ISSN PRINT 2319 1775 Online 2320 7876

 Journal Volume 11, Iss 12, 2022

REVIEW ON DAG BASED SCHEDULING ALGORITHMS IN CONTEXT OF HOMOGENEOUS AND HETEROGENEOUS PROCESSOR

Sunita Kushwaha[1]

[1]Associate Professor, Mats School of Information Technology, Mats University, Raipur (C.G.), India Drsunitak@Matsuniversity.Ac.In

Suresh M. Wadaskar[2]*

[2]*Research Scholar, Mats School of Information Technology, Mats University, Raipur (C.G.), India mwsuresh99@gmail.com

***Corresponding Author:** Suresh M. Wadaskar

*Research Scholar, Mats School of Information Technology, Mats University, Raipur (C.G.), India mwsuresh99@gmail.com

Abstract

Parallel computing emerges as a promising strategy to address the computational demands of numerous present and evolving applications. It involves the simultaneous execution of multiple tasks across multiple processors, with the primary aim of enhancing computational speed. Efficient task scheduling and mapping pose significant challenges in homogeneous and heterogeneous parallel computing environments. This paper explores various algorithm classes tailored for homogeneous and heterogeneous processors and delves into the roles and significance of performance parameters.

Keyword: DAG, DVS, Slack, Backfilling, MAKESPAN, Load balance, Speedup.

I. INTRODUCTION

Parallel computing involves the simultaneous execution of multiple tasks across multiple processors, with the primary objective being the acceleration of computation speed. Effective task scheduling and mapping pose significant challenges in homogeneous and heterogeneous parallel computing environments. Directed Acyclic Graph (DAG) task model find extensive application across various domains for representing dependencies, particularly within dependent task sets. The overall efficiency of the system hinges on the performance capabilities of the processors. There are numerous factors that exist to address and improve the system performance. Next section deals with effective methods which plays an important role in the improvement of the performance [1-4].

It is often difficult to schedule a program set over separate CPUs or processors in order to maximize the 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

ISSN PRINT 2319 1775 Online 2320 7876

Research paper © 2012 IJFANS. All Rights Reserved, Journal Volume 11, Iss 12, 2022

processor works together, scheduling becomes a challenging task. Nowadays, heuristic methods are widely used in various engineering and scientific applications. Several heuristic methods have also been provided for solving the parallel processor scheduling problems. A natural heuristic approach to the problem is the list scheduling which is again classified into BNP and APN class of scheduling. Heuristic is the approach for searching or doing anything on-the-basis of one specific property [1-4].

II. EFFECTIVE METHODS

There are so many methods which are used as effective parameters for scheduling, that can be evaluated by measurement parameter, to measure the performance of effective methods on scheduling. By the measurement parameter analysis, the performance of scheduling can be measured to decide whether the performance is improved or not. The below section present some effective method.

DAG (Direct Acyclic Graph):

DAG represents relationship between task and their precedence. The mathematical representation of a DAG is, G = (N, E), consists N is finite set of n nodes and E is set of m directed edges. In DAG, the nodes represent tasks partitioned from an application; the edges represent precedence constraints. An edge (i ,j) joins i^{th} and j^{th} nodes, where i^{th} node is called the parent node and j^{th} node is called the child node. That means j^{th} node cannot start its execution until i^{th} node finish its execution and send its data to j^{th} node. That represents as inter task communication. A task with no parent or predecessors is called an entry task and a task with no child or successor is known as exit task [1] [5]. In some cases, author represent four tuple in a DAG, G (V, E, C, W), where C represent set of communication and W represents computation costs [1][5][6].

DVS (Dynamic Voltage Scaling)

DVS used to save energy, which varies the processor speed and supply voltage according to the workloads at run-time. DVS takes advantage of the relationship between supply voltage and power consumption and save power, if work load is less, then supply less power while work load is increase supply was also increase [7].Energy of system is combination of energy of CPU and device standby energy. Optimal speed setting set the system energy according to specific task, and limited pre-emption reduces the numbers of possible preemptions [8].

Slack

Slack is a time gap between actual execution and deadline, it becomes positive or negative corresponding to execution [7]. The slack can be added to future task or used to reduce the" possibility of missing deadline. By reallocating the slack for future tasks to save energy or satisfy the deadline constraints [9].

