

Evaluation Performance of Task Scheduling Algorithms in Heterogeneous Environments

Hadi Yazdanpanah
Computer Student,
Department of Computer,
Iaub, Bushehr, Iran

Amin Shouraki
Computer Student,
Department of Computer,
Iaub, Bushehr, Iran

Najmeh Jamali
Computer Student,
Department of Computer,
PNU, Assaluyeh, Iran

ABSTRACT

A heterogeneous computing environment is a large-scale distributed data processing environment, it depends on some extent parameters on the application and that classified in three main categories such as the hardware, the communication layer, and the software. A computer system consists of hardware and software from two or more different manufacturers. Scheduling is one of the important factors in the heterogeneous environment and the aim of task scheduling in the processing environment is to move computation towards data. In order to achieve improve performance, increase the throughput and minimizing the makespan; scheduler must avoid unnecessary data transmission. Hence, different scheduling algorithms for heterogeneous computing environment are necessary to provide good performance. How to speedup scheduling the service resources to achieve the lowest cost becomes more and more important. This paper tries to illustrate and analyze the overview of eighteen different scheduling algorithms for heterogeneous computing environment and their scheduling issues and problems.

Keywords

Task Scheduling Algorithm, Heterogeneous Environment, Heuristic Algorithm, Directed Acyclic Graph, MapReduce.

1. INTRODUCTION

In heterogeneous environment, hardware and system software are different from various vendors. Organizations often use computers, operating systems and databases from a variety of vendors. Contrast with homogeneous environment in a heterogeneous environment, some tasks run faster on a particular node than others. Heterogeneous system consists of different system resources, which can utilized for performing computationally intensive applications. The executing parallel applications on heterogeneous systems for achieving efficiency performance depend on the methods used to schedule the tasks of a parallel application. In heterogeneous computing systems use different kind of processor and various operation system. Also multi-core systems that can obtain better performance by adding cores and through incorporating specialized processing capabilities to handle particular tasks. Heterogeneous System Architecture (HSA) systems can utilize multiple processor such as CPUs and GPUs, to obtain the better benefits: GPU processing, apart from its well-known 3D graphics rendering capabilities, can also perform mathematically intensive computations on very large data sets, while CPUs can run the operating system and perform traditional serial tasks. Multicore processing had some challenges. The extra cores and cache memory can gain increase cost such as processor size and high power consumption. Furthermore, the multi-core idea is important case for developments GPUs. GPUs have vector processing capabilities that enable them to perform parallel operations on

very large sets of data and to do it at much lower power consumption relative to the serial processing of similar data sets on CPUs. In addition, GPUs can drive capabilities such as incredibly realistic, multiple display stereoscopic gaming. And while their value was initially derived from the ability to improve 3D graphics performance by offloading graphics from the CPU, they became increasingly attractive for more general purposes, such as addressing data parallel programming tasks.

The objective of task scheduling in different environment is to move computation towards data [1]. In order to achieve a good performance in Heterogeneous System, Scheduling is one of the important factors that needful to consideration [2]. The main goal of scheduling is minimizing the overall completion time or makespan. In many systems, the task scheduling problem in heterogeneous environment is preferable to homogeneous computing systems because of the different execution rates among processors and possibly different communication rates among different processors [3]. The task scheduling problem divided into two categories, Static Scheduling and Dynamic Scheduling. In the Static scheduling, initiative information and the relationship between all tasks such as execution and communication costs for each task are known previously; in the dynamic scheduling, decisions scheduling are made at runtime. The process of scheduling parallel tasks determines the priority of task execution and the processor to which each task is assigned. Typically, an optimal schedule is achieved by minimizing the completion time of the last task. In homogeneous and heterogeneous environments, finding the optimal scheduling has been known as an NP-complete problem [4]. However, many heuristics approach are proposed to find a feasible solution within a suitable time and also some heuristics method can gain optimum scheduling. To addressed this issue, the authors in [5] try to consider and comparison different MapReduce scheduling methods in various environment, especially for heterogeneous environment for finding the suitable scheduling solution. MapReduce is a framework that allows to users quickly develops big-data applications and process big-data effectively. Also it had been implemented in multi environments, such as Mar [6], Phoenix [7], and Hadoop [5, 8].

This paper is organized as follows: Section 2 provides a background on different heuristic Mechanisms such as HEFT, CPOP, PETS, DAG and GA. Sections 3 introduce and describe the Scheduling Algorithms in heterogeneous environments and also represent in the form of table. In Section 4 conclude the paper.

2. BACKGROUND

This section briefly tries to describe and introduce different heuristic algorithms such as HEFT, CPOP, PETS and also other related methods such as Directed Acyclic Graph (DAG)

and Genetic algorithms (GA). These methods can achieve better scheduling and some methods in this paper are creating and development base on above algorithms. At the follow this paper tries to consider these algorithms how works.

2.1 HEFT Algorithm

Heterogeneous Earliest Finish Time (HEFT) is a heuristic scheduling algorithm and that is consists of two phases [9]. Tasks prioritizing and assigning tasks to workers. In inputs, HEFT takes a set of tasks and a set of workers, the times to execute each task on each worker, and the times to communicate the results from each job to each of its children between each pair of workers. It descends from list scheduling algorithms (make an ordered list of processes by assigning them base on priorities). In first phase, this method tries to prioritizing the tasks base on upwards rank. The upward rank is meant to represent the expected distance of any task from the end of the computation. The second phase, assigned each task to workers base on highest priority to the processor [10].

