

Dynamic Task Scheduling Methods in Heterogeneous Systems- A Survey

D. I. George Amalarethinam
Director – MCA and Associate Professor,
Department of Computer Science,
Jamal Mohamed College,
Tiruchirappalli, India.

A. Maria Josphin
Research Scholar,
Jamal Mohamed College,
Tiruchirappalli, India.

ABSTRACT

Heterogeneous systems create unlimited opportunities and challenges in the fields of parallel processing, design of algorithms, and partitioning and mapping of parallel tasks. In mapping of parallel tasks, scheduling plays an important role. The scheduling decisions are based on the dynamic parameters that may change during run time. The goal of scheduling is to utilize all the processors with minimum execution time by proper allocation of tasks to the processors. Task scheduling achieves high performance in the heterogeneous systems. A Parallel application can be represented by a Directed Acyclic Graph (DAG), which represents the dependency among tasks, based on their execution time and communication time. Previous studies reveal that many researchers have designed efficient scheduling algorithms with different parameters on multiprocessor systems. This paper investigates different aspects in scheduling and issues in various levels of the heterogeneous systems.

Keywords

Heterogeneous System, Dynamic Scheduling, Directed Acyclic Graph (DAG), Homogenous System, Static Scheduling

1. INTRODUCTION

1.1 Heterogeneous Systems

In general, a distributed computing system is defined as a group of processors connected via a high speed network, which chains the execution of distributed applications [1]. Distributed computing systems can be divided into two main categories: Homogeneous Distributed Computing Systems (HoDCS), a collection of identical processors and Heterogeneous Distributed Computing Systems (HeDCS), a collection of diversely capable processors. HeDCS requires compile – time and runtime support for executing applications. HeDCS is a novel method for solving computationally intensive problems with entrenched parallelism. HeDCS is generally used in scientific and commercial applications including real time safety – critical application [2]. They are adopted by many High Performance Computing (HPC) system designers and vendors [3]. Different architectures for heterogeneous computing consist of mainframe with integrated vector unit, vector processors, attached processors, Multiprocessor and Multi computing systems and special purpose architectures [4]. Heterogeneity can be proposed in two types, namely, Function-level heterogeneity and Performance – level heterogeneity[5]. In function – level heterogeneity, systems combine general-purpose processors with special-purpose processors, such as vector units, floating – point co – processors, and input/output processors. All tasks may not be executed on all functional

units. In performance – level heterogeneity, systems contain several interconnected general – purpose processors of varying speeds. In these systems, a task can be executed on any processor at a faster rate than others.

One of the key concepts of heterogeneous systems is scheduling problem. To achieve higher performance, the efficient scheduling of the tasks of an application is needed. Task scheduling problem includes the problem of assigning the tasks of an application to suitable processors and to order the task executions on each resource [6]. Given an application modeled by a Directed Acyclic Graph (DAG), the scheduling problem deals with mapping each task in heterogeneous environment in order to minimize the execution time (makespan).

1.2 Scheduling

Scheduling is a process that assigns tasks to resources for the execution at a particular time. The main objective of scheduling is to map the tasks onto the processors to minimize the makespan and maximize the reliability. The task scheduling can be categorized as static and dynamic.

In static scheduling, execution time of tasks and data dependencies between the tasks is known in advance. This type of scheduling takes place during compile time. It is called as offline deterministic scheduling. The problem of static task scheduling is known to be NP – complete [6] i.e., task list is not updated with new ordering at runtime. Scheduling a DAG in polynomial time is identified in three cases. First case is to schedule a uniform node – weight free – tree to an arbitrary number of processors and then to schedule an arbitrarily structured DAG with uniform node-weight to two processors. Finally the third case is to schedule an interval – ordered DAG with uniform node-weights to an arbitrary number of processors.

