ISSN: 2321-7782 (Online)
Volume 2, Issue 1, January 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Paper
Available online at: www.ijarcsms.com

Advance Heuristics for Task Assignment Using Heterogeneous Resources

Minal Shahakar[1]
Pune University
Department of Computer Engg
DYPIET
Pune - India

Rajesh Bharati[2]
Pune University
Department of Computer Engg
DYPIET
Pune - India

Rupesh Mahajan[3]
Pune University
Department of Computer Engg
DYPIET
Pune - India

Abstract: ***Task assignment occurs when larger applications take place, which is divided into different tasks and distributed among different resources. This assignment is done on the basis of heuristics methods. Advance heuristics are Minmin+, Maxmin+, and Sufferage+, which reduce time complexity without degrading solution quality. Hybrid heuristic is one of the methods for task assignment that is the combination of different heuristics which works more faster as well as achieves load balancing to make utilization of all the available resources. All this advance heuristics also works well in heterogeneous environment that is processors with different configurations. The major objective of this system is to decompose the larger tasks into smaller tasks, which are independent tasks. The performance of the system is measured by the reduction in the idle time and makesspan. Since this heuristics are more efficient than the other heuristic algorithms by reducing the idle time and make span.***

Keywords: Minmin+, Maxmin+, Sufferage+, Hybrid Heuristic, Load Balancing, Heterogeneity.

I. INTRODUCTION

Scheduling of task is one of the important concepts in the area of distributed systems. It occurs whenever the large applications take place. Since this application is decompose into smaller task and scheduled to respective available resources. In this system various heuristics methods are used to schedule task to resources maintaining proper load balancing to make utilization of all the available resources. This task assignment is done on the basis of makespan. Makespan is an important term for the various heuristics methods. It is nothing but maximum of completion time of respective resource. Various types of heuristics methods in this system are like minmin, maxmin, and sufferage. These systems are already existing systems whereas advance heuristics are like minmin+, maxmin+, and sufferage+ that perform task assignment faster reducing time complexity. Since Hybrid heuristic is one of the concept among these heuristics. It is nothing but combination of various heuristics methods [1, 2].

These heuristics can be divided into two classes Online mode and Batch mode heuristics. In online mode, a task is assigned to processor as soon as it arrives at the scheduler. 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, this system works well in batch mode for assigning task to processors. Heuristics in this system also works well in heterogeneous environment. Earlier systems was based on homogeneous environment due to which all available resources was not utilized in proper way wherein heterogeneous environment all the

resources are used achieving load balancing. Existing system works well only for small scale application that is whenever larger tasks are provided it produce poor performance, for this purpose only advance heuristics are used that works well for large scale applications providing better solution quality without degrading the system performance [3, 4].

Some common terms that are used throughout this system are minimum completion time, opportunistic load balancing and maximum completion time that is makespan, based on these tasks are assigned to processor. Minmin heuristic selects tasks with minimum completion time and assigns it to available resource but due to which tasks with longer completion time remains unassigned and some of the resources remains idle and load among the resources is not balanced properly [4]. Similarly remaining heuristics assigns task to resource that are described in this system have some disadvantages so heuristics like minmin+, maxmin+ and sufferage+ are used. This heuristics make use of queue implementation that will contain MCT values of all task in sorted order, which reduces time of calculating MCT value for each task, so these values are directly used further to assign task to processor. Once task is assigned to respective available resource it is deleted from queue, so that it indicates only those tasks that are not yet assigned to any processor. This process is continued until all the tasks are assigned to respective available resources. These heuristics are explained in detail in section 2.

II. Related Work

Many scheduling algorithm have been proposed for assigning task to different available resources taking into account load balancing of available resources. Batch mode is very suitable for existing scheduling algorithms that also work well in heterogeneous environment. Since this task assignment is related to some heuristics as mentioned below:

MET is nothing but minimum execution time that schedules each task to available resource that executes it in least amount of execution time. But it doesn't take into account whether the resource is available or not and this results into severe load imbalance i.e. load is not balanced properly due to which some resources remains idle [4].

MCT that is minimum completion time allocates task for execution to available resource which obtains earliest completion time for that particular task. But this causes one drawback i.e. it will also allocate some tasks to the resources that doesn't have minimum execution time [2].

OLB is very important term that is opportunistic load balancing is used to balance the load among various resources so that all the available resources are utilized specially in heterogeneous environment. Since this assign the task to available resource without considering execution time of task on that resource. So whenever one or more than one resource becomes ready, one resource is arbitrarily chosen for allocation of task [4, 5].