2.2 CPOP Algorithm

The Critical Path on Processor (CPOP) algorithm include of two phases [9]. In the first phase, algorithm assigns priority to the tasks base on upward rank and downward rank. CPOP algorithm use a different mapping method for Critical and non-Critical Path to compute upward rank. The downward rank is the critical path length from the begin task to the task, exception the execution time of the task. In the second phase, the algorithm used an approach like HEFT algorithm for assign the tasks to the processor except for the critical path tasks which are mapping to the processor for minimization the execution time of all critical path tasks [11].

2.3 PETS Algorithm

Performance Effective Task Scheduling (PETS) algorithm execute in three phases [12]. In the first phase, algorithms try to grouping each task at each level for parallel executing. Also the second phase, assigns priority to each task based on its rank. In the third phase, the PETS used the similar strategy like HEFT algorithm for executing the task [9].

2.4 Directed Acyclic Graph

Nowadays in mathematics and computer science DAG is used as an important key to finding the proper solution for solving the available challenges. A directed acyclic graph (DAG) is a directed graph with no directed cycles. An application program is represented by a Directed Acyclic Graph (DAG) , it is formed by a collection of vertices and directed edges, each edge connecting one vertex to another, $G=(V,E)$, where V is the set of v tasks nodes, and E is the set of e directed communication edges between the tasks [13]. DAGs can use to modeling many different kinds of information. The reachability relation in a DAG represented as two forms, a partial order and any finite partial. A collection of tasks that must be ordered into a sequence, certain tasks which have constraints must be performed earlier than others, hereby, a vertex for each task and an edge for each constraint; Also DAGs be able represent events of systems or potential events and the causal relationships between them. Furthermore, DAGs be used as a model processes in which data flows in a consistent direction through a network of processors [14].

2.5 Genetic Algorithms

Genetic Algorithms (GAs) are adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics [15]. As such they represent an intelligent exploitation of a random search used to solve very well on

mixed combinatorial problems and achieved optimization solution. Although randomized, GAs are by no means random, instead they exploit historical information to direct the search into the region of better performance within the search space. The basic techniques of the GAs are designed to simulate processes in natural systems necessary for evolution. GAs performs the search process in four stages: initialization, selection, crossover, and mutation to evolve the solutions in order to find the best one(s). In initialization, the initial population is generated randomly and allowing the entire range of possible solutions because population size depends on the nature of the problem. Furthermore Genetic algorithm is a powerful, robust optimization technique and it is able to explore and exploit simultaneously, growing amount of theoretical justification and successful application to real-world problems strengthens [16].

3. SCHEDULING ALGORITHMS

Suitable scheduling is one of the most important issues for processing data in various environments. There are many scheduling algorithms to address these issues with different techniques and approaches.

3.1 Delay scheduling algorithm based on history time

In heterogeneous environment, the Fair scheduler usually can't assign job to suitable slot, and make the performance of Hadoop degraded. In order to solve this problem, in [17] the authors proposed a new scheduling algorithm for multi-user Hadoop cluster which make an optimization based on Fair scheduler for heterogeneous. This method by taking the history time of the completed tasks and the Delay scheduler's strategy into the algorithm, it be able achieve good performance to assign job on its preferred resources, while guaranteeing fairness in shared heterogeneous environment. In this algorithm each user can submit many jobs. All the jobs submitted by the same user shared the slots that assign to the user, and can be scheduling by FIFO [18] or Fair [19], which is decided by the user. If the chosen job's task never have ran on the free slot, the scheduler will let the job launch a task on this slot immediately. Once a job to be scheduled, the scheduler judges whether the job's wait time exceed this threshold firstly, if the wait time longer than the threshold, launch the job's task on the free slot, else skip the job. Also more jobs can assign to suitable slot, and then increase the throughput of the system.

3.2 Using ACO for Task Scheduling

A basic task scheduling challenge in heterogeneous environment is the availability of resources. Especially when processors operate at different speeds and are not continuously available for computation, achieving a better make-span is an important case. Ding et al. [20], offered a new approach that tries to further optimize this scheduling strategy by using Ant Colony Optimization. The proposed technique using Ant Colony Optimization (ACO) is clearly a better solution for reducing the make-span than other popular scheduling algorithms such as SSAC and Min-Min. These methods try to creating trade-off between improving system availability and the response time of submitted tasks.

3.3 SDBATS: A Novel Algorithm for Task Scheduling

A heterogeneous computing system (HCS) efficiently utilizes the heterogeneity of diverse computational resources interconnected with high speed networks to execute a group of compute intensive tasks and Task prioritization it has a major criterion for achieving high performance in HCS.