In dynamic scheduling, tasks are allocated to processors upon their arrival and scheduling decisions must be made at run time. Scheduling decisions are based on dynamic parameters that may change during run time. In dynamic scheduling tasks can be reallocated to other processors during the run time [6, 7]. Dynamic scheduling is flexible and faster than the static scheduling.

1.3 Directed Acyclic Graph (DAG)

A Directed Acyclic Graph (DAG) is a directed graph which contains the vertex or the node weight represents task processing time and the edge weight represent data dependencies and communication time between tasks. The communication time is also called as communication cost [8, 9]. Directed Acyclic Graph $G = (V, E)$, where V is a set of v nodes/vertices and E is a set of e directed edges. Source node of an edge is called parent node and sink node is called child

node. A node with no parent is called entry node. A node with no child is called exit node [10]. A sample DAG is shown in Figure.1, which contains $T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9,$ and T_{10}

tasks. T_1 is an entry node, which has no parent node and T_9 is an exit node, with no child node. T_2 is a parent node of T_4 and T_5 .

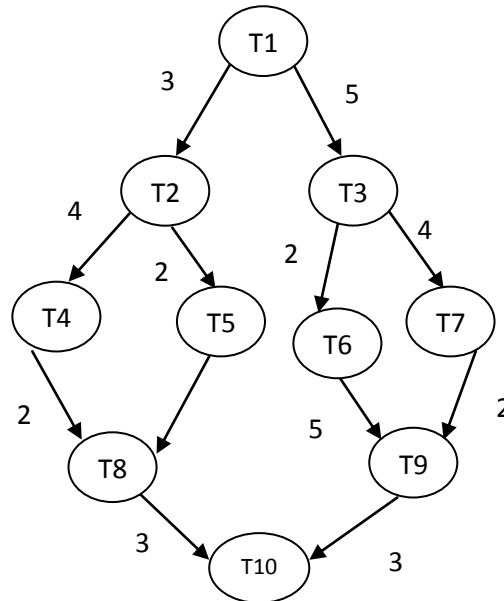


Figure1: A Sample DAG

2. RELATED WORKS

Many research works investigated the task scheduling problems in different aspects and the problems faced in various levels of the heterogeneous systems. The important contributions of task scheduling algorithms in heterogeneous systems are discussed in this section.

2.1 Monte Carlo Algorithm

Wei Zheng and Rizos Sakellariou proposed an algorithm called Monte Carlo approach [11]. This algorithm uses static scheduling method. The main purpose is to minimize the makespan. This algorithm avoids the complex computation involved with random variable applicable to any random distribution. Threshold values are used to optimize the scheduling. The probabilistic distribution is applied to minimize the makespan.

2.2 CBHD Algorithm

Abdelkader and Omara proposed Clustering based HEFT with Duplication, used for efficient scheduling and mapping for tasks on heterogeneous systems, to break the program into subtasks and simultaneously work with other tasks [12]. This algorithm is considered as combination between the Triplet clustering algorithm, by clustering tasks into interrelated group, and the HEFT algorithm, the task in each cluster is ordered based on rank to improve the performance of scheduling algorithm with load balance. The main objective is to minimize the execution time, maximize processor utilization and the load balancing between the processors.

2.3 P-HEFT Algorithm

Jorge G. Barbosa and Belmiro Moreira [13] proposed an algorithm called P-HEFT (Parallel Heterogeneous Earliest Finish Time). This algorithm handles parallel tasks in heterogeneous clusters with good efficiency without compromising the makespan. The main objective of this algorithm is to minimize makespan and batch of jobs with

different arrival times that change the processors assigned to a job during its execution.

2.4 High Performance and Energy Efficient Task Scheduling Algorithm

Ilavarasan and Monoharan [14] proposed a new algorithm which focuses on the minimization of schedule length and energy/power consumption. This algorithm has two phases namely, compilation time phase and run time phase. The compilation time phase has three stages, such as level sorting, task prioritization and processor selection. During the run time, to conserve the energy this algorithm reschedules the task to the ideal nodes from the busy nodes. This algorithm performs significantly better than the conventional algorithm.

