

A Novel Approach for Task Scheduling in Multiprocessor System

Ranjit Rajak
Research Scholar,
School of Computer and
System Sciences
Jawaharlal Nehru University
New Delhi-110067,India

ABSTRACT

In multiprocessor system, scheduling of tasks to assigned on the number of processors. The major objective of task scheduling is to find minimum execution time of a program. It is well known that the complexity of a general scheduling problem is NP-Complete [9], there are number of heuristic have been developed. Each of which may either find optimal or near optimal scheduling under the different conditions. The task scheduling is represented by a directed acyclic graph (DAG). In this paper, we present a new scheduling algorithm which is called Task Scheduling based on Breath First Search(TSB). The TSB is queue based approach to schedule parallel tasks on the homogenous parallel multiprocessor system. Its performance is evaluated in comparison with Highest Level First with Estimate Time (HLFET) algorithm, Modified Critical Path (MCP) algorithm, Earliest Time First (ETF) algorithm and Dynamic Level Scheduling (DLS) algorithm in terms of Speedup, Efficiency, Load Balance and Normalized Scheduling Length (NSL).

Keywords

Task scheduling, DAG, NP-complete, parallel processing.

1. INTRODUCTION

The task scheduling in multiprocessor system is also known as multiprocessor scheduling. It has a number of applications from scientific to engineering problems. In parallel processing ,the main concept is to speeding up the execution of a task by dividing into a number of sub-tasks and execute simultaneously. The goal of task scheduling is to minimize the overall execution time of a program i.e. to minimize the scheduling length. The task scheduling in multiprocessor system has been proven to be NP-Complete except for few restricted cases [1].The scheduling problem is NP –complete into simple cases [2]: scheduling tasks with uniform weights to an arbitrary number of processor and scheduling tasks with weights equal to one or two units to two processors. There are three components[3] consisting of task scheduling execution: first the performance of the homogenous processors, second the mapping of the tasks onto the processors and finally sequence of the execution of the tasks on each processor.

In this paper, we present TBS approach to the multiprocessor task scheduling problem. The TBS provides an effective tasks order of execution on the processors. It uses the concepts Breath First Search (BFS) graph traversal algorithm[4] to find order of tasks execution. We studied all the four Bounded Number of Processor (BNP) class of scheduling algorithms: Highest Level First with Estimate Time (HLFET) algorithm, Modified Critical Path (MCP) algorithm, Earliest Time First (ETF) algorithm and Dynamic Level Scheduling (DLS) algorithm and compared with our proposed algorithm. We will discuss our algorithm with the help of an example and

will see that proposed algorithm gives minimum scheduling length. We have taken four matrices: speedup, efficiency ,load balancing and normalized scheduling length (NSL) for performance analysis of algorithms.

The rest of this paper is organized as follows : section 2 defines task scheduling in multiprocessor system model. Section 3 presents the proposed TSB algorithm in detail. Section 4 presents the BNP class of algorithms introduction. The comparison matrices are in the section 5. The performance analysis and results are in section 6 and we finally come to conclusion in section 7.

2. TASK SCHEDULING IN MULTIPROCESSOR SYSTEM MODEL

The multiprocessor scheduling can be classified into deterministic and non deterministic scheduling[8].A scheduling problem is known as deterministic if all the information of the tasks about computation time ,communication time and their precedence constraint are known during program compilation time. It is also called static scheduling problem. In case non-deterministic scheduling is not till program execution time. It is also called dynamic scheduling problem. In this paper, we are considering only static scheduling problem[9]. There are some constraints of static scheduling problem : the number of tasks, execution time of task, inter task communication and number of processors [10].

The multiprocessor system consists of n number of homogenous processors

$$P=\{P_1,P_2,\dots,P_n\}$$

All the processors are fully connected with each other via same links. Figure 1 shows a fully connected four parallel system with same link.

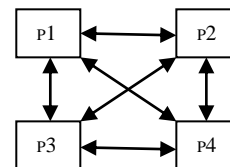


Figure 1. A fully connected parallel processors

In static scheduling a parallel program is represented by directed acyclic graph(DAG). The DAG model consists of two tuples $G=(V,E)$.

Where:

$V=\{T_1,T_2,\dots,T_m\}$ finite set of tasks.

$E=\{e_{ij}\}$ set of edges which connects the two tasks and also represents the precedence constraint among the computational tasks.