Munir et al. [21], stated a SD-Based Algorithm Task Scheduling (SDBATS) for scheduling precedence constrained task graphs, which uses the standard deviation of the expected execution time of a given task on the available resources in the heterogeneous computing system as a key factor assigning task priority for achieving high quality schedules with minimum makespan. This new approach takes into account the task heterogeneity and achieves a significant reduction in the overall execution time of a given application. Also the authors calculate performance of the proposed algorithm has been extensively under a variety of conditions on standard task graphs from Graph Partition Archive as well as on some real world application DAGs such as Gaussian Elimination and Fast Fourier Transformation application DAGs. The proposed algorithm consists of two phases, a task prioritizing phase, which arranges the tasks to schedule on the basis of some predefined criterion and a processor selection phase is a combination of task duplication and list scheduling techniques, which uses the same mechanism as adopted by Heterogeneous Earliest Finish Time (HEFT) insertion based policy. The HEFT algorithm uses a recursive procedure to compute the rank of a task by traversing the graph upwards from the exit task and vice-versa for Critical Path on a Processor (CPOP). The proposed algorithm (SDBATS) improve performance over four well known DAG scheduling algorithms such as PETS, HEFT, CPOP and DLS in terms of schedule length (makespan) and speed up.

3.4 Slack based scheduling for dependent tasks

Sirisha [22], presented a novel heuristic approach based on slack namely, Slack based Task Scheduling (STS) and it can achieve High Performance on Heterogeneous Distributed Computing Systems. The problem of task scheduling in Heterogeneous Distributed Computing Systems has been proved to be NP-complete and heuristics are generally used to obtain near optimal solution. The proposed heuristic method classified into two types: critical and non-critical tasks. Critical tasks are the tasks on the critical path. A critical path of a DAG is the longest path from the entry task to the exit task. The critical tasks cannot be delayed and delaying the non-critical tasks by their slack does not affect the schedule length. Also, first generates the schedule by ordering the execution of the tasks. Then the algorithm shortens the schedule length by inserting the tasks into the slack that is created by delaying some tasks. This method consists of four phase such as Level Classification Phase, Task Prioritization Phase, Task Adjustment Phase, and Processor Selection Phase. In Level Classification phase, the tasks in the graph are classified into levels and the tasks in each level are grouped so that they can be executed in parallel. In Task Prioritization phase, at each level, priority is assigned to each task. In Task Adjustment Phase, this phase determines the tasks that can be delayed without exceeding the schedule length. In Processor Selection Phase, the best processor for executing the task selected. By efficiently utilizing the slack of a task, shorter span schedules can be generated and performance can also be enhanced. That STS algorithm's performance is better than well-known DAG scheduling such as PETS and HEFT algorithms in terms of schedule length ratio, speedup and efficiency.

3.5 Data-Intensive Workflow Optimization based on Application Task Graph Partitioning

Optimization of metrics performance is a major issue in heterogeneous computing environment due to the difference

in the computing capacity of execution nodes and variations in the data transfer capability of communication links between these nodes, for solving this challenge, Ahmad et al. [23], proposed a novel method for heterogeneous computing systems namely; dual objective Partitioning based Data-intensive Workflow optimization Algorithm (PDWA) for optimizes the data-intensive workflow applications. In stream-based applications, throughput and latency are two main metrics to measure the performance of an application execution. The proposed Algorithm (PDWA) provides low latency schedules and increases the throughput. For a given data-intensive application, modeled as a DAG, $G(V,E)$, PDWA minimizes the latency with reasonable throughput for a stream-based data processing model. In the proposed algorithm, the application task graph is partitioned such that the inter-partition data movement is minimal. Such optimized partitioning enhances the throughput. Each partition is mapped to the execution node that gives minimum execution time for that particular partition. PDWA also exploits partial task duplication to reduce the latency. The proposed algorithm with workflows from the real-world workloads is able 60% reduced latency with 47% improvement in the throughput as compared to the approach when workflows are not partitioned.

3.6 A Multi-Objective Task Scheduling

Most of algorithms are applicable for single-cloud environment. Panda and Jana [24], proposed a multi-objective task scheduling algorithm for heterogeneous multi-cloud environment for solving these issues such as minimize the overall completion time (i.e., makespan) and minimization of the overall cost of the service. The algorithm has two phases. In the first phase, the algorithm goes through the process of normalization to scale the values between 0 and 1 and for minimization of the makespan used cloud min-min scheduling (execution time based task scheduling (ETBTS)) algorithm. In the second phase, the authors introduced TEMP matrix, it is processed using Min-Min algorithm and the TEMP matrix is defined to choose the best task-cloud pair repetitively. The proposed algorithm tries to perform balances between the makespan and total cost in contrast to other existing task scheduling algorithms in terms of various performance metrics including makespan, total cost and average cloud utilization.

3.7 New Heuristic for Minimizing Schedule Length

Sirisha and Kumari [25], proposed a new heuristic approach namely Minimizing Schedule Length (MSL) for list based task scheduling in Heterogeneous Computing Systems (HCS). The proposed algorithm (MSL) consists of two phases namely, the task ordering phase and task mapping phase. At the first phase, it defines for assigns the tasks and then arranges them in the order of their priority preserving their precedence constraints. Priorities to the tasks can be assigned based on their rank values. For ranking of a task, authors using two attributes namely mean execution time (MET) and total communication time (TCT). Second phase determines the best processor for executing the task. This method Scheduled the independent tasks with shorter span are achieved by reducing the start time. The proposed scheduling algorithm (MSL) can decrease length of scheduling than PETS, HEFT and CPOP algorithms. Furthermore, MSL be able improve performance and parallelization factor values.

