Analysis of an ETF Task Scheduling Algorithm

Navpreet Singh
Software Developer
Peelak, Ajmer

Komal Sharma
M.C.A Student
India

ABSTRACT

Parallel processing is the concept in which the processing of task is done simultaneously. Parallel processing helps in allocating the same task on two or more processors in order to obtain the faster results. The objective of parallel processing is to schedule the tasks onto multiple processors and analyze their performance, select the parallel processor scheduling algorithm on which the performance can be calculated, and then selecting the best algorithm. Directed Acyclic Graph (DAG) is used for this purpose. A DAG is a general model of a parallel program consisting of a set of processes. It is also known as task graph in which each task is expressed by node. A node has one or more inputs and can have one or more output to various nodes. In this study we will try to understand the meaning of parallel processing and one of the important task scheduling algorithm i.e. ETF.

Keywords

Parallel Processing, BNP, Task Scheduling, DAG, Early time First (ETF).

1. INTRODUCTION

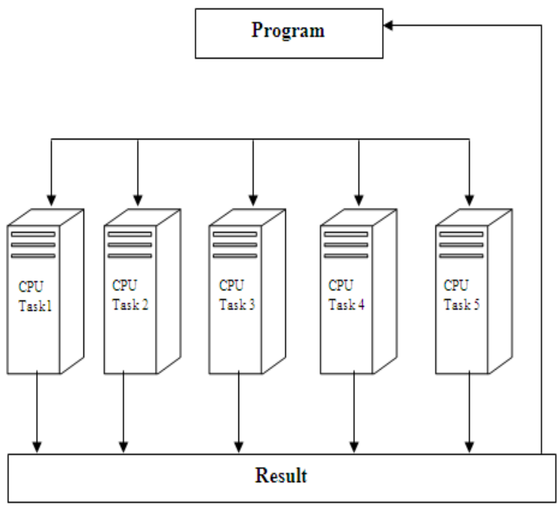
Parallel Processing[1][3] saves time and/or money and provides concurrency that means a single compute resource can only do one thing at a time but multiple computing resources can be doing many things simultaneously. Parallel processing makes a program run faster because there are more processors (CPUs) running it.

In parallel processing multiple operations or tasks can be executed simultaneously on many different processing devices and at the end they are put back together again to get the correct result. So, the larger task completes more quickly.

The computer resources can include a single computer with multiple processors, or a number of computers connected by a network, or a combination of both. Some tasks can be divided and are good candidates for parallel processing. For example, in a bank with one teller, customers must form one line to be served.

With two tellers, the task of waiting on customers can be easily split between the two tellers so customers are served twice as fast. But all tasks can not be divided.

However, parallel processing has a number of problems which are not encountered in sequential processing, for example designing a parallel algorithm for the application, partitioning of the application into tasks, coordinating communication and synchronization, and scheduling of the tasks onto the machine. If these problems are not properly handled, parallelization of an application may not be beneficial. The following diagram shows how parallel system work with number of tasks.



The working of ETF is very similar to Dynamic Level Scheduling algorithm [4][5] uses an attribute called dynamic level (DL), which is the difference between the static b-level of a node and its earliest start-time on a processor. The node-processor pair which gives the largest value of DL is selected for scheduling. This mechanism is similar to the one used by the ETF algorithm. However, there is one subtle difference between the ETF algorithm and the DLS algorithm: the ETF algorithm always schedules the node with the minimum earliest start-time and uses static b-level to break ties. But, the DLS algorithm tends to schedule nodes in a descending order of static b-levels at the beginning of a scheduling process but tends to schedule nodes in an ascending order of t-levels near the end of the scheduling process. The algorithm is briefly described below:

1. Calculate the b-level of each node in the graph.
2. Initially, the ready list includes only the entry nodes.

Repeat

3. Calculate the earliest start-time for every ready node on each processor. Hence, compute the DL of every node-processor pair by subtracting the earliest start time from the node's static b-level.

4. Select the node-processor pair that gives the largest DL. Schedule the node to the corresponding processor.

5. Add the newly ready nodes to the ready list.

Until all nodes are scheduled.

The complexity of the DLS algorithm is O(pv3).

2. OBJECTIVE OF STUDY

The various objective of this study are as shown below:

- To understand the meaning of parallel processing.
- To understand the meaning of task scheduling.
- To understand the working of ETF
- To implement DAG of 5 nodes with ETF.
- To analyze different metrics of ETF

3. ANALYSIS OF ETF

Consider a DAG consisting of 5 task nodes as shown in figure 3.1. The 5 task nodes used are T1, T2, T3, T4 and T5.