Minmin heuristic works on the basis of minimum completion time. As it name indicates, it selects the task that has minimum completion time. For this it begins with calculating the MCT of each task. Once this calculation is done it selects the task with overall minimum completion time and assigns that task to available resource. The collection of tasks is defined as a set of tasks also known as metatask. So once allocation of particular task to available resource is done it removes that task form metatask. And this process is continued until all the tasks are assigned to the resources [4].

Maxmin heuristic works similar to minmin heuristic, except it differs in task selection policy. That is it also begins with calculation of MCT for each task but it uses term makespan that is, it requires overall maximum completion time of task. Since, it differs from minmin in task selection policy. Thus once MCT is calculated of each task, the task with overall completion time is found and allocated to respective available resource. This process is done until all the tasks are assigned to available resources [5].

Sufferage heuristic makes use of sufferage value i.e. again task selection policy is different from minmin and maxmin heuristics. This heuristic calculates MCT of first and second task and difference between these two is defined as sufferage value i.e. in other words difference between first MCT and second MCT. Based on the maximum sufferage value task is selected and

allocated to respective available resource. This process is continued until all the tasks are assigned to available resources. Since rest of all the process is same that is calculating MCT similar to minmin and maxmin except it makes use of sufferage value instead of makespan [4, 5].

III. Problem Description

A scheduling problem is considered where tasks are to be allocated immediately to the available resources in a heterogeneous environment [8]. The tasks have to be assigned and completed on a unique resource. There are no dependencies between tasks that is each task is independent. In the proposed system, algorithms have been developed under a set of assumptions as mentioned below:

- The applications that should be executed are collection of indivisible tasks that are independent form each other, usually referred to as metatask.
- Tasks don't have any deadlines or priorities allocated with them.
- Each available resource executes a single task at a time in the order in which the tasks are allocated that is on First Come First Served – FCFS basis.
- The number of machines and the size of the meta-tasks in the heterogeneous computing environment is known.

Since these assumptions are well satisfied by advance scheduling algorithms wherein load balancing concept is also used to keep all the resources busy.

IV. Scheduling Algorithm

Existing scheduling algorithms or heuristics like minmin, maxmin and sufferage are easy to implement but are not suitable for large scale applications that is applications with large number of tasks, also it increases execution time that is time complexity, these drawbacks are overcome by advance scheduling algorithms that provides the better solution without degrading system performance, making utilization of all the available resources.

1. Minmin+ Scheduling

Minmin+ algorithm works using the advance concepts that is it makes use of queue that contains already calculated MCT for all tasks. Thus it works faster as compare to minmin and also reduces time complexity. It uses necessary methods that are called accordingly for making initializations of variables and for selection of task based on minimum completion time. Since this algorithm begins with initialization of variables and then it selects task that has overall minimum completion time for allocation to available resources. Since, once the task is allocated to resource, that particular task is deleted from queue. And process is repeated until all the tasks are assigned to available resources. This scheduling also maintains the array that provides information about the tasks that are not yet assigned to the resource. It makes use of linear sorting and binary heap. Thus it use various operations to sort elements in queue, to delete required tasks from queue and to maintain the remaining tasks that are not yet assigned to resources. Thus overall time complexity is reduced and processing of system becomes faster even in heterogeneous environment.

2. Maxmin+ Scheduling

Maxmin+ algorithm works similar to minmin+ because it makes use of some methods of minmin+ for initializations of variables but selection of task is done on the basis of existing methods of maxmin scheduling algorithm. Since again this scheduling differs in method of task selection that is this scheduling makes use of makespan that is overall maximum completion time. Based on this makespan allocation of particular task is done to available resource. These 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+.

3. Sufferage+ Scheduling

Sufferage+ scheduling works similar to other heuristics methods wherein, task selection policy is again based on sufferage value but along with some additional methods are used in it. Since, these scheduling also make use of methods of minmin+ scheduling for initializations of variables and selection of task as well as makespan for comparison with sufferage value. Since once initialization and selection of task is done, the sufferage scheduling method is used to calculate sufferage value that is calculated as difference between second and first MCT of tasks and this value is compared with makespan. 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. Once task is assigned it is deleted from queue and all this process is repeated until all the tasks are allocated to respective resource.

4. Hybrid Scheduling

As name indicates hybrid, it means the combination of one or more different heuristics. Since, existing systems contains the combination of minmin and maxmin wherein these scheduling algorithms are called alternatively if required that is depending on task minimum completion time. This also overcomes the drawback of minmin and maxmin that is minmin selects task with overall minimum completion time to be allocated to available resource due to which the tasks with maximum completion time remains unallocated for longer time. Similarly, maxmin selects tasks to be allocated to available resource with maximum of completion time also called as makespan due to which tasks with minimum completion time remains unallocated for longer time. Thus hybrid scheduling uses minmin if task has minimum completion time and maxmin if task has maximum completion time. This also reduces time complexity and produces better solution for the system without degrading its performance [9, 10].