Each task T_i has associated with weight W which is called computational time and it is denoted by $W(T_i)$. The

computation time is the amount of time of the task (T_i) required to execute on any one of n identical processors. Similarly, each edge is also associated with weight C which is called communication time between the two tasks but scheduled on the different processors. It is denoted by $C(T_i, T_j)$. The communication time of tasks will be zero they are scheduled to the same processor. If there is a direct edge between two tasks T_i and T_j , then T_i is the predecessor of T_j and T_j is successor of T_i . The one condition always that hold task T_j cannot execute until T_i completes if $i < j$. Considering DAG with six nodes :

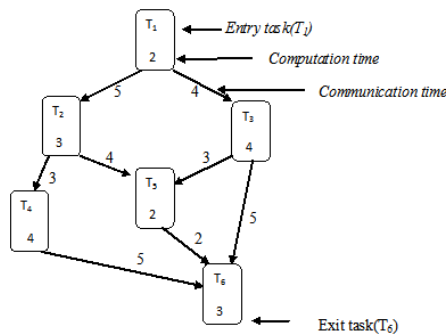


Figure 2. A DAG Model with six nodes

In a given DAG, a task without any predecessors and without any successors is called an entry task and exit task respectively.

3. PROPOSED ALGORITHM

This section presents new task scheduling algorithm using breath first search(TSB). All scheduling processes on the bounded number of processors and they are fully connected homogeneous in nature. The TSB is based on the breath first search (BFS) graph traversal based scheduling algorithm. The TSB consist of two queues: Ready Task Queue (RTQ) and Not Ready Task Queue(NRTQ). Initially, NRTQ contains only entry task of DAG and whenever a task pickup from NRTQ and it checks the precedence constraint is satisfying or not. If satisfied then add to RTQ, otherwise add to NRTQ. These process follows till the last task of the DAG.

Finally , we have a RTQ which contains all the tasks of a given DAG.

Here , we are using RTQ as priorities attributes of the tasks. Schedule the tasks on the processors as per RTQ . Assume that we have DAG and set of available processors.

3.1 The Algorithm

The procedure of the Task Scheduling Based on Breath First Search (TSB) Algorithm is

Step 1: Read DAG(n number of tasks)

Entry task : starting task of a DAG .

Step 2: Add entry task to Not Ready Task Queue(NRTQ)

Step 3: while NRTQ is not empty do

Remove a task from NRTQ

Step 4: If task is not exist task then

Generate successors of task

Step 5: For each successor of task do

If successor task is not present on RTQ and successor task is not present on NRTQ then

Add task on NRTQ // end for

Step 6: if task is not present on RTQ then

Add task on RTQ

Step 7: if task is exist task then

Check task predecessors is present or not on

RTQ

Step 8: if task predecessors are present then

Add task on RTQ

Otherwise

Add task on NRTQ // end while

Step 9: Now we have RTQ that contains all the tasks of DAG .

While RTQ is not empty do

Schedule the task on the RTQ to the processor that allows the earliest execution

Go for next task on RTQ

Step 10: Scheduling the tasks based on RTQ to the processors.

3.2 An illustrative Example TSB Algorithm

We have taken DAG model[8] consisting of nine tasks $T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9$ and their communication time and computation time are also given in the DAG.

Now, we have to find the RTQ based on our proposed algorithm of a given DAG. After that scheduling the RTQ tasks on the processors. Here, also considering that we have a multiprocessor system with four processors and they are identical and fully connected.

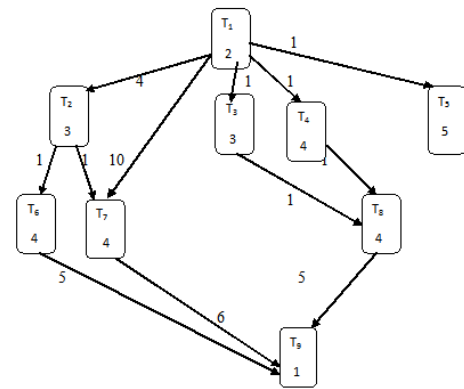


Figure 3. A DAG Model with nine nodes

Now we have to trace Not Ready Task Queue(NRTQ) and Ready Task Queue (RTQ) as per proposed algorithm is given below and remove a task from front of queue and add to rear of queue.