3.8 Parameter Dynamic-tuning scheduling algorithm based on history

In MapReduce, the straggler tasks, cause prolong the job execution time in heterogeneity environments. The LATE scheduler strategy [26], but it also has some challenge such as inaccurate estimated time and the wasting of system resources. In order to solve these problems, Zhao et al. [27], proposed two main algorithms: at the first, proposed the parameter dynamic-tuning algorithm based history, This algorithm works is classified into three stages: (1) read each of Map and Reduce phase values of specific jobs from history (The system looks up the history on the corresponding nodes according to the task's submission type. If found, then each worker reads into the values on the corresponding nodes of each stage, If not, then each worker reads into the default value of each stage from history, in which the default value is the average of all the types of jobs have already worked), (2) tune the deviation of proportional factor (The aim of this stage is to evaluate task as accurately as possible), (3) update the history after the task is completed. It can estimates progress of a task accurately and also dynamically tunes the weight of each phase of a map task and a reduce task according to the historical values of the weights. At The end, proposed the evaluation-scheduling algorithm reduce the wasting of system resources by evaluating the free slot before launching a straggler task on this node. The algorithms can significantly reduce the wasting of system resources.

3.9 Recursive Algorithm for Scheduling of Tasks

Optimal scheduling of parallel tasks is important case for achieving high performance in heterogeneous computing system, for solve this problem, Kang and Lin [28], proposed a recursive task scheduling algorithm for a bounded number of heterogeneous processors run on the network of Heterogeneous distributed systems for achieving high performance and fast scheduling time. The propose algorithm consist of three-phase. First phase is task-prioritizing, it is able to compute the upward rank of each task and assign the priority to all tasks. Second phase is processor selection; it is able to schedule the tasks onto the processors that give the latest start time for the task. At the end, moving phase is to move all the possible tasks until the starting time of the entry task is zero. Also the proposed algorithm increase performance on the long scheduling, and is more effectively with the graph structure is more flexible.

3.10 Energy-aware Task Scheduling

Mei and Li [29], proposed a method namely, Energy Aware scheduling algorithm by Minimizing Duplication (EAMD). It is able considers the energy consumption likes the makespan of applications and the propose methods can determine and delete the huge task copies in the schedules generated by duplication-based algorithms, which is easier to operate than Dynamic Voltage/Frequency Scaling (DVFS) and created without overhead of time and energy consumption. The objective of propose algorithm, it can try to reduce large amount of energy consumption while having the same makespan compared with duplication-based algorithms without energy awareness. The proposed algorithm (EAMD) can save up to 15.59% energy consumption for the existed duplication-based algorithms.

When a task comes to the cluster, at first, the cluster will find a server and schedule the task to the server to run it. However, the task will increase the utilization of server and the cluster, and also add power consumption to the server and the cluster.

Zhang et al. [30], offered an energy-aware task scheduling method based on greedy algorithm for achieving local optimal such as reduce energy consumption and increasing performance in a heterogeneous data center to solve the task scheduling problem in a heterogeneous data center because servers in data centers are usually heterogeneous. The proposed algorithm can achieve a good energy savings and good performance in comparison other similar algorithms.

3.11 Application-level Priority Scheduling

Chen et al. [31], presented a method namely, application-level priority scheduling algorithm (APS). The proposed algorithm consists of three stages. At the first, this method can improve utilization knowledge of applications to assign different jobs to heterogeneous resources for minimization the Flow time of the completed jobs. The second stage, the algorithm will dynamically adjust the priority according to the jobs, workload and deadline. Therefore, jobs with higher priority and longer waiting time will be scheduled properly to meet the specific deadline. The third stage is task-bundling for multiple users in MTC environment. The application-level priority scheduling approach can achieve better performance in Makespan and Flow time, and also gain better turnaround time of jobs in multiple Many-task Computing environment (MTC), versus other scheduling algorithms.

3.12 Data-Distribution Technique

The traditional Hadoop system assigns the same set of data to each Datanode, which may not be fruitful for the heterogeneous Hadoop cluster environments. Ubarhande et al. [32], proposed a novel approach to improve the overall performance of the heterogeneous Hadoop cluster to avoid the extra overheads of data transfer by assigning the data blocks on the basis of the processing speed of each Datanode. The Data Distribution method consists of the Speed Analyzer and Data-Distribution components. The Data-Distribution component has been implementing inside the Namenode under Hadoop Distributed File System (HDFS). The Speed Analyzer component measures the processing speed of each Datanode based on computing ratio. The Data-Distribution component is integrated as part of the Hadoop software framework and it assigns data blocks to Datanodes based on their computing ratio. The proposed method can improve performance in the heterogeneous Hadoop clusters when the data size is increased.