2.5 Constrained Earliest Finish Time (CEFT) Algorithm

Constrained Earliest Finish Time was proposed by Minhaj Ahmad Khan [15]. This novel approach is to provide better scheduling for heterogeneous systems using the concept of constrained critical paths (CCPs). Once the CCP in DAG are found, the tasks are scheduled using the finishing time of the entire CCPs. This approach facilitates in producing schedules with shorter makespan and works with small complexity.

2.6 Sorted Nodes in Leveled DAG Division (SNLDD) Algorithm

Nirmeen et al [16] proposed the algorithm called high performance task scheduling algorithm. This type of algorithm divides the DAG into different levels and each level is sorted in descending order according to their computation time, which reduces the dependency between the tasks. Static task scheduling is applied in heterogeneous systems with limited number of processors. This type may produce high quality of task scheduling. The computation time of all tasks in DAG is computed once, hence the run time overhead is eliminated. The sleek time of processors are also minimized, because DAG are leveled and assigned to the processors.

2.7 Multi – Queue Balancing Algorithm

Multi – Queue balancing algorithm was proposed by Yuxiong He et al [17] introduced scheduling concepts in functional units. This algorithm has various types of functional units, which does not use limited number of hardware and software. This type of algorithm is called offline and non-deterministic algorithm. The main objective of this algorithm is to minimize job completion time and assign tasks to the machines according to the availability of resources and maximize utilization of heterogeneous systems. Each task is executed on its matching processors.

2.8 Clustering Scheduling Strategies Algorithm

Zhi Chen et al [18] proposed the algorithm to reduce the number of configuration and execution time of applications while enhancing the utilization of field programmable gate way (FPGA) device. Two methods were introduced namely, heuristic scheduling strategy and dynamic programming scheduling strategy, to divide the Directed Acyclic Graph (DAG) on several clusters on reconfigurable computing systems.

2.9 SD-based Algorithm for Task Scheduling (SDBATS)

SD-Based Algorithm proposed by Ehsan Ullah Muniret et al [19], is a standard deviation approach for the expected execution time of a given task on the available resources in the heterogeneous computing environment. The proposed algorithm applied various conditions on a standard task graph application such as Gaussian Elimination and Fast Fourier Transformation applications DAGs. This approach generated high quality schedules and produced low cost schedule and system efficiency.

2.10 Online Scheduling of Dynamic Task Graphs

Pravanjan Choudhury et al [20] stated that the Online Scheduling of Dynamic Task Graphs is more realistic compared to conventional schemes and task graphs are subject to variation at runtime and interprocessor communications of fixed number of channels. Broadcast and point – to – point communication is applied in online scheduling. The main objective of this algorithm is to reduce makespan.

2.11 Node Duplication Modified Genetic Algorithm Approach (NMGA)

Rachhpal Singh [21] proposed Node Duplication Modified Genetic Algorithm Approach that adopted top and bottom level approach. It exhibits the efficiency of Node duplication techniques. The main purpose is to minimize the completion time and increase the system throughput. Task is duplicated to reduce the overall time.

2.12 Heuristic based for Genetic Algorithm

Heuristic based genetic algorithm was proposed by Probir Roy et al., which requires less computation time [22]. This algorithm was based on task scheduling in multiprocessor systems by choosing the eligible processor to reach the suboptimal solution. Bottom level approach is done by choosing eligible processor for assigning task to get minimum makespan.

2.13 List based Heuristic Task Scheduling

Shailza Kamal [23] proposed a list based Heuristic task scheduling with different type of task priorities to get minimum schedule length and speed. This inherits only static scheduling, which is further classified into job scheduling and task scheduling. The main purpose of this algorithm is to minimize execution time and communication delays and maximize resource utilization.