ISSN PRINT 2319 1775 Online 2320 7876
 Journal Volume 11, Iss 12, 2022

Backfilling

Backfilling are used in case of parallel job execution where gang of jobs is executing parallel when required resources are not available it allows to backfill small jobs previously. This scheduling technique dramatically improves utilization; it also requires that all jobs' service times be known. This information can come from estimates by prediction. This prediction makes inaccuracy on runtime and affects the performance of overall system. The cause of inaccuracy is the unfitted backfilling of jobs. There are two important things which are known before backfilling is; the service time of a job and the exact time that all needed resources will be free for the parallel job to start their execution. Backfilling used to minimize fragmentation of system resources by executing jobs in different order from their submission order. The backfilling is the cause of delay of parallel job execution. The backfilling applies on both single queue and multiple-queue system [10] [11].

Workload

Workload or load is another important method in scheduling, but it can calculate after initial scheduling of processes on processor. It required in condition of unfair workload distribution, where some processors are overloaded and some are idle. So keep the workload evenly distributed over the all processor, use "push migration" that periodically check the load, and "push" processes to less loaded queues and use "pull migration" in which idle processors "pull" processes from busy processors. It comes in three forms, sender initiated, receiver-initiated and hybrid sender-receiver-initiated [12] [13].

III. MEASUREMENT PARAMETERS

Measurement parameters are helpful to describe the performance of different algorithms.

Parallel time: Difference between the time at which the last process ends and the time at which the process starts is known as parallel time. For better performance Parallel time should be less. Parallel time denoted as T(n).

Speedup: Speedup is the ratio of sequential execution time and parallel execution time.

$S(n)=T(1)/T(n)$

Where T(1) is the Parallel time of uniprocessor or sequential time.

Efficiency: Efficiency is the ratio of speedup and number of processors used

$E(n)=S(n)/n$

Makespan: Makespan is the overall execution time of all processes on all processors. It is also known schedule length.

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

ISSN PRINT 2319 1775 Online 2320 7876
Research paper© 2012 IJFANS. All Rights Reserved, Journal Volume 11, Iss 12, 2022

Turnaround time: The interval from the time of submission of a process to the time of completion is the turnaround time.
Turnaround time= completion time - submission time [1-14]

IV. REVIEW ON BNP SCHEDULING

Author & title	Objective	Name of algorithm	Task model	System mode	Performance metric	Conclusion
S. Kushwaha, V. Thakur, "Heuristic Oriented Process Scheduling for Homogeneous Multiproce-ssor Environment"	Apply heuristic techniques to achieve optimum schedule length (makespan) for multiprocessor system.	LPT, SPT, ECT, EST, WSPT, EDF, EDD, LPEST	Independent Task set	Homogeneous system	AIR, Makespan, Speedup	On the basis of simulation result, most of the time, the performance of proposed LPEST algorithm is better than some well-known selected list heuristic scheduling algorithms LPT, SPT, EST ECT, EDF, ED and WSPT. Proposed algorithm LPEST is more suitable in use of independent process where processes arrived at random time.
H. Topcuoglu, S. Hariri and Min-You Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing"	-Present two novel scheduling algorithms for bounded number heterogeneous processor. - Objective is to simultaneous meet high performance and fast scheduling time	- HEFT, CPOP -Parametric Graph Generator Designed To Generate Weighted DAG With Various Characteristics	DAG, Dependent task	BNP Heterogeneous	Comparison study of algorithm is based on randomly created task graph and task graph of some real application	The algorithm significantly surpass previous approach in term of both quality and cost of schedule
Y. Kang and Y. Lin, A Recursive Algorithm For Scheduling of Tasks In A Heterogeneous Distributed Environment	- the complexity of the problem increases when task scheduling is to be done in a heterogeneous environment. -To achieving high performance in Heterogeneous computing system, author presents a recursive task Scheduling algorithm for a bounded number of heterogeneous processors run on the network of heterogeneous systems.	Recursive task Scheduling algorithm for a bounded number of heterogeneous processors run on the network of heterogeneous systems.	Dependent Task set	Heterogeneous distributed system	-	The performance of the algorithm is illustrated by comparing the scheduling length ratio, frequency of best results with the existing effectively scheduling algorithms, heterogeneous earliest finish time and iterative list scheduling algorithm.
Michael A. Iverson, Fusun Ozguner, "Parallelizing Existing Applications in a Distributed Heterogeneous Environment"	-Parallelizing existing application in a distributed heterogeneous computing to large finite element application code CSTEM.	LMT (Levelized Min-Time), CSTEM	Dependent task set	Heterogeneous system	Speedup	-Heterogeneous computing has potential to significantly increase performance -Effective mapping and scheduling algorithm is an essential element of Heterogeneous computing system.

