WORKFLOW SCHEDULING IN HETEROGENEOUS COMPUTING SYSTEMS : A SURVEY

Ashish Gupta
Department of Computer Engineering
National Institute of Technology
Kurukshetra, Haryana, India
guptaashish181@gmail.com

Ritu Garg
Department of Computer Engineering
National Institute of Technology
Kurukshetra, Haryana, India
ritu.59@gmail.com

Abstract—**In the world of managing and processing large data sets significantly and executing such complex applications on heterogeneous systems, efficient scheduling is of prime concern. The workflow scheduling deals with mapping of tasks on the available resources while maintaining the precedence constraints and guaranteeing Quality-of-Service requirements and fairness in execution of workflow. Numerous workflow scheduling schemes have been presented by authors in the past. In this paper, we are providing the comprehensive survey of workflow scheduling techniques by classifying them based upon the workflow structures, scheduling criteria's, scheduling strategies in heterogeneous systems. We classified workflow-scheduling structures as DAG and Non-DAG structures, scheduling criteria's on the basis of performance driven, market driven and trust driven, workflow-scheduling strategies into Heuristic, Meta-Heuristic and Hybrid scheduling. We also considered the scheduling issues in various other fields of engineering; Hadoop Based, Map-Reduce Based, Agent Based. A comparative analysis of each scheduling algorithm is highlighted through their objectives, algorithm proposed, properties and environment in tabular form. The survey results are used to identify workflow scheduling approaches and the areas that need further research.**

Keywords— Workflow scheduling, Task scheduling, Meta-Heuristic, DAG, Heuristic,Heterogeneous computing system

I. INTRODUCTION

The heterogeneous system has evolved as a worldwide infrastructure for future of electronic computing applications by immersing heterogeneous resources. Heterogeneous system includes cloud systems, grid systems, clusters etc. To maintain complex technical experiments, distributed resources like computational devices, scientific instruments and applications need to be organizing while supervising the operation of workflow on heterogeneous systems.

What is Workflow? A workflow symbolizes a sequence of procedures that ease the complexity of execution and management of applications. Workflow technology represents a large set of scientific applications that have to be deploying over heterogeneous systems. A Directed Acyclic Graph (DAG) in which computational tasks resemble as nodes commonly represents workflow and the edge between two tasks represents the precedence and flow constraints. Imposing the workflow applications on Heterogeneous system offers several benefits: 1) Enable to construct the dynamic applications by utilizing heterogeneous resources. 2) A complete resource utilization that occurs in a specific domain to boost up the outcomes and execution costs. 3) Execution intervals various administrator domains to get specific processing functionalities.

The Heterogeneous computing systems (HCs) refers to those systems, which utilizes several processors or cores for computing. It synchronizes the utilization of diverse kinds of machines, interface and networks to increase their combined performance and cost-effectiveness. Many computational intensive problems are effectively solves through heterogeneous system. The complex architecture of machines is required to obtain high performance at peak time, to the extent that a set of tasks can be worked on given planned feature such that computational properties matches the advance capabilities of machines. To help the future researchers in the area of workflow scheduling in heterogeneous system, we surveyed some research papers and analyzed the mechanisms for workflow scheduling in heterogeneous system. Therefore, the ground reason of this paper is to survey the previous techniques regarding scheduling techniques, policies, describe their properties, and clarify their pros and cons. The goals of this survey are as follows:

- Studying the existing workflow scheduling mechanisms.
- Providing a new classification of task scheduling mechanisms.
- Clarifying the scheduling objectives into market based categories, performance based and trust based.
- Studying the categorical scheduling algorithms in Heuristic, Meta- Heuristic and Hybrid approach.
- Study of various field fused with heterogeneous system like networking, Hadoop, map-reduce etc.
- Outlining new researches could be done to improve the task scheduling algorithms.