2.14 Efficient Genetic Algorithm

Mohammad I. Daoud and Nawwaf Kharma [24] proposed an algorithm called Efficient Genetic Algorithm which inherits a customized genetic algorithm to produce high – quality task schedules for HeDCSSs. The performance of the new algorithm was achieved by two scheduling algorithms namely HEFT algorithm and the DLS algorithm. This is applied on both randomly generated task graphs and task graphs of certain real-world numerical applications. The main objective of this algorithm is to increase schedule length, speedup and efficiency.

2.15 BNP Scheduling Algorithms

Navneet Singh [25] proposed an algorithm called BNP Scheduling, which introduced a parallel program represented by an edge-directed acyclic graph (DAG) to a set of homogenous processors. The main objective is to minimize the completion time. Several classes of algorithms and performance of a class of scheduling known as the bounded number of processors (BNP). Comparison is based on different scheduling parameters such as makespan, speed up, processor utilization and scheduled length ratio. The main focus is to increase the number of tasks on processors for the performance of these algorithms.

2.16 Predict Earliest Finish Time (PEFT)

Hamid Arabnejad and Jorge G. Barbosa [26] proposed a novel approach called Predict Earliest Finish Time (PEFT) for heterogeneous computing systems. This algorithm has the same time complexity that is, $O(v^2 \cdot p)$ for v tasks and p processors, introducing a feature without increasing the time complexity connected with computation of an Optimistic Cost Table (OCT). The designed value is an optimistic cost because processor availability is not measured in the computation. This algorithm is only based on an OCT table that is used to rank tasks and for processor selection. PEFT algorithm performs list-based algorithms for heterogeneous systems in terms of schedule length ratio, efficiency and frequency.

2.17 Hybrid Genetic Scheduling Algorithm

Yan Kang and Defu Zhang [27] stated a hybrid genetic task scheduling algorithm for the tasks run on the network of heterogeneous systems and represented by Directed Acyclic Graphs (DAGs). This algorithm assigns a coupling factor to each task and scheduled onto the same processor by avoiding the large communication time. Objective of this algorithm is to generate high quality initial solution by scheduling the tasks which are powerfully coupled with each other onto the same processor and to improve the quality of the solution by using coupling initial solutions, random solution, near optimal solutions obtained by the list scheduling algorithm in the crossover and mutation operator.

2.18 Fork – Join Method (TSFJ)

Ranjit Rajakand Katti [28] proposed an algorithm called Fork-Join Method (TSFJ) which inherits task scheduling in multiprocessor system is to minimize the total execution time, so that it can reach maximum speed – up and efficiency. The application program is represented by directed acyclic graph (DAG) and tasks are allocated to the processors based on fork-join structure. The main performance of the algorithm is based on schedule length, speedup efficiency, and load balancing.

2.19 Non – Dominated Sorting Genetic Algorithm – II (NSGA-II)

Rathna Devi and Anju [29] proposed an algorithm called Non – dominated sorting Genetic Algorithm – II, which introduced scheduling applications on heterogeneous multiprocessor system focused on a single purpose such as execution time, cost or total data transmission time. Proposed algorithm is to develop a multi objective scheduling algorithm using evolutionary techniques for scheduling a set of dependent tasks on available resources in a multiprocessor environment. The main objective is to minimize the makespan.

2.20 Task Scheduling and Energy Conservation Techniques

Nirmal Kaur et al [30] stated the algorithm called Task Scheduling & Energy Conservation Techniques for high performance computing systems provide versatile, low cost performance at the expense of huge energy consumption. This algorithm addresses the “state of the art” in energy aware scheduling techniques for dependent tasks in multiprocessor systems to reduce the overall makespan and reduction in energy consumption.