ISSN PRINT 2319 1775 Online 2320 7876

 Journal Volume 11, Iss 12, 2022

Author & title	**Objective**	**Name of algorithm**	**Task model**	**System mode**	**Performance metric**	**Conclusion**
Yu-Kwong Kwok, Ishfaq Ahmad, "Efficient Scheduling of Arbitrary Task Graphs to Multiprocessors Using a Parallel Genetic Algorithm"	Designing a new scheduling scheme Which has a high capability to generate optimal solutions and is also fast and scalable.	Novel GA based algorithm (PGS)	Dependent task set	Heterogeneous System, Homogeneous System	-complexity -scalability -perform-ance	-It is found that proposed algorithm can generate optimal solution for majority of test cases. -PGS algorithm shown encouraging performance
C. Fu, Y. Zhao, M. Li and C. J. Xue, "Maximizing common idle time on multi-core processors with shared memory"	Proposed technique used to reduce memory energy by orchestrating cores activities and turning the memory into sleep mode/state as much as possible	LEPDA, Least-Laxity-First Assigning Algorithm (LLFAA), LLF	Preemptive, single interval , Independent task set	-BNP -bounded number of cores -Heterogeneous system	idle time (memory sleep time) , schedule length	-LLFAA (Least-Laxity-First Assigning Algorithm) is an efficient and effective and can reduce memory energy by approx. 26% as compared to a conventional multi-core scheduling scheme.
T. Hagras and J. Janecek, "A high performance, low complexity algorithm for compile-time task scheduling in heterogeneous systems"	-Introduced simple list scheduling mechanism for task selection. -low complexity duplication-based mechanism for machine assignment	-HCPFD, (Heterogeneous Critical Parents with Fast Duplicator), HEFT, CPOP	DAG, Dependent task set	BNP fully connected Heterogeneous system	-Avg SLR. -Makespan quality of scheduling	For bounded number of computing machines with different capabilities, HCPFD outperformed than other algorithms in term of performance & complexity
D. Sirisha, "Slack based Scheduling for dependent tasks in Heterogeneous Computing environments"	Utilization of slack between tasks. And proposed a slack based algorithm STS. And STS algorithm performance compared with HEFT and PETS algorithm	STS (Slack Based Task Scheduling.), HEFT and PETS.	DAG, Dependent task set	BNP fully connected. Heterogeneous system	Average scheduling length, Speedup, Efficiency	-STS performance compared with PETS & HEFT and found to be generating better quality schedules. - Randomly generated task-graph shows that, STS outperformed PETS & HEFT
Y. Zhou, G. Sun, Y. Jiang and Y. Xu, "An Effective Scheduling Algorithm for Homogeneous System"	-Proposed a Novel Algorithm Based on List-Scheduling and Task Duplication on BNP Fully Connected Homogeneous Machines -Compile time task scheduling algorithm based on list-scheduling and task duplication algorithm.	FDEFT, CUPFD, MCP, FLB	DAG, Dependent task set	BNP fully connected, homogeneous system	SLR Speedup QRS	-FDEFT outperformed then other three algorithms and found better with all algorithm. -Performance result compared and average SLR shown in graph -HCPFD compared with CUPFD, MCP and FLB
K. Agrawal, I. A. Lee, J. Li, K. Lu and B. Moseley, "Practically Efficient Scheduler for Minimizing Average Flow Time of Parallel Jobs"	Scheduling parallel job that arrival online by introducing an adaptive of work-stealing scheduler for average flow time.	New algorithm DREP (Distributed Random Equi-Partition)	Dependent task set	Homogeneous System	-	-Practically efficient scheduler for optimizing average flow time of parallel job. -The algorithm has slightly worse theoretical than best known algorithm for problem. -In practice, it is an efficient algorithm for parallel job -Evaluation demonstrates strong performance.

V. CONCLUSION

After study of various research articles it is clear that the task scheduling tends to be a complex and important operation, which plays an important role to improve the scheduling. There are some methods, such as DAG, DVS, Slack, backfilling, and load, that are useful to achieve different aspect such as energy saving, optimal scheduling length, better throughput, and balance for scheduling of processes on processor. Some other methods are also exist that are used to improve the scheduling of various type of system. These methods are very useful to take advantages of system resources and utilize their capacity and provide better performance. Several performance measurement parameters are also used to check the performance of scheduling algorithms. This paper explores various algorithm classes tailored for homogeneous and heterogeneous processors, delving into the roles and significance of performance parameters and effective techniques used to improve the performance of algorithms.