II. WORKFLOW SCHEDULING TAXONOMY IN HCS.

The author [1] classified scheduling of tasks into local and global task scheduling. The local scheduling is to map or allocate tasks to single resource in different time slots. On the other side, global scheduling is used to do decision-making. As per the definition of workflow scheduling, the global task scheduling, this is concentrated on mapping of inter-dependent tasks as well as managing its execution on set of heterogeneous resources. In this section, we will be discussing workflow-scheduling taxonomy from the view of scheduling structure or pattern and scheduling strategy taking into account user's objectives and QoS constraints, which are very important in process of workflow scheduling.

2.1. Workflow structure

A workflow consists of multiple tasks connected together in such a way that some are dependent on others. The workflow structure\workflow pattern [2][3], indicates the connection among these tasks. Workflow structure is represented with directed acyclic graph (DAG) [4] or non-DAG as shown in Figure 1.

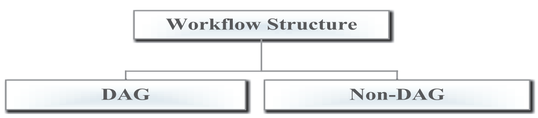

Figure 2: Classification of Workflow Structure

2.1.1 DAG based: It is further subcategorized into sequence, parallelism and choice. Sequence is a well-ordered sequence of tasks where output of a task is input to another task. Parallelism is defined as tasks running concurrently instead of in series or in sequence. In choice control pattern, when appropriate conditions are true then single task is selected to execute while scheduling.

2.1.2 Non-DAG based: In addition to DAG based workflow patterns such as sequence, parallelism and choice, Non-DAG workflow also consists of iteration structure. In iteration structure, some part of workflow tasks repeat in the specific iteration block. The case when one or more tasks repeat frequently in workflow then this structure is used that is also known as loop or cycle. It is quite popular pattern in scientific applications.

2.2 Workflow Scheduling Criteria

In general, workflow scheduling in heterogeneous system is NP-Hard problem. Therefore, several scheduling algorithms had developed in past few years to provide the optimal solution which is close to exact solution. We categories the major scheduling criteria into performance-driven, market-driven and trust-driven as shown in Figure 2.

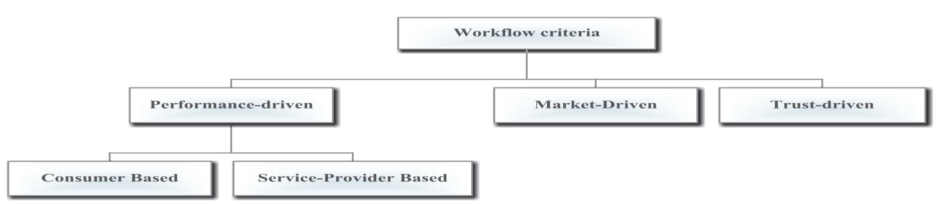

Figure 1 : Workflow Scheduling Strategy Classification

2.2.1 Performance-driven: The Performance-driven criteria help in allocating the workflow tasks onto resources with the aim to attain best execution performance like minimum makespan\execution time or maximum resource utilization. It is sub-categorized into consumer based and service provider based objectives. Consumer-based performance objectives includes user's requirements like makespan\ execution time, economic cost etc. *Makespan* is the overall completion time required to execute the entire application. It is also defined as the time interval from user submitting the task to the complete execution of entire schedule. *Cost* is the economical cost that user's have to pay for executing its applications on fundamental resources including computation cost and cost to transfer the data. Whereas service-provider based objectives are concerned with requirements like load balancing, resource utilization, energy-efficiency etc. *Load balancing* is the process of distributing workloads on computing resources in a distributed computing environment. Imbalance of load over the resources\VMs lets the scheduler to reschedule the workload on the free available resources so that no resource is heavily loaded in comparison to other. Increasing in the *Resource utilization* is profitable and consumes less resources and time, thereby providing benefits to service provider. The resource utilization directly effects the *Energy consumption* of an application. When resources processors are not efficiently utilized, the energy consumption will become high due to the presence of idle processors that consumes the power at same rate.