3.13 Tasks Security Scheduling Based on DPSO

Zhang et al. [33], offered task scheduling approach namely, Discrete particle swarm optimization (DPSO) in heterogeneous grid for solving the task scheduling problems such as combinatorial optimization, security scheduling. The propose method for optimizing the particle swarm, offer a new natural number coding discrete particle swarm algorithm for solving some issue such as particle position, speed and operation rules. Also DPSO can convergence better and quick finds the optimal solution for problem. Furthermore, for testing this algorithm in the grid, the authors try to comparison the propose algorithm with Genetic algorithm. As the result, discrete particle swarm optimization algorithm can optimize performance better than GA algorithm for independent tasks scheduling. Also DPSO is able to better accordance to the dynamic characteristics than GA algorithm in grid environment. DPSO algorithm for full use of high parallel search efficiency of the particle swarm algorithm can performed with less iteration than the GA algorithm.

3.14 Meta-Heuristic Hybrid dynamic task scheduling

In [34], the authors presented a novel hybrid approach using genetic algorithms (GAs) namely, Hybrid Genetic Algorithm Case based reasoning (HGAC) for heterogeneous Computing environment by predicting the performance of online dynamic resource system to increase the overall effectiveness and improve decision faster for improve the makespan. Genetic algorithms and Case Based Reasoning (CBR) are suitable intelligent approaches for solving the problem of dynamic task scheduling. Genetic algorithms (GAs) can assign the important attributes for case-based retrieving and also it is effective to extract of knowledge for case based retrievals such as job information, machine information, and job run time. Furthermore, this approach can significantly increase overall prediction accuracy rate and robust optimal task scheduling.

3.15 Energy Efficient Scheduling

Ding et al. [35], proposed a heuristic task scheduling algorithm based on the copy namely DTSV (Duplication-based task scheduling algorithm for AOV network) to improve the performance of heterogeneous system such as energy saving. It performs on heterogeneous parallel applications on the cluster and also the propose method can reduce scheduling length and energy consumption at the same time. The DTSV algorithm consists of two basic phase, the initial allocation of tasks and task duplication. First phase for the initial tasks allocation, based on sort of task set and second phase consider the performance and energy balance of task duplication. The propose approach can achieve better performance than Balanced Energy Aware Task Allocation (BEATA) algorithm in heterogeneous system.

3.16 Heuristic Algorithm for Scheduling Out-Tree Task Graphs

The authors [36], offered a heuristic greedy algorithm based on list and task duplication for scheduling Out-Tree task graphs for the problem of scheduling in heterogeneous computing systems (HCS). The propose method for improve the performance scheduling, attempts to discover the best point between load balancing and reducing the schedule time without increasing the time complexity of the algorithm. The goals of this algorithm are provided good quality for scheduling and saving the cost such as decrease scheduling times and number of using processors. Also it can scheduled tasks base on task duplication scheduling method for assigning task slot on proper processor and then assign the task that have biggest computation cost to the fastest available processor for achieving the aims that described in above.

Briefly the proposed algorithm can achieve lessen schedule length with using minimum amount of processors.

3.17 Independent task scheduling via makespan refinery approach

Bey et al. [37], presented an approach near optimal for solving the independent task scheduling problems in heterogeneous distributed computing. The propose scheduler attempts to present a solution in two phase. In first phase, that uses the task reassignment algorithm in initial of the scheduling for minimization of total completion time. Also in the second phase, authors for solving the task scheduling issue, that use Makespan Refinery Approach (MRA) to lessen the maximum completion time. The propose scheduler method can decrease completion time of a set of tasks and improve the response time when assigning the resources to tasks.

3.18 AuctionNet: Market Oriented Task Scheduling

The authors [38] offered a model for solving computationally large-scale distributed systems problems. In heterogeneous distributed environments some challenging exists such as various heterogeneity and dynamism in systems, applications, and local policies, efficient and fair task scheduling. For solve these challenges, market-oriented task scheduling mechanism can use in a distributed environment. Furthermore, the authors combined approach leveraging the advantage of both hyper graph theory and incentive mechanisms. At the first, the authors propose a general framework of market-oriented task scheduling in distributed systems and then presented two algorithms for task-bundle scheduling. The proposed approach has been following aspects. (1) Product-bundle has been a common marketing strategy in the emerging commercial clouds and desktop clouds; task-bundle could be a useful concept for computing and storage markets. (2) This algorithm combining hyper graph and incentive mechanisms in distributed systems for achieving multi-objective optimization. (3) This algorithms ensure efficiency, fairness and maximize social welfare. At the end, provides holistic mechanisms and policies to ensure efficient and fair resource sharing and task scheduling in distributed systems.

In a heterogeneous environment each node has different computing power. Scheduling approaches are expressed in the Table 1. Objective column tries to identify aims of Algorithms.

Table 1. Task Scheduling approaches with performance considerations

Algorithm	Objectives	Method	Observations
Delay scheduling algorithm based on history time	<ul style="list-style-type: none"> Improve performance. Increase the throughput. 	<ul style="list-style-type: none"> Optimize scheduling based on Fair and Delay scheduler. 	<ul style="list-style-type: none"> Assign jobs to suitable slot for increasing the throughput.
Using ACO for Task Scheduling	<ul style="list-style-type: none"> Achieving a better makespan. 	<ul style="list-style-type: none"> Optimize scheduling base on Ant Colony. 	<ul style="list-style-type: none"> Reducing the makespan than other similar scheduling. Trade-off between system availability and the response time.

