

Performance Analysis of List Scheduling on Homogeneous Multiprocessors System

Sunita Kushwaha
Research Scholar
Dept. of C.S. & IT
Pt. Ravishankar Shukla university,
Raipur, India

Sanjay Kumar
Associate Professor
Dept. of C.S. & IT
Pt. Ravishankar Shukla university,
Raipur, India

ABSTRACT

Nowadays multiprocessor system has received a lot of attention because of its efficiency and reliability, which play an important role in success of multiprocessor systems. Hence, there is a necessity for developing efficient scheduling algorithms. Therefore, various researches have been done in different areas of scheduling of multiprocessor system. This paper evaluates the performance of homogeneous multiprocessor system on the basis of Throughput.

General Terms

List heuristic scheduling

Keywords

Throughput, List Scheduling, Static Scheduling

1. INTRODUCTION

Scheduling is simply allocating a set of tasks or jobs to resources such that the optimum performance is obtained [1]. It is often difficult to schedule a program set to separate CPUs or processors in order to maximize system utility. Scheduling is a method by which processes or threads access the resources of the system that they will require in any particular manner to get optimal solution. When more than one processor work together scheduling becomes a challenging task.

1.1. Classification of Scheduling

Scheduling in multiple processor environment may be broadly classified in two categories. First is local scheduling in which all processors maintain their own individual queue. Second is global scheduling in which a central queue is maintained for all processors [2].

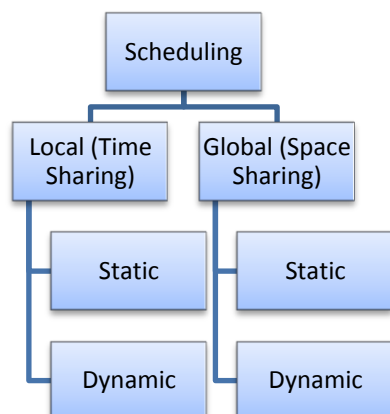


Fig1. Board classification of scheduling methods

Generally, global scheduling policy uses space sharing mechanism where a single job is assigned among several processors. On the other hand local scheduling uses time sharing mechanism where several tasks assigned to a single processor. Further, both global and local scheduling policies are again classified as static and dynamic scheduling. In static scheduling the assignment of tasks to processors is done before program execution begins, while in dynamic scheduling, the redistribution of tasks to processors is done during execution time [3-4]. Static scheduling is also known as deterministic task scheduling and compile time scheduling algorithm. In static scheduling, the following information of tasks is known in advance: communication time between tasks, computation time of tasks, and the precedence constraints of tasks. Dynamic task scheduling is also known as nondeterministic scheduling and run time scheduling algorithms. In dynamic scheduling algorithm, all the information of tasks are known at run time and not in advance.

1.2 Component of scheduling problem

A Scheduling problem consists of three main components.

- 1) Processes
- 2) Processors
- 3) Policy (Scheduling Approach)

This relationship [4] between the scheduler, policies, processors and processes is shown in Figure 2.

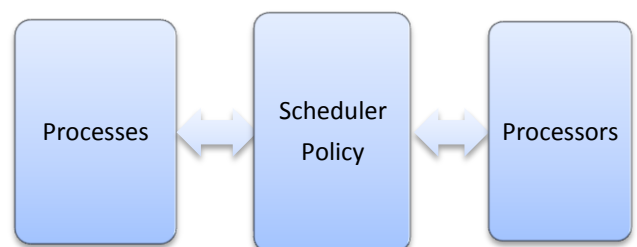


Fig 2. Component of scheduling system

In this paper, processes are independent to each other. Both uniprocessor and multiprocessor systems are considered. Static scheduling algorithms are further classified as heuristic algorithms that give near-optimal solution. It has satisfactory performance and has polynomial time complexity. Heuristic algorithm is classified in three categories namely List task scheduling algorithms, Clustering algorithms and Task duplication algorithms.