REFERENCE:

1. Singh, E. N., Kaur, G., Kaur, P., & Singh, G., "Analytical performance comparison of BNP scheduling algorithms" Global Journal of Computer Science and Technology, Volume 12 Issue 10 (2012) , PP 1-9.
2. 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.
3. Yu-Kwong Kwok, Ishfaq Ahmad, "Static Scheduling Algorithm for Allocating Directed Task Graph to multiprocessors", ACM Computing Surveys, Vol. 31,no. 4, December 1999.
4. 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.
5. Young Choon Lee, Member, and Albert Y. Zomaya," Energy Conscious Scheduling for Distributed Computing Systems under Different Operating Conditions" , IEEE Trans Parallel and Distributed System, vol. 22, pp. 1374-1381, August 2011.
6. Olga Rusanova and Alexandr Korochkin, "Scheduling Problems for Parallel and Distributed Systems", ACM 1-58113-127-5/99/0010, pp. 195-202. Sep 1999.
7. Santhi Baskaran1 and P. Thambidurai2, "Energy Efficient Real-Time Scheduling in Distributed Systems", IJCSI International Journal of Computer Science Issues, vol. 7, Issue 3, No 4, pp. 35-42, May 2010.
8. Jianli Zhuo , Chaitali Chakrabarti ," System Level Energy Efficient Dynamic Task Scheduling.", ACM DAC ,Anaheim, California, USA , pp. 13-17, 2005.
9. Jaeyeon Kang, Sanjay Ranka," Dynamic slack allocation algorithms for energy minimization on parallel machines", JPDC, vol 70, Issue 5, pp. 417–430, May 2010.
10. Sofia K. Dimitriadou, Helen D. Karatza, "Job Scheduling in a Distributed System Using Backfilling with Inaccurate Runtime Computations", 2010 International Conference on Complex, Intelligent and Software Intensive Systems, pp. 329-336, 2010.
11. Barry G. Lawson, Evgenia Smirni, Daniela Puiu, "Self-adapting Backfilling Scheduling for Parallel System", National Science Foundation under grants EIA-9977030, EIA-9974992, CCR-0098278, and ACI-0090221.

ISSN PRINT 2319 1775 Online 2320 7876

 Journal Volume 11, Iss 12, 2022

12. Vinay Harsora, Dr.Apurva Shah," A Modified Genetic Algorithm for Process Scheduling in Distributed System", IJCA Special Issue on "Artificial Intelligence Techniques - Novel Approaches & Practical Applications"AIT, pp. 36-40, 2011.
13. Nazleeni Samiha Haron, Anang Hudaya Muhamad Amin, Mohd Hilmi Hasan, Izzatdin Abdul Aziz, and Wirdhayu Mohd Wahid, "Time Comparative Simulator for Distributed Process Scheduling Algorithms", World Academy of Science, Engineering and Technology, pp. 84-89, 2006.
14. Sunita Kushwaha, Varsha Thakur "Heuristic Oriented Process Scheduling for Homogeneous Multiprocessor Environment", International Journal of Mechanical Engineering ISSN: 0974-5823 Vol. 7 No. 5 May, 2022.
15. Topcuoglu, Haluk, Salim Hariri, and Min-You Wu. "Performance-effective and low-complexity task scheduling for heterogeneous computing." IEEE transactions on parallel and distributed systems 13.3 (2002): 260-274.
16. Kang, Yan, and Ying Lin. "A recursive algorithm for scheduling of tasks in a heterogeneous distributed environment." 2011 4th International Conference on Biomedical Engineering and Informatics (BMEI). Vol. 4. IEEE, 2011.
17. Iverson, Michael A., Fusun Ozguner, and Greg Follen. "Parallelizing existing applications in a distributed heterogeneous environment." 4th Heterogeneous Computing Workshop (HCW'95). 1995.
18. Kwok, Yu-Kwong, and Ishfaq Ahmad. "Efficient scheduling of arbitrary task graphs to multiprocessors using a parallel genetic algorithm." Journal of Parallel and Distributed Computing 47.1 (1997): 58-77.
19. Fu, Chenchen, et al. "Maximizing common idle time on multicore processors with shared memory." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 25.7 (2017): 2095-2108.
20. Hagras, Tarek, and Jan Janecek. "A high performance, low complexity algorithm for compile-time task scheduling in heterogeneous systems." 18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.. IEEE, 2004.
21. Sirisha, D. "Slack based Scheduling for dependent tasks in Heterogeneous Computing environments." (2013): 279-286.
22. Zhou, Yipeng, et al. "An effective scheduling algorithm for homogeneous system." 2006 Fifth International Conference on Grid and Cooperative Computing (GCC'06). IEEE, 2006.
23. Agrawal, Kunal, et al. "Practically efficient scheduler for minimizing average flow time of parallel jobs." 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, 2019.