International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 22 July 2023 **Revised:** 10 September 2023 **Accepted:** 26 September 2023*

Load Balancing Factor of DAG Based BNP Scheduling Algorithms

Suresh M. Wadaskar[1*], Sunita Kushwaha[2]

[1*]Research Scholar, MATS School of Information Technology, MATS University, Raipur (C.G.), India
mwsuresh99@gmail.com
[2]Associate Professor, MATS School of Information Technology, MATS University, Raipur (C.G.), India
drsunitak@matsuniversity.ac.in

Abstract:
Effective scheduling of applications is crucial for achieving optimal performance in uniform computing environments. The scheduling problem is known to be NP-complete in both general and specific cases. Given its paramount importance, various BNP algorithms, including HLFET, MCP, ETF, and DLS, have been extensively explored, primarily designed for parallel processing systems. This study evaluates the performance of these four algorithms utilizing a Direct/Arbitrary Task Graph (DAG) comprising 11 tasks, focusing on key performance parameters such as efficiency and load balancing. The MCP algorithm demonstrates superior efficiency, while HLFET excels in terms of load balancing.

Keywords: Static Scheduling, Task Graphs, DAG, Homogeneous processors, Parallel Computing, MCP, HLFET, ETF, DLS.

1. INTRODUCTION

In multiple processor system, task allocation commonly known as scheduling is referred to as multiprocessor scheduling. The principal objective of allocation of task is to reduce the time of execution of a set of task. Task scheduling algorithms can be categorized into following types: Deterministic or Static, and Non-deterministic or dynamic scheduling algorithm. The purpose of scheduling is to reduce the completion time of a parallel application by appropriately assigning tasks to processors [1], [2]. Scheduling of set of dependent task set is done with the help of Directed Acyclic Graph commonly known as (DAG). In Multiprocessor system, Task allocation/ scheduling is a well-known problem as finding an optimal schedule is generally an NP-complete problem, therefore researchers use devise efficient heuristics approach to solve it [1], [2].
Static scheduling, performed at compile time, involves advance knowledge about the task set characteristics (such as processing times, communication, data dependencies, and synchronization requirements) before execution [1]. While, dynamic scheduling allows for making some assumptions about the task set before execution, requiring on-the-fly scheduling decisions. The aim of a dynamic scheduling algorithm extends beyond minimizing completion time; it also aims to minimize scheduling overhead, a substantial cost incurred in running the scheduler [1]. In the DAG model, where nodes symbolize the tasks and directed edges represent execution dependencies and communication amount between tasks.
The next portion of the paper is organized as follow. The next section describes the DAG model, the BNP and taxonomy of DAG scheduling. The section 3 describes the study of BNP scheduling algorithms. Section 4 describes the methodology and experimentation part where section 5 elaborates the result and performance analysis of simulation part. Section 6 concludes the paper work.

2. Taxonomy of DAG based Scheduling Algorithms

Directed Acyclic Graph (DAG)
The Directed Acyclic Graph (DAG) serves as a widely used model in parallel computing environment to represent the dependency between tasks. It comprises four elements denoted as G (V, E, W, C), encompassing the set of tasks/process (V), set of edges/dependency (E), execution time of tasks (W), communication cost, and task priorities (C). Consequently, task scheduling can be translated into scheduling DAG models [1][4].

Bounded Number of Processors (BNP)
Bounded Number of Processors is operates on list scheduling techniques. But list are prepared by the help of Directed Acyclic Graph (DAG) to a predefined number of processors, envisioning them as fully-connected. List scheduling, a category of scheduling heuristics, prioritizes the tasks on the basis of some selected properties and organizes them in a list with descending priority order. In BNP scheduling the priority is calculated as a terms of Static Level (SL), Bottom level (b-level), Top level (t-level) etc., these are some property which is calculated on the basis of DAG. Nodes with higher priority are considered for scheduling ahead of those with lower priority to the available processors [5][6][7].