V. Example of Task Allocation

Task allocation is done using the scheduling algorithms between server machine and client machine i.e. when provided server machine with some application it divides given application into number of tasks and allocates these tasks to available client machine depending on minimum completion time of each task also achieving load balance on each client machine. Following figures explain an example of just one task assignment from server to client for identification of whether the given number is even or odd.

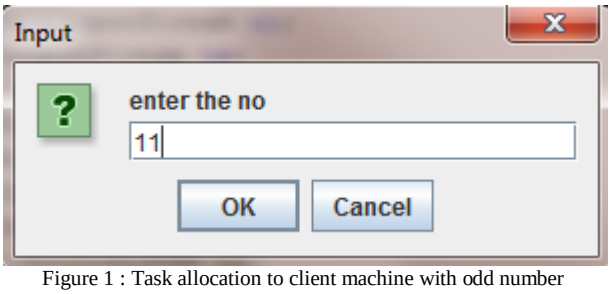

Figure 1 : Task allocation to client machine with odd number

As shown in figure 1, when server and client machines are enabled, server machine identifies IP address of client machine and checks the load on it then server machine allocates the task to client when some input is provided from user. Thus figure 1 indicates that input is provided and the task is ready to be allocated to client machine where it is processed and result is send back to client machine for display and this is shown in figure 2.

The result of number identification is shown in figure 2 that is when task is allocated form server machine to client machine, the given task to it, is processed on client machine and result is return back to server machine.

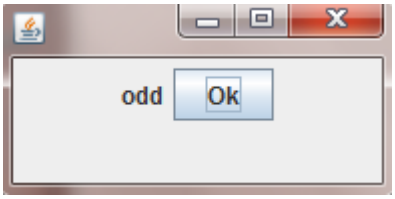

Figure 2 : Available result on server machine

Since the actual code of processing task is written on client machine so, when number 11 is given as a input, this number is identified by client machine as odd and it is return to server machine. The display code is written on server machine so the result is displayed on server machine shown in figure 2.

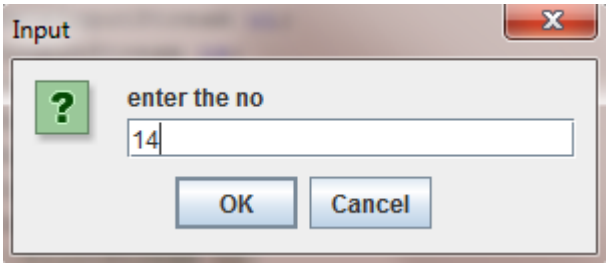

Figure 3 : Task allocation to client machine with even number

Similarly, second time when task is provided to client machine the result is change as second time the input provided by user is different as shown in figure 3.

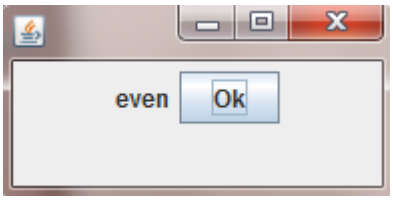

Figure 4 : Available result on server machine

Since result shown in figure 4 produced by server machine is even as the task processed and identified by client machine as even that is send back to server machine.

Actual implementation according to scheduling algorithms will show the results of which task is executed on which machine and at what expected execution time when the task is provided to available resources. Since, above shown example is implemented using Java platform so it is also well suitable for heterogeneous system that is, the systems with different configurations including different operating systems.

VI. Conclusion

The aim of this paper was to present the concept of load balancing and various scheduling and advance scheduling methods like minmin, maxmin, sufferage and minmin+, maxmin+, sufferage+, hybrid respectively. Since advance scheduling methods are best suitable for large scale applications that contain the maximum number of tasks providing the better solution quality without degrading system performance. This scheduling algorithm overcomes the drawback of existing heuristic that produces solution but takes longer time for processing of tasks that are to be allocated to available resources. This system combines the advantages of minmin and maxmin and overcomes the disadvantages of these scheduling algorithms. This study can be further extended by considering machine as well as task heterogeneity.

References

1. 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.
2. Kamali Gupta, Manpreet Singh, “Heuristic Based Task Scheduling In Grid”, International Journal of ngineering and Technology (IJET), vol. 4, pp. 254260, Aug-Sep 2012.

3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. Yagoubi. B, and Slimani. Y, "Task Load Balancing Strategy for Grid Computing", Journal of Computer Science, Vol. 3, No. 3, pp. 186-194, 2007.
12. 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.
13. 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.
14. 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.