Not Ready Task Queue (NRTQ)	Ready Task Queue (RTQ)
T1	...
T2 T7 T3 T4 T5	T1
T7 T3 T4 T5 T6	T1 T2
T3 T4 T5 T6 T9	T1 T2 T7
T4 T5 T6 T9 T8	T1 T2 T7 T3
T5 T6 T9 T8	T1 T2 T7 T3 T4
T6 T9 T8	T1 T2 T7 T3 T4 T5
T9 T8	T1 T2 T7 T3 T4 T5 T6
T8 T9	T1 T2 T7 T3 T4 T5 T6
T9	T1 T2 T7 T3 T4 T5 T6 T8
...	T1 T2 T7 T3 T4 T5 T6 T8 T9

Figure 4. Processing of NRTQ and RTQ

We have RTQ tasks T1, T2, T7, T3, T4, T5, T6, T8 and T9. Use these tasks to draw the Gantt. Chart for scheduling of the tasks.

TIME	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
P1	T1	T1	T2	T2	T2	T7	T7	T7	T7									
P2				T3	T3	T3	T6	T6	T6	T6								
P3				T4	T4	T4			T8	T8	T8	T8					T9	
P4				T5	T5	T5												

Figure 5. The scheduling generated by the TSB algorithm with scheduling length 16.

4. BNP CLASS OF SCHEDULING ALGORITHMS

This class of scheduling algorithm was developed for bounded number of processors and homogenous. Also, all the processors are fully connected. Here, we have taken four BNP classes of scheduling algorithms: Highest Level First with Estimate Time (HLFET) algorithm, Modified Critical Path (MCP) algorithm, Earliest Time First (ETF) algorithm and Dynamic Level Scheduling (DLS) algorithm. All these four algorithms based on the priority attributes: The t-level (top level), b-level (bottom level) [12], static level(sl) and ALAP (As –Late-As-Possible) start time[13,14].

4.1. HLFET Algorithm

The HLFET algorithm[11] is the simplest among all the BNP class of scheduling algorithms. Its priority is based on static level(sl) attribute[8]. It does not use communication time during the entire process. The scheduling length of HLFET[8] is 19.

4.2.MCP Algorithm

The MCP algorithm [14] is based on the ALAP priority attribute. It includes the communication time in the entire process. The scheduling length of MCP[8] is 20.

4.3.ETF Algorithm :

The ETF algorithm[15] also uses static level priority attribute to compute the priority of the task. The scheduling length of ETF[8] is 19.

4.4.DLS Algorithm :

The DLS algorithm[16] uses dynamic level(dl) as a priority attribute, where dl is the difference between sl and t-level of the every task. The scheduling length of DLS[8] is 19.

5. COMPARISON MATRICES

The main comparison metric is scheduling length of the algorithms. All the performance metrics depends on the scheduling length. We are using four types of matrices: speedup(S_p), efficiency(Eff), normalized scheduling length (NSL) and load balancing(LBal).

5.1 Speedup

Speedup is the ratio between sequential execution time and parallel execution time [5] where the sequential time execution time is sum of total computation time of each task and parallel time execution is the scheduling length on limited number of processors.

$$S_p = \left(\sum_{i=1}^n T_i \right) / T_p$$

where $\sum_{i=1}^n T_i$: sum of computational time of tasks(T_i) in sequential order. $i=1,2,3...n$.

T_p : total parallel execution time or scheduling length of a DAG.

5.2 Efficiency

The efficiency of a parallel program is a measure of processor utilization [5].

$$Eff = S_p / N_p$$

Where S_p : Speedup, N_p : Number of processors

5.3 Normalized Scheduling Length

The Normalized scheduling length (NSL) [6] of a scheduling algorithm is given by

$$NSL = \frac{\text{Scheduling length of particular algorithm}}{\text{Max\{sum of computation costs along a path\}}$$

5.4 Load Balancing

The load balancing [7] is calculated by the ratio of scheduling length to average execution time over all processors.

$$L\ Bal = SL / Avg$$

Where

SL: Scheduling length.

Avg: It is ratio of sum of processing time of each processor and numbers of processors are used.