Taxonomy
Illustrating the varieties of scheduling algorithms, this section deals with the taxonomy of static scheduling (fig: 1). It is essential to emphasize that our taxonomy primarily addresses the static class of scheduling and is consequently not complete as dynamic scheduling is not consider here. At

the peak of the taxonomy, the scheduling problem is bifurcated into two categories, static and dynamic, further static is categorize as heuristic and guided random search.

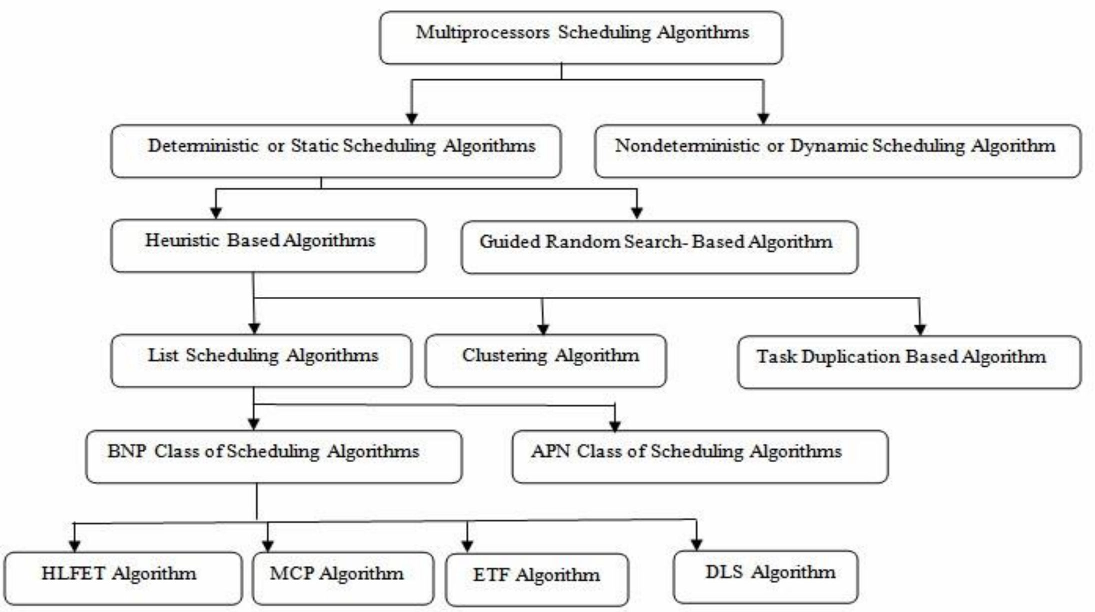

Fig 1: Taxonomy of Multiprocessor Scheduling Algorithms [1].

While our focus is on heuristic scheduling which is classified as, the list scheduling, task duplication based and cluster based scheduling. In which list scheduling categorized as BNP and APN. Further BNP algorithm is use to directly schedule the Directed Acyclic Graph (DAG) to bounded number of processors, envisioning them as fully connected. There are list of BNP based algorithm, some well known algorithms are mentioned in the fig: 1[1-7].

3. RELATED WORK

Author & title	Objective	Name of algorithm	Task model	System mode	Performance metric
Gurjit Kaur, [7]	Study several DAG based algorithms and evaluate their performances by using various performance metrics.	HLFET, ISH, ETF, LAST, MCP, DLS	DAG, Dependent Task set	BNP Multiprocessor system	Processor Utilization, Speedup, Makespan, SLR, NSL, Average Run Time
Ranjit rajak, [2]	Study the some static scheduling algorithms and find out the scheduling length of each algorithm. Finally compared these four algorithms based on four performance metrics.	-HLFET -MCP -DLS -ETF	DAG, Dependent task	BNP class Homogeneous computing system	Speedup, efficiency, NLS, load balancing
Liu yuan, pingui jia and yiping yang, [4]	To get better efficiency and performance of multi-core processor To reduce the quantity of processor and schedule length - new scheduling algorithm is proposed.	Proposed algorithm combined the cluster base concept with interval insertion.	DAG based task model - Random task model shows general performance	BNP Multicore Processor Based Parallel System (Homogeneous)	NSL Speedup rate
L. Qin, F. Ouyang and g. Xiong, [5]	To get better performance in distributed system with dependent tasks. To update the existing table scheduling algorithms.	Combine the Table Scheduling with the concept of Replication of task and propose new Heuristic algorithm HCPTD	DAG with dependent task set	Homogeneous, Heterogeneous, Distributed system	-SLR, Speedup