SDBATS	<ul style="list-style-type: none"> Minimizes the makespan and speed up. Improve performance. 	<ul style="list-style-type: none"> Uses the standard deviation for given task on the available resources. 	<ul style="list-style-type: none"> Reducing the overall execution time. Scheduling tasks with minimum makespan.
Slack based scheduling for dependent tasks (STS)	<ul style="list-style-type: none"> Enhanced the performance. 	<ul style="list-style-type: none"> Scheduling Task based on Slack. 	<ul style="list-style-type: none"> Shortens the scheduling length by insert the tasks into the slack.
Data-Intensive Workflow Optimization based on Application Task Graph Partitioning (PDWA)	<ul style="list-style-type: none"> Optimizing the data-intensive workflow. Increase the throughput. 	<ul style="list-style-type: none"> Partitioning based on Data-intensive Workflow. 	<ul style="list-style-type: none"> Reduced latency with improvement in the throughput.
Multi-Objective Task Scheduling	<ul style="list-style-type: none"> Minimize the makespan. Minimization of the overall cost of the service. 	<ul style="list-style-type: none"> Used cloud min-min scheduling for minimization of the makespan. 	<ul style="list-style-type: none"> Create balances between the makespan and total cost in contrast to other existing task scheduling algorithms.
New Heuristic for Minimizing Schedule Length (MSL)	<ul style="list-style-type: none"> Minimize the Scheduling Length. 	<ul style="list-style-type: none"> Assigns the tasks base on priority. 	<ul style="list-style-type: none"> Improve performance and parallelization. Scheduled the independent tasks for reducing the start time.
Parameter Dynamic-tuning scheduling algorithm based on history	<ul style="list-style-type: none"> Reduce the wasting of system resources. 	<ul style="list-style-type: none"> Read each Map and Reduce values of specific jobs from history. Tune the deviation of proportional factor. Update the history after the task is completed. 	<ul style="list-style-type: none"> Reduce the wasting of system resources by evaluating the free slot before launching a straggler task on this node.
Recursive Algorithm for Scheduling of Tasks	<ul style="list-style-type: none"> Increase performance. Decrease scheduling time. 	<ul style="list-style-type: none"> Task prioritizing. Processor selection. Moving. 	<ul style="list-style-type: none"> Algorithms can achieve high performance and fast scheduling time for a bounded number of heterogeneous processors on Heterogeneous distributed systems.
Energy-aware Task Scheduling	<ul style="list-style-type: none"> Reduce energy consumption. Increase the performance. 	<ul style="list-style-type: none"> Scheduling by deleting the duplication tasks. Scheduling based on greedy algorithm. 	<ul style="list-style-type: none"> Algorithms can save energy consumption and improve performance in heterogeneous environment.
Application-level Priority Scheduling (APS)	<ul style="list-style-type: none"> Achieving better performance in Makespan and Flow time. 	<ul style="list-style-type: none"> Assigning jobs to resources. Priorities the jobs. Task-bundling for multiple users. 	<ul style="list-style-type: none"> This approach can achieve better Makespan and Flow time, and also better turnaround time of jobs in multiple Many-task Computing environments.
Data-Distribution Technique	<ul style="list-style-type: none"> Improve the overall performance. 	<ul style="list-style-type: none"> Avoid the extra overheads of data transfer by assigning the data blocks on each Datanode. 	<ul style="list-style-type: none"> This method can improve performance in the heterogeneous Hadoop clusters when the data size is increased.
Tasks Security Scheduling (DPSO)	<ul style="list-style-type: none"> Solving the combinatorial optimization and security scheduling. 	<ul style="list-style-type: none"> Discrete particle swarm optimization algorithm for full use of high parallel search efficiency offer a new natural number coding discrete particle swarm algorithm. 	<ul style="list-style-type: none"> The method can solve the particle position, speed and operation rules. Optimizing the performance better than GA algorithm.

Meta-Heuristic Hybrid dynamic task scheduling (HGAC)	<ul style="list-style-type: none"> • Optimal task scheduling. • Improve the makespan. 	<ul style="list-style-type: none"> • Hybrid approach base on genetic algorithms and case based reasoning. 	<ul style="list-style-type: none"> • This algorithms can predict the performance of online dynamic resource system to increase the overall effectiveness and improve decision faster for improve the makespan.
Energy Efficient Scheduling	<ul style="list-style-type: none"> • Improve the performance. • Energy saving. 	<ul style="list-style-type: none"> • Sorting the task. • Consider the performance and energy balance of task duplication. 	<ul style="list-style-type: none"> • This approach can achieve better performance than similar algorithm in heterogeneous system.
Heuristic Algorithm for Scheduling Out-Tree Task Graphs	<ul style="list-style-type: none"> • Improve the scheduling performance • Saving the cost. 	<ul style="list-style-type: none"> • Greedy algorithm based on list and task duplication for scheduling Out-Tree task graphs. 	<ul style="list-style-type: none"> • Algorithm can achieve lessen schedule length with using minimum amount of processors.
Independent task scheduling via makespan refinery approach	<ul style="list-style-type: none"> • Minimization the makespan. 	<ul style="list-style-type: none"> • Task reassignment. • Makespan Refinery Approach (MRA) to decrease the maximum completion time. 	<ul style="list-style-type: none"> • This method can decrease completion time of a set of tasks and improve the response time when assigning the resources to tasks.
Market Oriented Task Scheduling (AuctionNet)	<ul style="list-style-type: none"> • Solving computationally large-scale distributed systems problems. 	<ul style="list-style-type: none"> • Combined both of approach hyper graph theory and incentive mechanisms. 	<ul style="list-style-type: none"> • Provides fair resource sharing and task scheduling. • Combining hyper graph and incentive mechanisms for achieving multi-objective optimization.