There are so many heuristic scheduling approaches existing but for getting optimal schedule with less complexity, list heuristic scheduling is used which is the simplest and efficient approach. This paper evaluate some static list scheduling algorithms namely Longest Processing Time (LPT) first, Shortest Processing Time (SPT) first, Earliest Completion Time (ECT) first and Earliest Starting Time (EST) first [5].

2. PROBLEM DESCRIPTION

Static process scheduling for a homogeneous multiprocessor system is defined as the problem of allocating the processes to the processors that have same processing capabilities, and specifying the start execution time of each task. Efficient task schedule is the one that minimizes the total completion time, or the schedule length of the application. Static task scheduling takes place during the compilation time hence, the characteristics of a processes, such as execution times, processing time etc. of processes, are known in advance [1, 6, 7]. In multiprocessor system scheduling became more challenging than uniprocessor system due to presence of more processing unit. Mapping of processes on processor is difficult.

3. LITERATURE SURVEY

A Hybrid Flow Shop (HFS) [8-9] consists of series of production stages, each of which has several machines operating in parallel. Some stages may have only one machine, but at least one stage must have multiple machines. Eric Angel, Evripidis Bampis and Fanny Pascual studied LPT (Longest processing time), SPT (shortest processing time) and DSPT (delay SPT) algorithms for the problem of scheduling tasks on parallel identical machines in order to minimize the makespan [10]. Hamid, described the CPOP (Critical Path On a Processor), HEFT (Heterogeneous Earliest Finish Time), HCPT (Heterogeneous Critical Parent Trees), HPS (High Performance task Scheduling), PETS (Performance Effective Task Scheduling) and lookahead list-based scheduling heuristic algorithms, for scheduling tasks on heterogeneous processors and proposed a new scheduling algorithm PEFT (Predict Earliest Finish Time) [11]. HLFET (Highest Level First with Estimate Times), ISH (Insertion Scheduling Heuristic), MCP (Modified Critical Path), ETF (Earliest Time First), DLS (Dynamic Level Scheduling) and CNPT (Critical Node Parent Tree) list scheduling for homogeneous environments studied by Nidhi and Anurag [12]. S. Guirchoun, P. Martineau and J.C. Billaut studied a computer system with a server and two parallel processors using SPT/FAM (shortest processing time first/First available machine) scheduling algorithm [13].

4. EXPERIMENTAL SETUP AND PERFORMANCE METRICS

4.1 Experimental setup

No of processors- 1, 2, 3, 4, 5

No of processes- 2, 3, 4, 5, 6

Scheduling algorithms- LPT, SPT, EST, ECT

Range of time instant for arrival of each process- [1- 5]

Range of processing time for each process- [1-10]

4.2 Performance Metrics

Performance metrics are used to evaluate the performance of scheduling algorithm. There are various performance metrics exists such as speedup, efficiency, makespan, throughput etc. However this paper deal only one metric- Throughput.

Throughput: Throughput is the number of processes per time unit that the system completes. This rate reflects the computing power of system [5].

5. RESULT AND DISCUSSION:

This paper evaluates the performance of some static list scheduling algorithms for 'zero' arrival time that means all the processes arrived before a time instant.