2.21 Critical Path Scheduling with T – level (CPST)

Vinay Kumar et.al [31] proposed an algorithm called critical path scheduling with t – level (CPST) for grid computing system. In CPST, a critical path based task sequence is generated with t – level value of tasks, where variance based computation and communication cost is used. Task are ordered and selected in non increasing order of their priorities and scheduled on processors to optimize various performance

2. COMPARISON OF VARIOUS FACTORS IN TASK SCHEDULING ALGORITHMS

metrics. The objective is to map tasks onto the suitable resources and minimize the makespan.

2.22 Multi – Core Scheduling

Poonam Karande et al [32] stated an algorithm called Multi – core scheduling by Introducing, Heterogeneous coprocessor that is one processor and multiple heterogeneous coprocessor. The Processor is to handle the control signal & heterogeneous coprocessor for data computation because of multiple co-processors dispatcher is required to establish which subtasks transmit to which co-processor. This algorithm applied migration policy to increase the utilization of resources.

2.23 Task Scheduling Algorithm using Merge Conditions

Task Scheduling Algorithm, proposed by Tae-Young Choe [33], introduced post-processing scheduling algorithm that reduces the number of processors while preserving its schedule length. This algorithm finds merge pairs based on the condition and merge them without increasing schedule length. A function is applied to find maximum pair to reduce the number of merge.

2.24 Linear Programming based Scheduling Algorithms

Duksu Kim et al [35] proposed an algorithm called Linear Programming based scheduling Algorithms applied into variety of proximity queries in five different applications that have different characteristics, and robustly provide the performance improvement with additional computing resources. The proposed algorithm is designed for parallel proximity computation and to minimize the largest time spent on computing resource.

2.25 Comparative Study of Scheduling Algorithm

Klavdiya Bochenina [36] proposed an algorithm called comparative study of scheduling algorithms for mapping the tasks of multiple workflows with different deadlines on the static set of resources. The main purpose is to map the tasks to the processors, maximizing the efficiency of resource utilization while meeting all workflows deadlines.

Table. 1 Existing Task Scheduling Algorithms

S. No	Algorithm	Objectives	Merits	Future Enhancement
1	Monte Carlo based DAG scheduling approach	To minimize the makespan	Avoid the complex computation with random variable and applicable to any random distribution	Dynamic scheduling
2	Clustering based HEFT with duplication	To minimize the execution time	Maximize processor utilization and load balancing.	Better optimization method
3	P-HEFT	To minimize the completion time	Efficient and best make span	Optimization of multi user environment

4	High performance and energy efficient task scheduling algorithm	To minimize the makespan	Schedule length and energy consumption	Large set of task graph
5	Constrained Earliest Finish Time	To minimize the execution time	Minimum threshold values for schedule length	Use to multiple constrained critical path in subsequent of machine
6	Sorted Nodes in Leveled DAG Division	Speedup, efficiency, complexity and quality	Generate high quality of task schedule	Dynamic scheduling
7	Multi Queue Balancing Algorithm	To minimize the completion time and utilization of maximum resources	Minimum completion time and maximum utilization of processors	Schedule more job on heterogeneous system
8	Heuristic and Dynamic programming scheduling strategy	To minimize the Execution time	High parallelism without loss of generality	Thermal and power performance
9	SD-Based Algorithm for Task Scheduling	Schedule length and speed up	Reduce execution time and assign task priority.	Homogenous systems
10	Online Scheduling of Dynamic Task Graphs	To minimize the make span	Interprocessor communications of fixed number of channels	Off line
11	Node duplication Modified Genetic Algorithm Approach	To minimize the completion time and throughput	Task is duplicated to reduce the overall time	Non deterministic homogenous systems
12	Heuristic based for genetic algorithm	To minimize the make span	Bottom level approach by choosing eligible processor for assigning task	Non deterministic homogenous systems
13	List based heuristic task Scheduling	Execution time, communication delays and maximizing resource utilization	Efficiency of parallel architectures. Task execution time, communication cost and task dependencies are available before execution.	Dynamic scheduling
14	An Efficient Genetic Algorithm	Schedule length, speedup and efficiency.	Both randomly generated application graphs and application graphs that correspond to	Partially-connected Networks of heterogeneous processors.
15	BNP Scheduling Algorithms	To minimize the completion time	Increasing the number of tasks and processors on the performance	Heterogeneous Systems
16	Predict Earliest Finish Time	To minimize the make span	Schedule length ratio, efficiency and frequency	Homogenous Systems
17	A Hybrid Genetic Scheduling Algorithm	Assign tasks to available processors and Make span	Avoid large communication time. Long communication cost, more effective with the graph structure is more flexible.	Better optimization method
18	Fork-Join Method	To minimize the total execution time	Scheduling length, speedup, efficiency and load balancing	More number of nodes.

