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 applications [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-purpese 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 $T_1$, $T_2$, $T_3$, $T_4$, $T_5$, $T_6$, $T_7$, $T_8$, 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 $T_5$.

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

Nirmeenet 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 at 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 Pranab 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. Daoudand Nawwaf Khurma [24] proposed an algorithm called Efficient Genetic Algorithm which inherits a customized genetic algorithm to produce high – quality task schedules for HeDCSs. 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 (v2: 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 Delu 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 Kati [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 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 co-processor. 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
Klaviya 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.

### 2. COMPARISON OF VARIOUS FACTORS IN TASK SCHEDULING ALGORITHMS

<table>
<thead>
<tr>
<th>S. No</th>
<th>Algorithm</th>
<th>Objectives</th>
<th>Merits</th>
<th>Future Enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Monte Carlo based DAG scheduling approach</td>
<td>To minimize the makespan</td>
<td>Avoid the complex computation with random variable and applicable to any random distribution</td>
<td>Dynamic scheduling</td>
</tr>
<tr>
<td>2</td>
<td>Clustering based HEFT with duplication</td>
<td>To minimize the execution time</td>
<td>Maximize processor utilization and load balancing.</td>
<td>Better optimization method</td>
</tr>
<tr>
<td>3</td>
<td>P-HEFT</td>
<td>To minimize the completion time</td>
<td>Efficient and best make span</td>
<td>Optimization of multi user environment</td>
</tr>
<tr>
<td>No.</td>
<td>Algorithm Type</td>
<td>Objective</td>
<td>Constraints</td>
<td>Architecture</td>
</tr>
<tr>
<td>-----</td>
<td>----------------</td>
<td>-----------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>4</td>
<td>High performance and energy efficient task scheduling algorithm</td>
<td>To minimize the makespan</td>
<td>Schedule length and energy consumption</td>
<td>Large set of task graph</td>
</tr>
<tr>
<td>5</td>
<td>Constrained Earliest Finish Time</td>
<td>To minimize the execution time</td>
<td>Minimum threshold values for schedule length</td>
<td>Use to multiple constrained critical path in subsequent of machine</td>
</tr>
<tr>
<td>6</td>
<td>Sorted Nodes in Leveled DAG Division</td>
<td>Speedup, efficiency, complexity and quality</td>
<td>Generate high quality of task schedule</td>
<td>Dynamic scheduling</td>
</tr>
<tr>
<td>7</td>
<td>Multi Queue Balancing Algorithm</td>
<td>To minimize the completion time and utilization of maximum resources</td>
<td>Minimum completion time and maximum utilization of processors</td>
<td>Schedule more job on heterogeneous system</td>
</tr>
<tr>
<td>8</td>
<td>Heuristic and Dynamic programming scheduling strategy</td>
<td>To minimize the Execution time</td>
<td>High parallelism without loss of generality</td>
<td>Thermal and power performance</td>
</tr>
<tr>
<td>9</td>
<td>SD-Based Algorithm for Task Scheduling</td>
<td>Schedule length and speed up</td>
<td>Reduce execution time and assign task priority.</td>
<td>Homogenous systems</td>
</tr>
<tr>
<td>10</td>
<td>Online Scheduling of Dynamic Task Graphs</td>
<td>To minimize the make span</td>
<td>Interprocessor communications of fixed number of channels</td>
<td>Off line</td>
</tr>
<tr>
<td>11</td>
<td>Node duplication Modified Genetic Algorithm Approach</td>
<td>To minimize the completion time and throughput</td>
<td>Task is duplicated to reduce the overall time</td>
<td>Non deterministic homogenous systems</td>
</tr>
<tr>
<td>12</td>
<td>Heuristic based for genetic algorithm</td>
<td>To minimize the make span</td>
<td>Bottom level approach by choosing eligible processor for assigning task</td>
<td>Non deterministic homogenous systems</td>
</tr>
<tr>
<td>13</td>
<td>List based heuristic task Scheduling</td>
<td>Execution time, communication delays and maximizing resource utilization</td>
<td>Efficiency of parallel architectures. Task execution time, communication cost and task dependencies are available before execution.</td>
<td>Dynamic scheduling</td>
</tr>
<tr>
<td>14</td>
<td>An Efficient Genetic Algorithm</td>
<td>Schedule length, speedup and efficiency.</td>
<td>Both randomly generated application graphs and application graphs that correspond to</td>
<td>Partially-connected Networks of heterogeneous processors.</td>
</tr>
<tr>
<td>15</td>
<td>BNP Scheduling Algorithms</td>
<td>To minimize the completion time</td>
<td>Increasing the number of tasks and processors on the performance</td>
<td>Heterogeneous Systems</td>
</tr>
<tr>
<td>16</td>
<td>Predict Earliest Finish Time</td>
<td>To minimize the make span</td>
<td>Schedule length ratio, efficiency and frequency</td>
<td>Homogenous Systems</td>
</tr>
<tr>
<td>17</td>
<td>A Hybrid Genetic Scheduling Algorithm</td>
<td>Assign tasks to available processors and Make span</td>
<td>Avoid large communication time. Long communication cost, more effective with the graph structure is more flexible.</td>
<td>Better optimization method</td>
</tr>
<tr>
<td>18</td>
<td>Fork-Join Method</td>
<td>To minimize the total execution time</td>
<td>Scheduling length, speedup, efficiency and load balancing</td>
<td>More number of nodes.</td>
</tr>
</tbody>
</table>
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 emerging 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