Popa, E., Iacono, M. and Pop, F. Adapting, [8]	Case study Using two BNP class modified algorithms MCP and HLFET	-Modified MCP, modified HLFET -The simulation tested is based on MTS2 (Many Task Scheduling Simulator)	Independ ent Task set	Heterogeneous System	Scheduling time
I. Ahmad, Yu-Kwong Kwok and Min-You Wu, [3]	Compared some popular algorithms from the different classes of list scheduling BNP, UNC, TDB and APN. And compared these algorithms on the basis of Normalized Schedule Length (NSL).	PY,LWB, DSH, BTDH, LCTD, CPFD, HLFET, ISH, MCP, ETF,DLS, LAST, EZ, LC, DSC,MD, DCP, MH, DLS, BU, AND BAS.	Depende nt task set	Homogene ous	-Five different values of CCR were selected. Normalized Schedule Length (NSL) .
Samriti, Sandeep Gill, Ankur Bharadwaj, [9]	Study and compare the HLFET and MCP algorithm for parallel environment.	-HLFET -MCP	DAG, Depende nt Task set	Homogene ous system	Makespan, Speedup, SLR, processor utilization, complexity
Sharma, A., Kaur, H., [10]	Combined the fuzzy logic concept to the selected algorithms and verify the effect on the performance matrix in 3 case scenario	HLFET, MCP, DSL, ETF	Depende nt task set	Homogene ous System	Makespan, processor utiliztion, speedup
Arora Nidhi, Navneet Singh and Parneet Kaur, [11]	Selection of best performing algorithm from all 4 BNP scheduling algorithm (HLFET, MCP, DLS, ETF). Allocate parallel program represented by DAG based on homogenous processor	-HLFET, MCP, DLS, ETF, DYNAMIC	Depende nt Task set	Homogene ous System	Makespan, Speedup, SLR, processor utilization

4. METHODOLOGY AND RESULT ANALYSIS

In this section, the DAG based allocation/scheduling algorithms is compared with their performance results. Performance measured in term of efficiency and load balancing. Mainly HLFET, DLS, ETF and MCP algorithms are scripted in the MATLAB based tool TORSCHE and simulate the all algorithm for the mention parameters and performance has been calculated [8]. Figure-2 shows methodology and the Figure-3 represents the DAG model for 11 tasks, and the Table-1 represents Task matrix of 11 tasks DAG model.