4. CONCLUSION

This paper attempted to comparison and analyzed eighteen different scheduling algorithms in heterogeneous environment. The Heterogeneous environments in contrast with homogeneous environments are consists of different hardware, communication layer and software. Chose the suitable scheduling algorithm is the most critical factors for achieves the efficient tasks scheduling, improve the performance and minimizing the overall completion time or makespan in heterogeneous systems. Number of task scheduling algorithms represented for high performance computing and near optimal solution. Furthermore, many heuristics approach based on DAG scheduling and GA are proposed in this paper and that tries to find a feasible solution within a suitable time for obtaining the optimal scheduling and maximize the resource utilization. Furthermore, the future work can do improving the efficient task scheduling algorithms for achieving more performance and optimal solution on heterogeneous environment by using the heuristic methods for minimizing the makespan and that scheduling algorithms can achieve high efficiency and energy saving.

5. REFERENCES

[1] Kondikoppa , P., Chiu, C. H., Cui, C., Xue, L. and Park, S. J. (2012).Network-Aware Scheduling of MapReduce Framework on Distributed Clusters over High Speed Networks, Workshop on Cloud Services, Federation, and the 8th Open Cirrus Summit, San Jose, CA, USA, September 21.

[2] He, Y., Liu, J., & Sun, H. (2011).Scheduling Functionally Heterogeneous Systems with Utilization

Balancing. IEEE International Parallel and Distributed Processing Symposium (pp. 1187 – 1198).

[3] Hai, Y., sheng, S., & Lian, X. (2007).A New Dynamic Scheduling Algorithm for Real Time Heterogeneous Multi Processor Systems. Workshop on Intelligent Information Technology Application (pp.112 – 115).

[4] Gary, M. R., & Johnson, D. S., (1979).Computers and Intractability: A Guide to the theory of NP-Completeness, W.H. Freeman and Co., San Francisco, CA.

[5] Yazdanpanah, H., Shouraki, A., Abshirini, A. A. (2015). A Comprehensive View of MapReduce Aware Scheduling Algorithms in Cloud Environments. International Journal of Computer Applications, Vol.127, No.6, (pp. 10-15).

[6] He, B., Luo, Q., & Govindaraju, N. K. (2011).Mars: accelerating MapReduce with graphics processors. IEEE Trans. Parallel Distribute System, Vol.22, Issue.4, (pp. 608 - 620).

[7] Ranger, C., Raghuraman, R., Penmetsa, A., Bradski, G., & Kozyrakis, C. (2007).Evaluating mapreduce for multi-core and multiprocessor systems. IEEE 13th International Symposium on High Performance Computer Architecture.

[8] The Apache Software Foundation: Hadoop (2015).<http://hadoop.apache.org>. Accessed 1 October 2015.

[9] Illavarasan, E., & Thambidurai, P. (2007) .Low complexity performance effective task scheduling