19	Non-dominated sorting Genetic Algorithm-II	To minimize the makespan and reliability cost	Used to select best optimal schedule in minimum time	Dynamic scheduling
20	Task Scheduling & Energy Conservation Techniques	To reduce the overall makespan and reduction in energy consumption	High performance computing systems versatile, low cost performance at huge energy consumption.	Heterogeneous Systems a set of random and with respect to different task graph characteristics.
21	Critical path scheduling with t-level	To minimize overall execution time, and makespan	Tasks are ordered and selected in non-increasing order of their priorities	Homogenous System
22	Multi-core Scheduling	To minimize the response time and to increase the utilization of resources	It uses dispatcher mechanism to dispatch different subtask and migration policy to reduce response time	Homogenous System
23	Task Scheduling Algorithm using Merge Conditions	To minimize the schedule length	To reduce the Number of merge.	Pre – process scheduling approach
24	Scheduling in Heterogeneous Computing Environments for Proximity Queries	To minimize the makespan and proximity computation in heterogeneous computing systems	Robust and designed for parallel proximity computations	Best optimization methods for more variety of jobs
25	Comparative study of scheduling algorithm	To maximizing the efficiency of resource utilization	Efficiency and meeting deadlines.	Dynamic and e-Science infrastructure Platform

3. COMPARISON OF ALGORITHM WITH THE FACTORS INFLUENCING THE HETEROGENEOUS SYSTEM

A Comparison of factors influencing the heterogeneous environment has been taken for the consideration and different algorithms are discussed. The study reveals that the application of dynamic scheduling is quickly emergent research area. The feature of dynamic scheduling shows that it needs special attention in scheduling of resources. Standard algorithms to be reviewed with new parameters are identified in this survey.

Dynamic scheduling can be applied to Monte Carlo algorithm. Scheduling more jobs on heterogeneous system can be applied to Multi Queue Balancing. Sorted Nodes in Leveled DAG Division (SNLDD), List based Heuristic task Scheduling and Comparative study of scheduling algorithm can be applied by dynamic scheduling. Heterogeneous systems can be applied to BNP scheduling Algorithms and Task Scheduling & Energy Conservation techniques.

4. CONCLUSION AND FUTURE WORK

Scheduling algorithms were developed to schedule tasks in heterogeneous systems for efficient scheduling. Heterogeneous systems maximize the performance of a given application program with constraints. In this paper, various task scheduling algorithms in heterogeneous systems has been studied and classified. The merits and demerits of the algorithm were also discussed. This survey is mainly focused on the parameters such as makespan, resource utilization load balancing, speedup, efficiency and high performance.

Various list task scheduling algorithms were proposed for static scheduling. Hence, the future work will focus on developing efficient task scheduling algorithms for higher performance by applying heuristic methods and evolutionary algorithm for better optimization in the task scheduling that will reduce the makespan. Monte Carlo based DAG scheduling approach and Sorted Nodes approaches can be applied in dynamic scheduling to maximize the resource utilization.