2.2.2 Market driven: Market-driven criteria represent the market models to handle the resource assignment to process tasks in workflow. Hence, economy workflow mechanism is applied, so that consumer buying service from the resources provider and paying for execution of its application. Market driven criteria consist of resource cost, deadline, budget etc. *Resource Cost* is calculated by cloud service provider on quantity of a cloud resource costs to users for utilization of resources. Many companies provide cloud services to users these days on dollars per hour basis and charge very nominal fee. *Deadline* is the time bound for the entire workflow execution. *Budget* is the estimated cost that has to pay for a workload execution prior to its submission.

2.2.3 Trust-driven : In Trust-driven criteria, schedulers select resources based on their trust factor. It helps in selecting the less malicious host. Trust-driven criteria consist of security, self-defense capabilities, attack history, reliability etc. *Security* is becoming major concern from user's point of view while selecting the cloud services. User is always concerned about that its privacy and security of data should remain conserved, so it is the service provider responsibility to provide security and privacy in complex distributed systems for their users. *Self-defense* capability is also trust-driven approach in heterogeneous environment. *Attack history* is the history of the attack happens on particular resources in past. *Reliability* is the probability of task to complete its execution on given resource successfully. To improve the reliability of the service, several techniques such as active replication and backing up data from resources are applied in scheduling.

2.3 Workflow Scheduling Strategies

Based upon the strategies used for scheduling, in this section, we categorized workflow scheduling into: Heuristic scheduling, Meta-heuristic scheduling and Hybrid scheduling (see the Figure 3).

Example in this group are Modified Critical Path (MCP) and Mobility Directed (MD) [5], Dominant Sequence Clustering (DSC) [6], Linear Clustering Method [7], Cluster based HEFT with Duplication [13].

2.3.1.2 List Scheduling

List scheduling is one of the main heuristic models in scheduling algorithms. Here scheduling works in two phases first is task prioritization phase and other is processor selection. In former phase, the scheduling nodes have been assigned with some priority and these nodes are arranged in a list in descending order of their priority. Nodes with high priority are being executed first then the nodes with low priority as in [8]. If there is any chance of tie, then tiebreaker uses some method given in ref [9]. List scheduling algorithms covered in this paper are Critical Path on a Processor [10], Mapping Heuristic [11], Heterogeneous Earliest Finish Time [10], Triplet [12], and Performance Effective Task Scheduling [14].

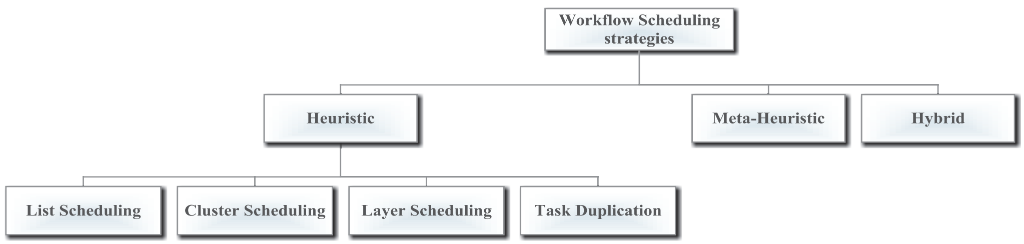

Figure 3 Workflow scheduling approaches Classification

The *heuristic* scheduling strategy provides solution close to optimal solution. It is further divides into four groups: Clustering scheduling, List scheduling, Layer scheduling and Task duplication. *Meta-Heuristics* strategy provides good solutions with less computational effort than simple heuristic solutions. These are approximate, not problem-specific and usually non-deterministic. These algorithms generally include evolutionary algorithms, swarm-intelligence-based, stochastic algorithm, probabilistic based, neural algorithms etc. *Hybrid* scheduling strategy provides the solution with combination of two approach either two meta-heuristic\ heuristic algorithms or combination of both.