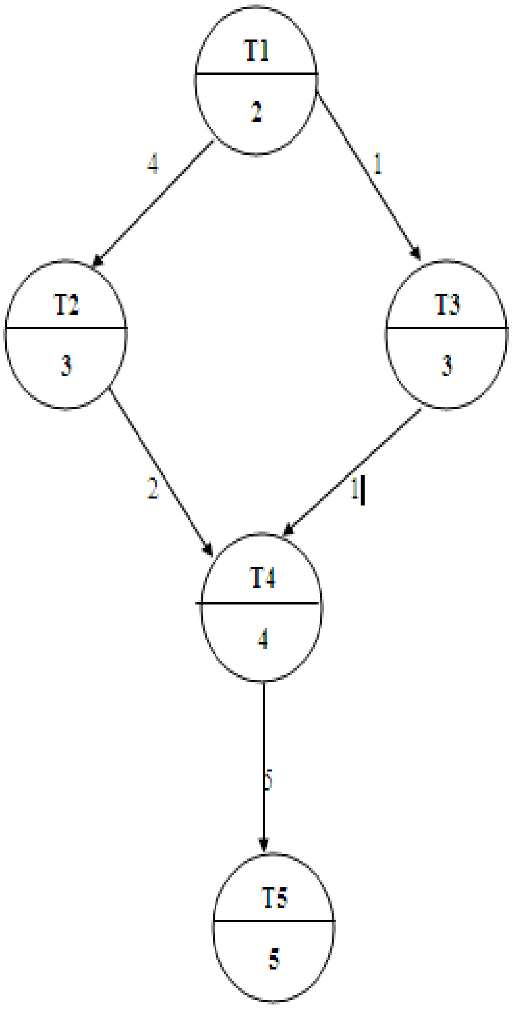


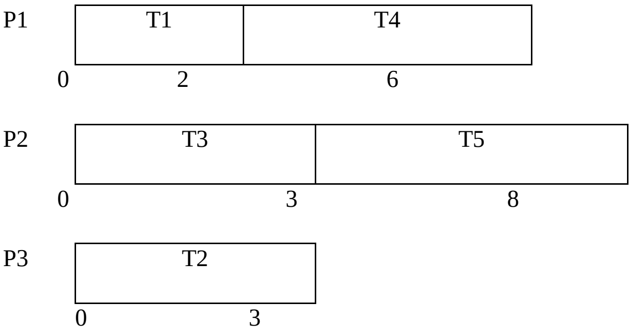
Figure 1: DAG with 5 Nodes

The following table shows the various scheduling attributes used to find the performance of algorithms:

Tasks	Execution Time	Static b-level	t-level	b-level	ALAP Time	Dynamic Level
T1	2	14	0	25	0	14
T2	3	12	6	19	6	6
T3	3	12	3	18	7	9
T4	4	9	11	14	11	-2
T5	5	5	20	5	20	-15

Table1 : Scheduling attributes of 5 task nodes

ETF:- The schedule generated by ETF algorithm is given below:



Makespan = 8 time units

4. CONCLUSION

We have studied one of the parallel task scheduling algorithm for allocating task to different machines or sites. It si found the ETF algorithm shows an improvement over serial task scheduling algorithm. The study shows that the total makespan time when task are scheduled with ETF is 8 time units which otherwise will be 22 time units if scheduled with serial task scheduling algorithms. No doubt ETF reduces the makespan but it uses complex mechanism as compare to serial task scheduling algorithm.

5. REFERENCES

[1] Yu-Kwong Kwok, Ishfaq Ahmad, "Static Scheduling Algorithm for Allocating Directed Task Graph to multiprocessors", ACM Computing Surveys, Vol. 31, no. 4, December 1999.

[2]. Ishfaq Ahmad, Yu-Kwong Kwok, Min-You Wu, "Analysis, Evaluation, and Comparison of Algorithms for Scheduling Task Graphs on Parallel Processors", Proceedings of the 1996 International Symposium on Parallel Architectures, Algorithms and Networks, IEEE Computer Society Washington, DC, USA.

[3]. T. Hagras, J. Janecek, "Static versus Dynamic List-Scheduling Performance Comparison", Acta Polytechnica Vol. 43 No. 6/2003.

[4]. Ishfaq Ahmad, Yu-Kwong Kwok, Min-You Wu, "Performance Comparison of Algorithms for Static Scheduling of DAG to Multiprocessors", ACM Computing Surveys, Vol. 31, no. 4, December 1999.

[5] Sharma, Manik, et al. "Comparative study of static metrics of procedural and object oriented programming languages."International Journal of Computers & Technology 2.1 (2012): 15-19.

[6]. Yu-Kwong Kwok, Ishfaq Ahmad, "Dynamic Critical-Path Scheduling: An Effective Technique for allocating Task Graphs to Multiprocessors", IEEE Transactions on Parallel and Distributed System, Vol. 7, No. 5.

[7]. Manik Sharma, Gurdev Singh, Harsimran Kaur. "A Study Of BNP Parallel Task Scheduling Algorithms Metric's For Distributed Database System". International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.1, January 2012..