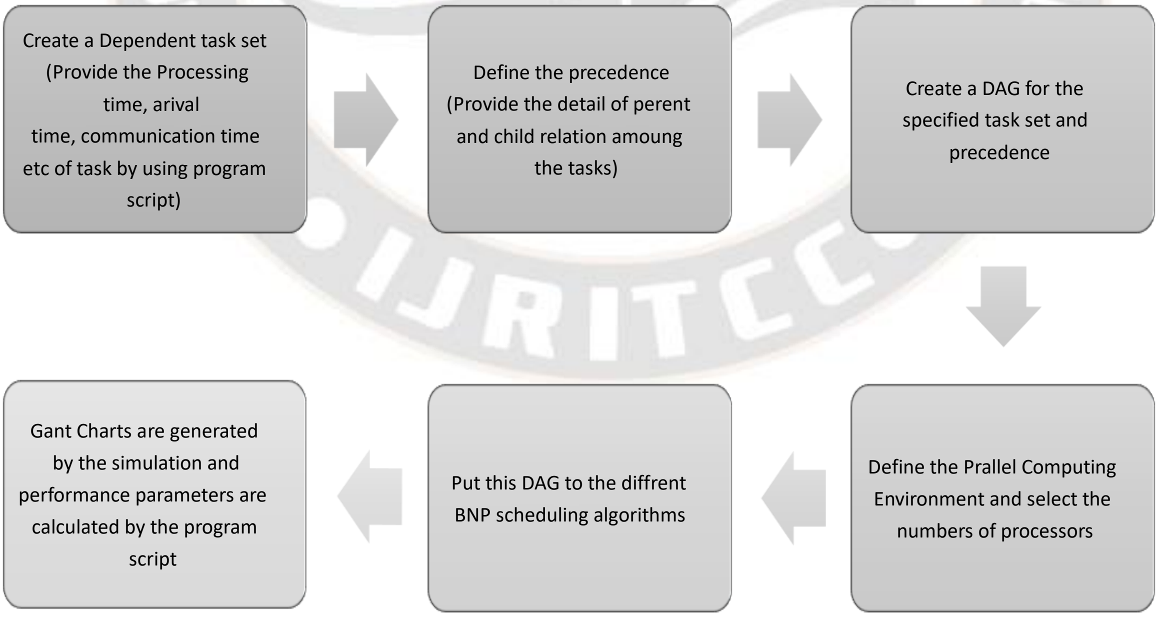

Fig 2: Methodology used in the simulation of the Scheduling Algorithms.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 22 July 2023 **Revised:** 10 September 2023 **Accepted:** 26 September 2023*

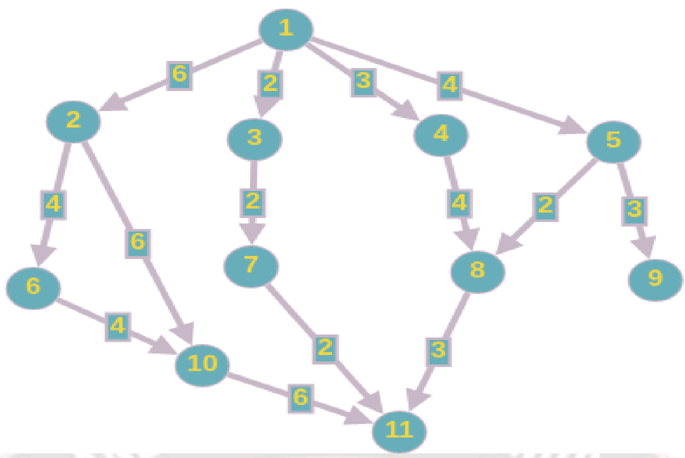

Fig 3: DAG Model of 11 tasks.

Table-1 : Task Matrix Of 11 Task Dag Model

Computation Time	Tasks	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	
2	T1	0	6	2	3	4	0	0	0	0	0	0	Critical Path = 34
4	T2	0	0	0	0	0	4	0	0	0	6	0	
4	T3	0	0	0	0	0	0	2	0	0	0	0	
5	T4	0	0	0	0	0	0	0	4	0	0	0	
4	T5	0	0	0	0	0	0	0	2	3	0	0	
3	T6	0	0	0	0	0	0	0	0	0	4	0	CCR= 0.095865
2	T7	0	0	0	0	0	0	0	0	0	0	2	
5	T8	0	0	0	0	0	0	0	0	0	0	3	
4	T9	0	0	0	0	0	0	0	0	0	0	0	
3	T10	0	0	0	0	0	0	0	0	0	0	6	
2	T11	0	0	0	0	0	0	0	0	0	0	0	

Computation time is mention in the first column in table-1 for each task, second column shows the task number and rest of the columns show their dependency and communication time between the respective tasks. Last column shows the Critical path value and the communication to computation ratio (CCR).