6. PERFORMANCE ANALYSIS

6.1 TSB Algorithm with scheduling length 16

$$S_p = 30 / 16 = 1.875$$

$$Eff = (1.875 / 4) * 100 = 46.87\%$$

$$NSL = 16 / 23 = 0.695$$

$$L\ Bal=16/((9+10+16+6)/4)=16/10.25=1.560$$

6.2 HLFET Algorithm with scheduling length 19

$$S_p=30/19=1.578, \text{ Eff}=(1.578/4)*100=39.47\%$$

$$NSL=19/23=0.826,$$

$$L\ Bal=19/((13+16+8+11)/4)=19/12=1.583$$

6.3 MCP Algorithm with scheduling length 20

$$S_p=1.50, \text{ Eff}=37.50\%, \text{ NSL}=.869, \text{ L Bal}=1.509$$

6.4 ETF Algorithm with the scheduling length 19

$$S_p=1.578, \text{ Eff}=39.47\%, \text{ NSL}=.826, \text{ L Bal}=1.425$$

6.5 DLS Algorithm with scheduling length 19

$$S_p=1.578, \text{ Eff}=39.47\%, \text{ NSL}=.826, \text{ L Bal}=1.425$$

Table 1. Comparison based on the comparison matrices

S.No	Algorithms	SL	Sp	Eff (%)	L Bal	NSL
1	TSB	16	1.875	46.87	1.560	0.695
2	HLFET	19	1.578	39.47	1.583	0.826
3	MCP	20	1.500	37.50	1.509	0.869
4	ETE	19	1.578	39.47	1.425	0.732
5	DLS	19	1.578	39.47	1.425	0.732