5. REFERENCES

- [1] 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.
- [2] YuHai, Shengsheng and XueLian. 2007. "A New Dynamic Scheduling Algorithm for Real – Time Heterogeneous MultiProcessor Systems," Workshop on Intelligent Information Technology Application, pp.112 – 115.
- [3] Mudalige, Giles, Thiyagalingam, Reguly, Bertolli, Kelly and Trefethen. 2013. "Design and initial performance of a high-level unstructured mesh framework on Heterogeneous Parallel systems", Parallel Computing, pp.669 – 692.
- [4] Ercegovac. 1983. "Heterogeneity in Supercomputer Architectures", Parallel Computing, Vol.7, Issue. 3, pp. 367 – 372.
- [5] Yuxiong He, Jie Liu and Hongyang Sun. 2011. "Scheduling Functionally Heterogeneous Systems with Utilization Balancing", IEEE International Parallel and Distributed Processing Symposium, pp. 1187 – 1198.

- [6] 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.
- [7] Zomaya, C. Ward, and B. Macey. 1999. "Genetic Scheduling for Parallel Processor Systems: Comparative Studies and Performance Issues", *IEEE Trans. Parallel and Distributed Systems*, Vol. 10, No. 8, pp. 795 – 812.
- [8] D.I.George Amalarethinam and G.J.Joyce Mary. 2011. "A new DAG based Dynamic Task Scheduling Algorithm (DYTAS) for Multiprocessor Systems", *International Journal of Computer Applications*, Vol. 19, No.8, pp. 0975 – 8887.
- [9] D.I.George Amalarethinam, and G.J.Joyce Mary. 2011. "DAGEN – A Tool To Generate Arbitrary Directed Acyclic Graphs Used For Multiprocessor Scheduling", *International Journal of Research and Reviews in Computer Science (IJRRCS)*, Vol.2, No.3, p.782.
- [10] Chitra, Rajaram, and Venkatesh. 2011. "Application and comparison of hybrid evolutionary multiobjective optimization algorithms for solving task scheduling problem on heterogeneous systems", *Applied soft computing*, pp.2725 – 2734.
- [11] Wei Zheng and Rizos Sakellariou. 2013. "Stochastic DAG scheduling using a Monte Carlo approach", *J.Parallel Distrib.Comput*, pp. 1673 – 1689.
- [12] Doaa M.Abdelkader and Fatma Omara. 2012. "Dynamic task scheduling algorithm with load balancing for heterogeneous computing system", *Egyptian International Journal*, pp.135 – 145.
- [13] Jorge G.Barbosa and Belmiro Moreira. 2011. "Dynamic scheduling of a batch of parallel task jobs on heterogeneous clusters", *Parallel computing*, pp. 428 – 438.
- [14] Ilavarasan and Monoharan. 2010. "High performance and energy efficient task scheduling algorithm for Heterogeneous mobile computing system", *International Journal of Computer Science and Information Technology*, Vol.2.
- [15] Minhaj Ahmad Khan. 2012. "Scheduling for heterogeneous systems using constrained critical paths", *Parallel Computing*, pp. 175 – 193.
- [16] Nirmeen A. Bahnasawy, Magdy A. Koutb, Mervat Mosa and Fatma Omara. 2011. "A new algorithm for static task scheduling for heterogeneous distributed computing systems", *African journal of Mathematics and Computer Science Research*, Vol.4(6), pp.221 – 234.
- [17] Yuxiong He, Jie Liu and Hongyang Sun. 2011. "Scheduling Functionally Heterogeneous Systems with Utilization Balancing", *IEEE International Parallel & Distributed Processing Symposium*, pp.1187 – 1198.
- [18] Zhi Chen et. al. 2013. "Clustering scheduling for hardware tasks in reconfigurable computing systems", *Journal of Systems Architecture*, pp.1424 – 1432.
- [19] Ehsan Ullah Munir et. al.2013. "SDBATS: A Novel Algorithm for Task Scheduling in Heterogeneous Computing Systems", *IEEE 27th International Symposium on Parallel and Distributed Processing Workshops and PHD Forum*, pp. 43 – 53.