5. PERFORMANCE PARAMETERS

Efficiency: Performance of the parallel system must be check against the efficiency, for better performance efficiency must be high. Efficiency is define as a ratio between Speedup and number of processors [11][12].

Load balancing: If the load is equally divided among the processor then the performance is increased, here load is considered as an amount of processing time. Due to dependency and other factor equal distribution of load is not feasible in parallel computing. Load the proportion of makespan and the average processing time of all processes over all processors [11][12].

Experimental Setup:
The simulation has been performed for the evaluation of the performance of the above given DAG based BNP scheduling algorithms. Experimental parameters have been set as processing time is between 1 to 10 time units. Release time (arrival time) is in the range of 1 to 10 time units, processors are 4 in the numbers and the tasks/ processes numbers are 11.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 22 July 2023 **Revised:** 10 September 2023 **Accepted:** 26 September 2023*

Table2: Schedule for HLFET algorithm for Fig3.

Step	Start Time	Task No	Execution Time	Finish Time	Processor No
1	0	1	2	2	1
2	2	2	4	6	1
3	5	4	5	10	2
4	6	5	4	10	1
5	4	3	4	8	3
6	10	6	3	13	1
7	12	8	5	17	2
8	13	10	3	16	1
9	8	7	2	10	3
10	13	9	4	17	3
11	20	11	2	***22***	1

Fig

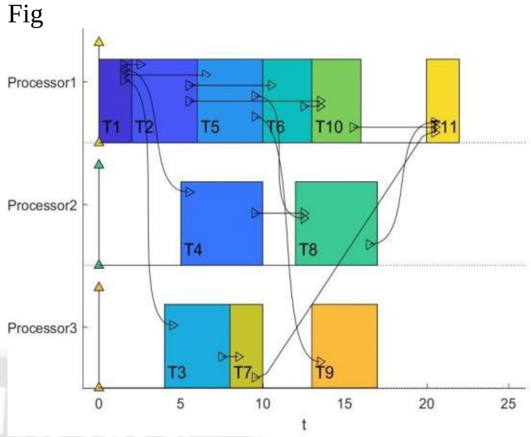

Fig 4: Gantt Chart of HLFET algorithm for Fig3.

Table3: Schedule for MCP algorithm for Fig3.

Step	Start Time	Task No	Execution Time	Finish Time	Processor No
1	0	1	2	2	1
2	2	2	4	6	1
3	5	4	5	10	2
4	6	6	3	9	1
5	6	5	4	10	3
6	4	3	4	8	4
7	9	10	3	12	1
8	12	8	5	17	2
9	8	7	2	10	4
10	10	9	4	14	3
11	18	11	2	***20***	2

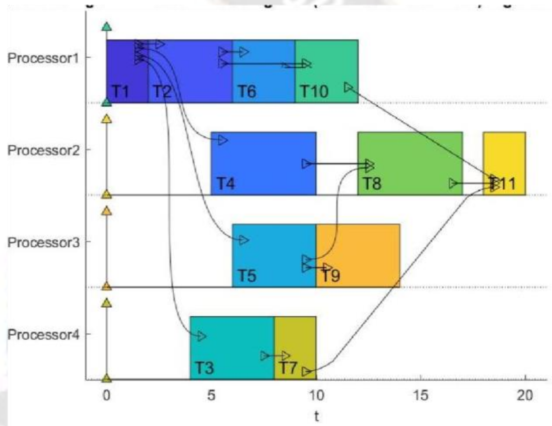

Fig 5: Gantt Chart of MCP algorithm for Fig3.

Table4: Schedule for ETF algorithm for Fig3.