- algorithm for heterogeneous computing environments. *Computer Science*, Vol.3, Issue.2, (pp. 94-103).
- [10] Topcuoglu, H., Hariri, S., & Wu, M. Y. (2002). Performance effective and low complexity task scheduling for heterogeneous computing. *IEEE Trans Parallel Distributed Systems*, Vol.13, Issue.3, (pp. 260-274).
- [11] Alexander, M., et al. (2011). *Euro-Par 2011: Parallel Processing Workshops. The Critical Path on Processor (CPOP)*. Retrieved from <https://books.google.com/books?isbn=3642297374>. (p.443), New York, Springer.
- [12] Ilavarasan, E., Thambidurai, P., Mahilmanan, R. (2005). Performance Effective Task Scheduling Algorithm for Heterogeneous Computing System. *The 4th International Symposium on Parallel and Distributed Computing*, (pp. 28-38).
- [13] Amalarethinamand, D. I. G., Mary, G. J. J. (2011). A new DAG based Dynamic Task Scheduling Algorithm (DYTAS) for Multiprocessor Systems, *International Journal of Computer Applications*, Vol. 19, No.8, (pp. 975 – 987).
- [14] Baskiyar, S., & Dickinson, C. (2005). Scheduling directed acyclic task graphs on a bounded set of heterogeneous processors using Task Duplication. *Journal of Parallel and Distributed System*, Vol.65, (pp.911-921).
- [15] Schmitt, L. M. (2001), *Theory of Genetic Algorithms, Theoretical Computer Science*, Vol.259, (pp. 1–61).
- [16] Zomaya, Ward, C., & Macey, B. (1999). Genetic scheduling for Parallel Processor Systems: Comparative Studies and Performance Issues. *IEEE Trans. Parallel and Distributed Systems*, Vol.10, Issue.8, (pp. 795 – 812).
- [17] Ye, B., Dong, X., Zheng, P., Zhu, Z., Liu, Q., & Wang, Z. (2013). A delay scheduling algorithm based on history time in heterogeneous environments. *8th Annual China Grid Conference* (pp. 86-91).
- [18] Hadoop, “Hadoop home page.” <http://hadoop.apache.org/>.
- [19] Hadoop’s Fair Scheduler. https://hadoop.apache.org/docs/r1.2.1/fair_scheduler.
- [20] Ding, L., Fan, P., & Wen, B. (2013). A Task Scheduling Algorithm for Heterogeneous Systems Using ACO. *2nd International Symposium on Instrumentation and Measurement Sensor Network and Automation* (pp. 749-751).
- [21] Munir, E. U., Mohsin, S., Hussain, A., Nisar, M. W., & Ali, S. (2013). SDBATS: A Novel Algorithm for Task Scheduling in Heterogeneous Computing Systems. *IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum* (pp. 43-53).
- [22] Sirisha, D. (2013). Slack based Scheduling for dependent tasks in Heterogeneous Computing environments. *Third International Conference on Computational Intelligence and Information Technology* (pp. 279-286).
- [23] Ahmad, S. G., Liew, C. S., Rafique, M. M., Munir, E. U., & Khan, S. U. (2014). Data-Intensive Workflow Optimization based on Application Task Graph Partitioning in Heterogeneous Computing Systems. *IEEE Fourth International Conference on Big Data and Cloud Computing* (pp. 129-136).
- [24] Panda, S. K., & Jana, P. K. (2015). A Multi-Objective Task Scheduling Algorithm for Heterogeneous Multi-Cloud Environment. *International Conference on Electronic Design, Computer Networks & Automated Verification* (pp.82-87).
- [25] Sirisha, D. & Kumari, G. V. (2015). A New Heuristic for Minimizing Schedule Length in Heterogeneous Computing Systems. *IEEE International Conference on Electrical, Computer and Communication Technologies* (pp. 1 - 7).
- [26] M. Zaharia, A. Konwinski, A. D. Joseph, R. Katz and I. Stoica, "Improving MapReduce performance in heterogeneous environments", In: *OSDI 2008: 8th USENIX Symposium on Operating Systems Design and Implementation*, 2008.
- [27] Zhao, X., Dong, X., Cao, H., Fan, Y., & Zhu, H. (2012). A parameter dynamic-tuning scheduling algorithm based on history in heterogeneous environments. *Seventh China Grid Annual Conference* (pp. 49-56).
- [28] Kang, Y., & Lin, Y. (2011). A Recursive Algorithm for Scheduling of Tasks in a Heterogeneous Distributed Environment. *4th International Conference on Biomedical Engineering and Informatics* (pp. 2099-2103).
- [29] Mei, J., & Li, K. (2012). Energy-Aware Scheduling Algorithm with Duplication on Heterogeneous Computing Systems. *ACM/IEEE 13th International Conference on Grid Computing* (pp. 122-129).
- [30] Zhang, S., Wang, B., Zhao, B., & Tao, J. (2013). An Energy-aware Task Scheduling Algorithm for a Heterogeneous Data Center. *12th IEEE International Conference on Trust Security and Privacy in Computing and Communications* (pp. 1471-1477).
- [31] Chen, S., Zhang, Y., Hu, Z., & Yu, H. (2013). An Application-level Priority Scheduling for Many-Task Computing in Multi-user Heterogeneous environment. *International Conference on High Performance Computing and Simulation* (pp. 558-565).
- [32] Ubarhande, V., Popescu, A. M., & Gonzalez Velez, H. (2015). Novel Data-Distribution Technique for Hadoop in Heterogeneous Cloud Environments. *Ninth International Conference on Complex Intelligent and Software Intensive Systems* (pp. 217-224).
- [33] Zhang, D., Zhu, H., Wang, Y., & Miao, Z. (2010). Tasks Security Scheduling Based on DPSO in Heterogeneous Grid Environment. *Second International Conference on Networks Security Wireless Communications and Trusted Computing* (pp. 143-148).
- [34] Sri, R. L., & Balaji, N. (2013). Meta-Heuristic Hybrid dynamic task scheduling in heterogeneous Computing environment. *International Conference on Computer Communication and Informatics* (pp. 1 - 6).
- [35] Ding, L., Fan, P., Zhao, X., & Wen, B. (2013). Energy Efficient Scheduling Algorithm in Heterogeneous Environment. *2nd International Symposium on Instrumentation and Measurement Sensor Network and Automation* (pp. 909-913).

- [36] Zhang, J., Kuang, W., & Yuan, H. (2012). A Heuristic Algorithm for Scheduling Out-Tree Task Graphs in Heterogeneous Computing Systems. IEEE fifth International Conference on Advanced Computational Intelligence (pp. 123-128).
- [37] Bey, K. B., Benhammadi, F., Mokhtari, A., & Guessoum, Z. (2010). Independent task scheduling in heterogeneous environment via makespan refinery approach. International Conference on Machine and Web Intelligence (pp. 211-217).
- [38] Zhao, H. and Li, X. (2010). AuctionNet: Market Oriented Task Scheduling in Heterogeneous Distributed Environments. IEEE International Symposium on Parallel & Distributed Processing Workshops and PhD Forum (pp. 1-4).