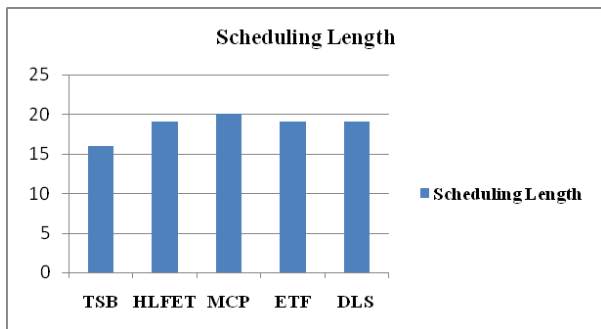


Figure 4. Scheduling Length

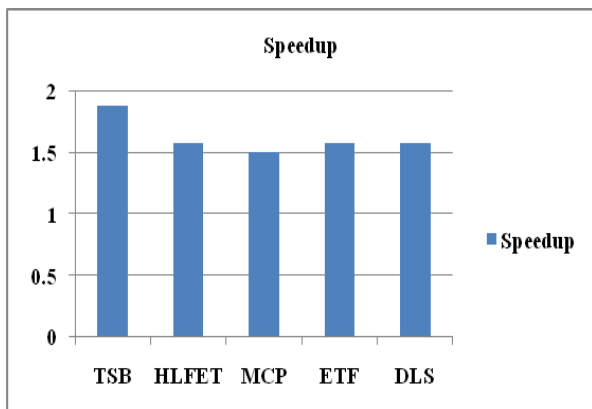


Figure 5 . Speedup

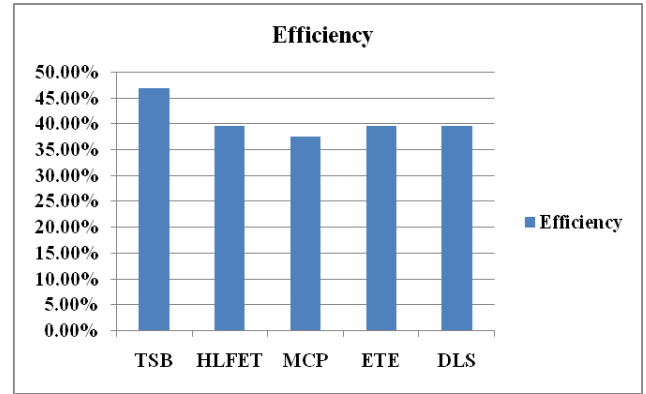


Figure 6. Efficiency

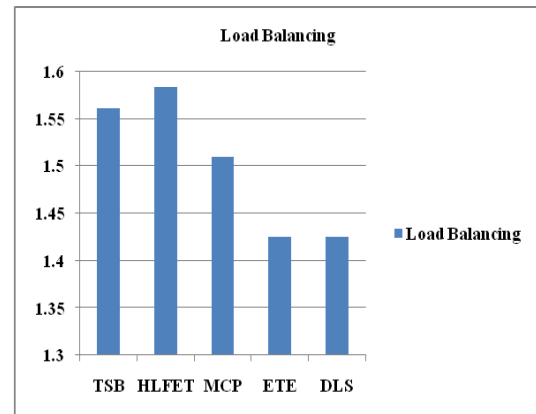


Figure 7. Load Balancing

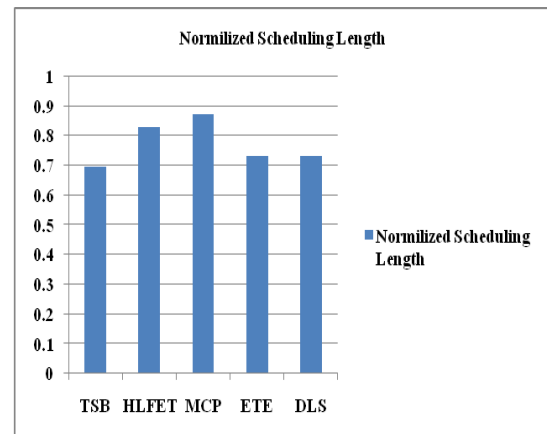


Figure 8. Normalized Scheduling Length

7. CONCLUSION

The TSB algorithm takes less scheduling length among BNP Class of scheduling: HLFET, MCP, ETE and DLS scheduling algorithms. We compared their performance based on speedup, efficiency, load balancing and normalized schedule length. The TSB scheduling are the highest speedup and efficiency than HLFET MCP, ETF, and DLS scheduling algorithm. The load balancing of TSB is good in compare to HLFET scheduling and NSL of TSB is the highest than HLFET MCP, ETF, and DLS scheduling algorithm. The DLS scheduling gives better load balance than HLFET, MCP and

ETF scheduling. All these comparison, we have studied on nine tasks of a DAG model.

8. REFERENCES

- [1] M.R.Gary and D.S.Johnson, "Computer and Intractability: A Guide to the theory of NP Completeness", San Francisco .CA, W.H freeman, 1989.
- [2] Ullman, J,"NP-Complete Scheduling Problem", Journal of Computer System Science, 10,pp384-393,1975.
- [3] Jagbir Singh , " Improved Task Scheduling on Parallel System using Genetic Algorithm" International Journal of Computer Applications (0975 – 8887) Volume 39– No.17, February 2012.
- [4] Yashwant Kanetkar , " Data Structure Using C",BPB Publicatins 2003.
- [5] M.J.Quinn,"Parallel Programming in C with MPI and OpenMP",Tata McGraw-Hill, Edition 2003.
- [6] KwangSik Shin,MyongJin Cha, MunSuck Jang,"Task Scheduling Algorithm using Minimized Duplication in Homogeneous Systems", Journal of Parallel and Distributed Computing ,Vol.68,pp. 1146-1156,2008.
- [7] Fatma A.Omara, Mona M.Arafa,"Genetic Algorithm for Task scheduling Problem",Journal of Parallel and Distributed Computing Vol.70,13-22,2010
- [8] Yu-Kwong Kwok and Ishfaq Ahmad," Static Scheduling Algorithms for Allocating Directed Task Graphs to Multitprocessors", ACM Computing Surveys, Vol.31 N0.4, December 1999.
- [9] H.El-Rewini,T.G.Lewis and H.H.Ali,"Task Scheduling in Parallel and Distributed System" Prentice Hall 1994.
- [10] S.R.Vijaylakshmi and G.Padmavarth," International Journal of Computer Science and Security",Vol.7,pp125-132,2009.
- [11] L.Adam,K.M.Candy and J.Dickson, "A Comparison of list scheduling for parallel processing system", Communication ACM 17, No.12 , pp 685-690,1974.
- [12] Amir Masoud Rahman and Mohammad Ali Vahedi," A Noval Task Scheduling in Multiprocessors System with Genetic Algorithm by Using Elistism Stepping Method", Science and Research Branch,Tehran, Iran,May 26,2008.
- [13] Y-K.Kwok and I.Ahmad, "Dynamic Critical Path Scheduling : An Effective Techniques for Allocating Tasks Graph onto Multiprocessor", IEEE Transaction on Parallel and Distributed System,Vol.7(5) pages 506-621, May 1996.
- [14] M-Y.Wu and D.D.Gajski,"Hyperpool: A programming Aid for Message Passing ", IEEE Transaction on Parallel and Distributed System.Vol 3 ,pp. 330-343,July 1990.
- [15] J.J.Hwang.Y.C.Chow,F.D.Anger and C.Y.Lee, "Scheduling precedence graph in systems with interprocessor communication times",SIAM Journal of Computing . 18 No 2,pp244-257, April 1989.
- [16] G,C.Sih and E.A.Lee," Compile time scheduling heuristic for interconnection –constrained heterogeneous processor architecture ",IEEE Transaction on Parallel and Distributed System 4. No 2,pp-75-87, Feb.1993.