Step	Start Time	Task No	Execution Time	Finish Time	Processor No
1	0	1	2	2	1
2	2	3	4	6	1
3	5	4	5	10	2
4	6	5	4	10	1
5	8	2	4	12	3
6	8	7	2	10	4
7	10	9	4	14	1
8	12	8	5	17	2
9	12	6	3	15	3
10	15	10	3	18	3
11	20	11	2	***22***	3

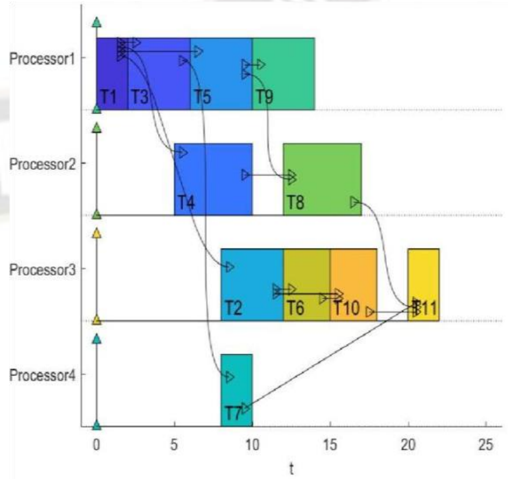

Fig 6: Gantt Chart of ETF algorithm for Fig3.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 22 July 2023 **Revised:** 10 September 2023 **Accepted:** 26 September 2023*

Table5: Schedule for DLS algorithm for Fig3.

Step	Start Time	Task No	Execution Time	Finish Time	Processor No
1	0	1	2	2	1
2	2	4	5	7	1
3	6	5	4	10	2
4	7	2	4	11	1
5	4	3	4	8	3
6	8	7	2	10	3
7	11	8	5	16	2
8	11	6	3	14	1
9	13	9	4	17	3
10	14	10	3	17	1
11	19	11	2	***21***	1

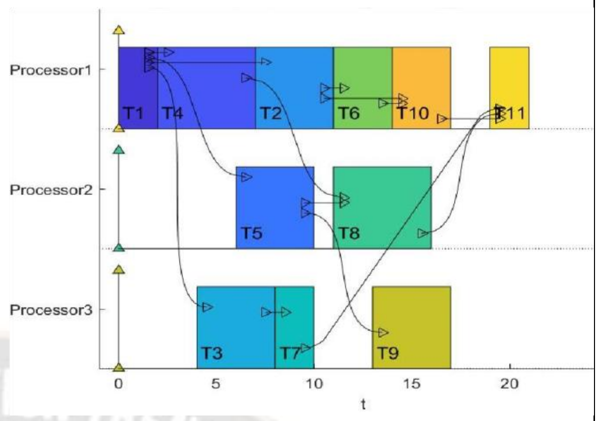

Fig 7: Gantt Chart of DLS algorithm for Fig3.

Table6: Priority Attributes of 11 Task DAG Model (Fig:3)

Task No.	Burst Time	Top Level	Bottom Level	Static Level	ALAP Time	Dynamic Level
1	2	0	34	14	0	14
2	4	8	26	12	8	4
3	4	4	12	8	22	4
4	5	5	19	12	15	7
5	4	6	16	11	18	5
6	3	16	18	8	16	-8
7	2	10	6	4	28	-6
8	5	14	10	7	24	-7
9	4	13	4	4	30	-9
10	3	23	11	5	23	-18
11	2	32	2	2	32	-30

Table7: performance metrics for all five algorithm

Algorithm	Efficiency	Load Balance
HLFET	43.18	1.57
MCP	47.50	1.43
ETF	43.18	1.40
DLS	45.24	1.56

