 695714, 2007.

[3] Kamali Gupta, Manpreet Singh, "Heuristic Based Task Scheduling In Grid", International Journal of Engineering and Technology (IJET), vol. 4, pp. 254260, Aug-Sep 2012.

[4] M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen, and R. F. Freund, "Dynamic mapping of a class of independent tasks onto heterogeneous computing systems", J. Parallel Distrib. Comput., vol. 59, pp. 107131, 1999.

[5] H. J. Siegel and S. Ali, "Techniques for mapping tasks to machines in heterogeneous computing systems", J. Syst. Archit., vol. 46, no. 8, pp. 627639, 2000.

[6] T. Kokilavani, Dr. D.I. George Amalarethinam, "Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing", International Journal of Computer Applications, vol. 20, April 2011.

[7] George Amalarethinam. D.I, Vaaheedha Kfatheen .S, "Max-min Average Algorithm for Scheduling Tasks in Grid Computing Systems", International Journal of Computer Science and Information Technologies, Vol. 3, pp. 3659-3663, 2012.

[8] K. Kaya, B. Ucar, and C. Aykanat, "Heuristics for scheduling file-sharing tasks on heterogeneous systems with distributed repositories", J. Parallel Distrib. Comput., vol. 67, no. 3, pp. 271285, 2007.

[9] Doreen Hephzibah Miriam. D and Easwarakumar. K.S, "A Double MinMin Algorithm for Task Metascheduler on Hypercubic P2P Grid Systems", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 4, No 5, July 2010.

[10] He. X, X-He Sun, and Laszewski. G.V, "QoS Guided Minmin Heuristic for Grid Task Scheduling", Journal of Computer Science and Technology, Vol. 18, pp. 442-451, 2003.

[11] Kamalam.G.K and Muralibhaskaran.V, "A New Heuristic Approach: MinMean Algorithm For Scheduling Meta-Tasks On Heterogenous Computing Systems", International Journal of Computer Science and Network Security, VOL.10 No.1, January 2010.

[12] Sameer Singh Chauhan,R. Joshi. C, "QoS Guided Heuristic Algorithms for Grid Task Scheduling", International Journal of Computer Applications (09758887), pp 24-31, Volume 2, No.9, June 2010.

[13] Singh. M and Suri. P.K, "QPS A QoS Based Predictive Max-Min, Min-Min, Switcher Algorithm for Job Scheduling in a Grid", Information Technology Journal, Vol. 7, pp. 1176-1181, 2008.

[14] Yagoubi. B, and Slimani. Y, "Task Load Balancing Strategy for Grid Computing", Journal of Computer Science, Vol. 3, No. 3, pp. 186-194, 2007.