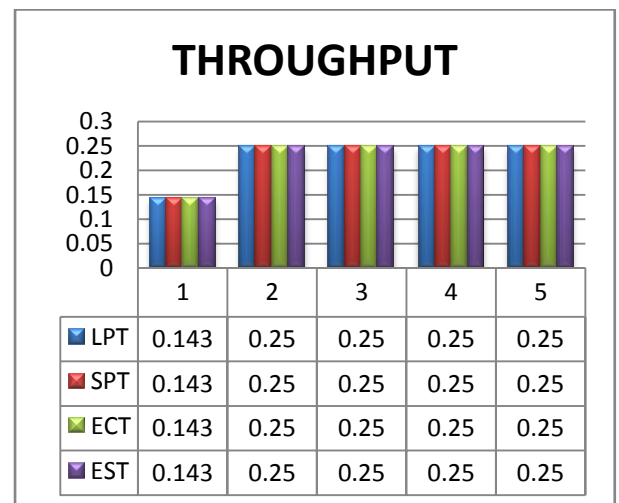


Fig 3. Throughput for 2 processes

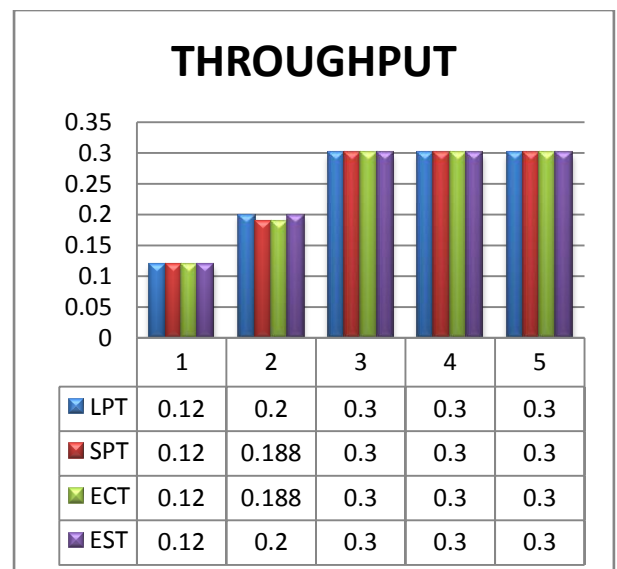


Fig 4. Throughput for 3 processes

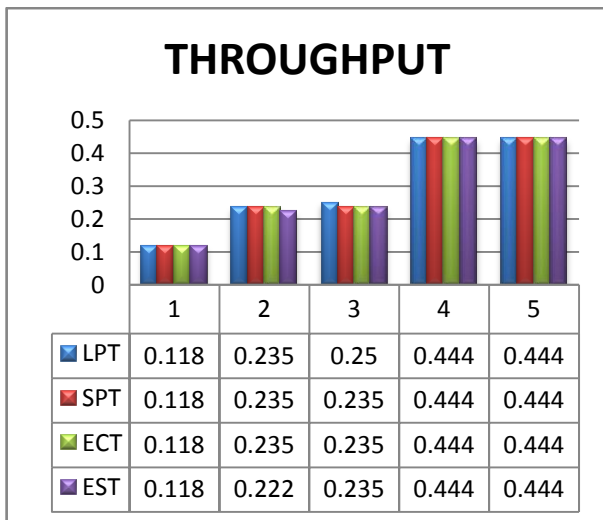


Fig 5. Throughput for 4 processes

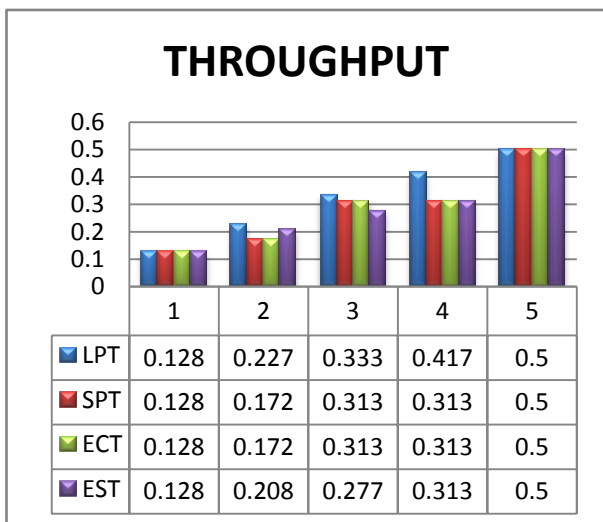


Fig 6. Throughput for 5 processes

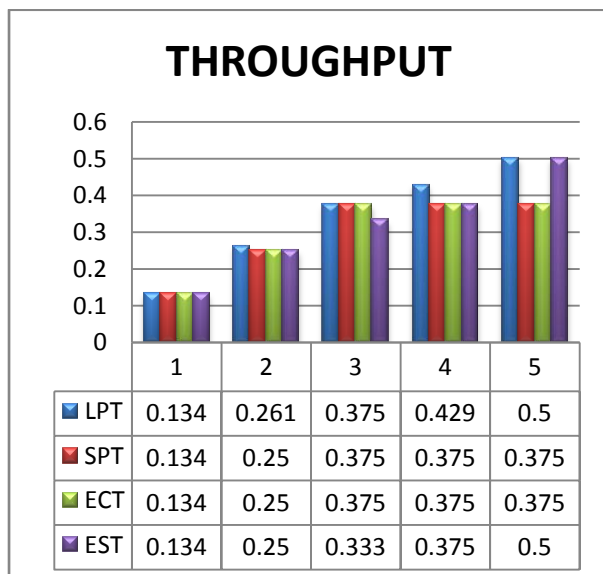


Fig 7. Throughput for 6 processes

- The throughput of multiprocessor system increases only when number of processes quite greater than number processes.
- When number of processes is less than the number of processors throughput remain same for all scheduling algorithms.

6. CONCLUSION

This paper evaluates the performance of some static list scheduling algorithms namely LPT, SPT, ECT and EST for 'zero' arrival time. In this experiment number of processes increases, keeping the number of processors fixed and observes that the throughput of LPT is better than other list scheduling algorithms on the other hand sometime throughput of EST is also better than SPT and ECT.

7. REFERENCES

- [1] Albert Y. Zomaya, Chris Ward, (1999). "Genetic scheduling for parallel processor systems: comparative studies and performance issues", IEEE Transaction on Parallel and Distributed System, vol. 10, pp 795-812.
- [2] T. Hagias and J. Janacek, (2003). "Static vs. Dynamic List-Scheduling Performance Comparison", ActaPolytechnica, Vol.3 No. 6/2003, pp 16-21.
- [3] Chapin, Steven J. and Weismann Jon B, (2002). "Distributed and Multiprocessor Scheduling", *Electrical Engineering and Computer Science, Head book*, Paper 40.
- [4] Thomas casavant, Jong.kuhl, (1988). "A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems", IEEE Trans. on Software Engineering, vol. 14, no. 2, pp 141-154.