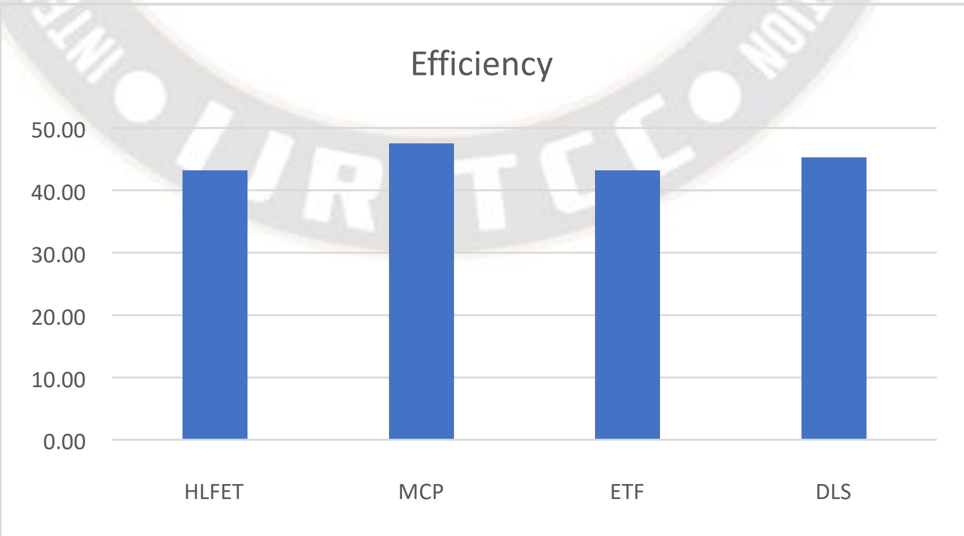

Fig 8: Efficiency for Scheduling Algorithm

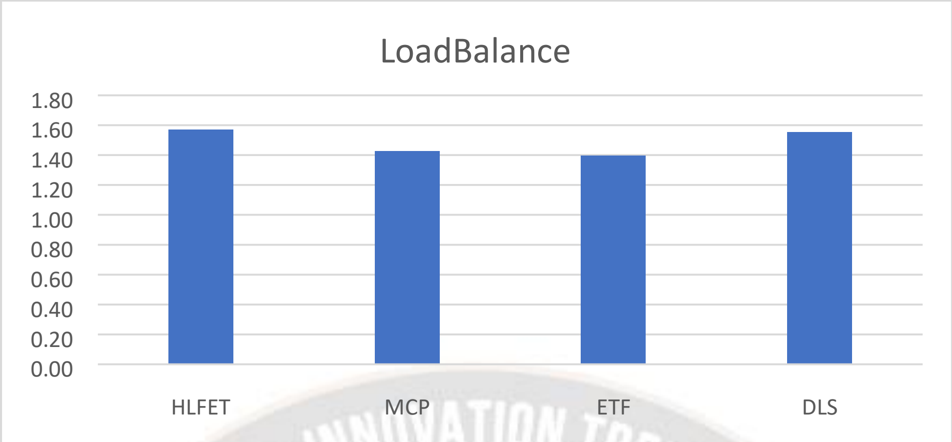

Fig 9: Load balance for Scheduling Algorithm

Table 2 shows the priority sequence for algorithm HLEFT, after that tasks are assign to the processors according to the availability of processor and fig:4 represent the Gantt Chart according to the scheduling algorithm HLEFT.
Similarly table 3, table 4 and table 5 are represent the priority sequence of same task set (which is shown fig:3 DAG model of 11 task) for the scheduling algorithm MCP, ETF and DLS, and assign to the available processor accordingly. Fig:4, fig:5 and fig:6 are present the Gantt charts for same.
Performance of above mention algorithms are compared with the help of graph which is shown on fig:8 and fig:9. Table-6 shows all the Priority Attributes of 11 Task DAG Model calculated by a script written in the MATLAB.
According to this the MCP algorithm performs better as compare to other three algorithms in terms of the efficiency. In term of load balance, the HLFET algorithm performs better as compare to other selected BNP scheduling algorithms.