- [20] Pravanjan Choudhury et al. 2012. "Online Scheduling of Dynamic Task Graphs with communication and contention for Multiprocessors", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 23, No. 1, pp.126 – 133.
- [21] Rachhpal Singh. 2012. "Modified Genetic Algorithm Approach to optimize Task Scheduling on Heterogeneous Multiprocessor Parallel System using Node duplication", *International Journal of Computers and Distributed Systems*, Vol.1, No.1.
- [22] Probir Roy et. al. 2012. "Heuristic based task scheduling in multiprocessor systems with genetic algorithm by choosing the eligible processor", *International Journal of Distributed and parallel Systems (IJDPS)*, Vol.3, No.4, pp.111 – 121.
- [23] Shailza Kamal. 2013. "Scheduling of parallel constrained task graphs on multiprocessor systems", *American International Journal of Research in Science, Technology, Engineering and Mathematics*, pp.153 – 158.
- [24] Mohammad I. Daoud and Nawwaf Kharma. 2006. "An Efficient Genetic Algorithm for Task Scheduling in Heterogeneous Distributed Computing Systems", *IEEE Congress on Evolutionary Computation*, pp.16 – 21.
- [25] Navneet Singh. 2012. "Analytical Performance Comparison of BNP Scheduling Algorithms", *Global Journal of Computer Science and Technology Hardware & Computation*, Vol. 12, Issue 10, Version 1.0.
- [26] Hamid Arabnejad and Jorge G. Barbosa. 2013. "List Scheduling Algorithm for Heterogeneous Systems by an Optimistic Cost Table", *IEEE Transaction on Parallel and Distributed Systems*, Vol. 25, Issue.3, pp.682 – 694.
- [27] Yan Kang and Defu Zhang. 2012. "A Hybrid Genetic Scheduling Algorithm to Heterogeneous Distributed System", *Applied Mathematics*, No. 3, pp.750 – 754.
- [28] Ranjit Rajak and Katti. 2013. "Task Scheduling in Multiprocessor System using Fork-Join Method (TSFJ)", *International Journal of New Computer Architectures and their Applications (IJNCAA)*, pp.47 – 53.
- [29] Rathna Devi and Anju. 2014. "Multiprocessor Scheduling of Dependent Tasks to Minimize Makespan and Reliability Cost Using NSGA-II", *International Journal in Foundations of Computer Science & Technology (IJFCST)*, Vol.4, No.2.
- [30] Nirmal Kaur et. al. 2013. "Task Scheduling & Energy Conservation Techniques for Multiprocessor Computing Systems", *International Journal of Networks and Systems*, Vol. 2, No.2.
- [31] Vinay Kumar et.al. 2014. "Scheduling Approach with Processor and Network Heterogeneity for Grid Environment", *International Journal on Computer Science and Engineering (IJCSE)*. ISSN: 0975-3397, Vol. 6 No. 01.
- [32] Poonam Karande et. al.2014. "Task Management for Heterogeneous Multi-core Scheduling (IJCSIT) *International Journal of Computer Science and Information Technologies*, Vol.5, pp. 636 – 639.
- [33] Tae – Young Choe. 2012. "Task Scheduling Algorithm to Reduce the Number of Processors using Merge Conditions", *International Journal on Computer Science and Engineering (IJCSE)*, ISSN: 0975-3397 Vol. 4 No. 02.
- [34] Duksu Kim et. al.2013. "Scheduling in Heterogeneous Computing Environments for Proximity Queries", *IEEE Transactions on Visualization and Computer Graphs*, Vol.9, No.9, pp.1513 – 1525.
- [35] Klavdiya Bochenina. 2014. "A comparative study of scheduling algorithms for the multiple deadline-constrained workflows in heterogeneous computing systems with time windows", *Procedia computer science*, Vol.29, pp.509 – 522.