- [5] Sunita Kushwaha, Sanjay Kumar, (2014). "Analysis of List Scheduling Algorithms for Parallel System", IEEE International conference on High Performance Computing and Applications (ICHPCA), 22-24 dec., Bhubaneswar, India.
- [6] H. Topcuoglu, S. Hariri, and M.Y. Wu, (2002). "Performance-Effective and Low- Complexity Task Scheduling for Heterogeneous Computing", IEEE Trans. Parallel and Distributed Systems, Vol. 13, No. 3, pp. 260-274.
- [7] Y.K. Kwok and I. Ahmad, (1999). "Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors", ACM Computing Surveys, Vol. 31, No. 4, pp. 406-471.
- [8] Richard Linn and Wei Zhang (1999), "HYBRID FLOW SHOP SCHEDULING: A SURVEY", Computers & Industrial Engineering 37, pp 57-61.
- [9] S. Guirchoun, P. Martineau, J.-C. Billaut (2005), "Total completion time minimization in a computer system with a server and two parallel processors", Computers & Operations Research 32, Elsevier, pp 599-611.
- [10] Eric Angel, Evripidis Bampis, Fanny Pascual (2009), "Truthful algorithms for scheduling selfish tasks on parallel machines", Theoretical Computer Science, Elsevier (press article), pp 1-12.

- [11] Hamid Aranejad and Jorge G. Barbosa (2014), “List Scheduling Algorithm for Heterogeneous Systems by an Optimistic Cost Table”, *IEEE tran. On Parallel and Distributed systems*, Issue No.03, vol.25, pp: 682-694.
- [12] Nidhi Rajak and Anurag Dixit and Ranjit Rajak (2014)” Classification of List Task Scheduling Algorithms: A Short Review Paper”, *Journal of Industrial and Intelligent Information* Vol. 2, No. 4, December, pp320-323.
- [13] Total completion time minimization in a computer system with a server and two parallel processors S. Guirchoun, P. Martineau, J.-C. Billaut, *elsevier Computers & Operations Research* 32 (2005) 599–611 .