Observation:

Allocation of the root node is a crucial step. Once the root task is assigned to a processor, execution is initiated; during this time all the remaining processors remain idle until the entry task completes its execution. During the completion of this task on the chosen processor, all successors of this task queue up for scheduling. The execution of the predecessor task is finished on the same processor (e.g., P1), if more than one child (successors) ready in the list for the execution of same task, then they require communication time if scheduled on a processor other than the one used in the execution of the entry task (parent task). Consequently, when more children are exist for the same task than only one successor task must avoid the communication time by scheduling on the same processor, and all other tasks require communication time for execution because the data must be need by the successor task is not available on that processor.

6. CONCLUSION

Nowadays, parallel computing is strongly used in computational applications. This approach allows for the completion of more computations in less time through effective workload distribution among processors. The distribution of tasks is facilitated by scheduling, a challenging aspect of computing systems. Within the realm of scheduling, the BNP class stands out, where a bounded number of processors collaborate to execute assigned tasks that often require proper communication. In this study various BNP scheduling algorithm namely, HLFET, DLS, MCP and ETF are selected for the simulation of the dependent task set for bounded number of processors, for the performance metrics Efficiency and Load balancing. The performance of MCP algorithm in term of efficiency is performing better than other selected scheduling algorithms. In term of load balancing the HLFET share better load as compare to others and DLS algorithm is next to performs better than the other selected algorithms.

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] Ranjit Rajak, COMPARISON OF BOUNDED NUMBER OF PROCESSORS (BNP) CLASS OF SCHEDULING ALGORITHMS BASED ON MATRICES, GESJ: Computer Science and Telecommunications 2012, volume .2, issue 34.
[3] I. Ahmad, Yu-Kwong Kwok and Min-You Wu, “Analysis, evaluation, and comparison of algorithms for scheduling task graphs on parallel processors”, Proceedings Second International Symposium on Parallel Architectures, Algorithms, and Networks, Beijing, China, 1996, pp. 207-213.
[4] Liu Yuan, Pingui Jia and Yiping Yang, “Efficient scheduling of DAG tasks on multi-core processor based parallel systems”, TENCON 2015 - 2015 IEEE Region 10 Conference, Macao, 2015, pp. 1-6.

[5] L. Qin, F. Ouyang and G. Xiong, “Dependent task scheduling algorithm in distributed system”, 2018 4th International Conference on Computer and Technology Applications (ICCTA), Istanbul, 2018, pp. 91-95.
[6] J. Ullman, “NP-Complete Scheduling Problems,” Journal of Computer and System Sciences, Vol 10, 1975, pp. 384-393.
[7] Gurjit Kaur, A DAG based Task Scheduling Algorithms for Multiprocessor System - A Survey, International Journal of Grid and Distributed Computing Vol. 9, No. 9 (2016), pp.103-114.
[8] Popa, E., Iacono, M., Pop, F., “Adapting MCP and HLFET Algorithms to Multiple Simultaneous Scheduling”, Int J Parallel Prog vol 46, pp. 607–629 (2018).
[9] Samriti, Sandeep Gill, Ankur Bharadwaj, “Analysis of HLFET and MCP Task Scheduling Algorithms”, International Journal of Modern Engineering Research (IJMER), Vol.2, Issue.3, May-June 2012 pp-1176-1180.
[10] Sharma, A., Kaur, H. (2013). QOS, Comparison of BNP Scheduling Algorithms with Expanded Fuzzy System. International Journal of Advanced Computer Science and Applications, 4(5).
[11] Arora, Nidhi, Navneet Singh and Parneet Kaur, “Performance Comparison of BNP Scheduling Algorithms In Homogeneous Environment.”, Global journal of computer science and technology, vol 12 (2012): pp 1-9
[12] H. Topcuoglu, S. Hariri and Min-You Wu, “Performance-effective and low- complexity task scheduling for heterogeneous computing”, IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 3, March 2002, pp. 260-274.