2.3.1 Heuristic

2.3.1.1 Clustering scheduling

The clustering is one of the techniques under heuristic scheduling algorithm. Clustering is the process of grouping the set of tasks into clusters. In clustering [6], initially tasks are groups into clusters based on their dependency constraints. Tasks of same cluster are assigns to the single processor with the aim to minimize communication overhead. These clusters can be merges if combination of two or more cluster generates lesser makespan. Number of clusters formed should be approximately equal to number of resources available. Few

2.3.1.3 Layering Scheduling

Layering scheduling is another important sub-category of heuristic scheduling, which divide the nodes of DAG based on layers or level and these layer are inter-dependent on each other. Some of layering scheduling algorithms are LP Longest-Path Layering algorithm [15, 16], Coffman-Graham Algorithm [17], Promote-Layering [18], Double Layer Scheduling algorithm [19], Dynamic level scheduling covered [20]

2.3.1.4 Task Duplication

Task Duplication is the process of mapping the group of tasks to processors redundantly to decrease the inter-process communication overhead. Some of the Duplication algorithms are: Duplication based Scheduling strategy [21-23], Critical Path fast Duplication [24] Duplication Scheduling Heuristic [25], Bottom-Up Top-down Duplication Heuristic [26].

TABLE 1: CLASSIFICATION OF HEURISTIC SCHEDULING ALGORITHMS

Author, Year	Algorithm	Workflow Structure	Objectives	Description	Environment
Wu, Gajsk, (1990) [5]	Modified Critical Path (MCP), Mobility Directed (MD)	Clustering	Execution Time	MCP is used as clustering algorithm when there is unbounded number of processors and also it is reduce to edge-zeroing algorithm similar to Dominant Sequence Cluster as they both implement on earliest starting time. The complexity of algorithm is $O(v^2 \log v)$time.	Heterogeneous System
Yang and Gerasoulis, (1994) [6]	Dominant Sequence Cluster (DSC)	Clustering	Minimizing parallel time	Several clustering algorithm have been merged together with taking care of complexity. Main idea of this algorithm is to find the Dominant Sequence at every level followed by zero-edging in this DS. The complexity is $O((v + e)\log v)$ time.	-
Kim and Browne (1988) [7]	Linear Clustering	Clustering	Length of the Path or Makespan time	Clustering algorithm, also known as the KB/L algorithm At first put all the edged in list of unexamined task and find the longest path by using a cost function, the nodes are cluster and the edges are zeroed	-

				then Mark all those edges which fall to the node in cluster. The complexity is O(v(v+e)).	
Bozdag et al (2006)[13]	Cluster based HEFT with Duplication (CBHD)	Cluster Scheduling	Minimize the execution time	It is combination of HEFT and Triplet with task duplication. Cluster the task and then prioritize those task. Mapping of cluster of tasks to those machine which generated minimum total execution time.	Heterogeneous system
Topcuoglu et al. 1999 [10]	Critical Path On a Processor (CPOP)	List Scheduling	Scheduling length or makespan	The ranking mechanism is used to assign the nodes. critical path node is find out and critical-path-processor CPP will reduce the length of critical path. CPOP algorithm uses ranking mechanism to assigning the priority. Time complexity is $O(v^2P)$	Heterogeneous system
Braun et al. (1998) [11]	Mapping Heuristic (MH)	List Scheduling	Minimizing execution time	In the algorithm Priorities are assigned to task and processor in mechanism of ranking. Next processor starting time is for a task is time which is the end time or completion time of last task of previous processor and in ready state to execute new one. The algorithm do not schedules a task to less loaded slot for tasks. The time complexity of this algorithm us to $O(v^2 \times q^3)$	Heterogeneous system
Topcuoglu et al. (1999) [10]	Heterogeneous Earliest Finish Time (HEFT)	List Scheduling	Minimum finish time	HEFT works on bounded number of processor in its application. Main two phases of this algorithm is Prioritizing the task and a processor selection . Former phases assign the ranking to the task so that one task have higher priority than other and in later phase one of the task is selected based on order set of task, declared in first phase.	Heterogeneous system
C Bertrand et al. (2001) [12]	Triplet	List Scheduling	Execution time	Purpose of the Triplet algorithm in the heterogeneous system is to merged processors into clusters and mapping the clusters of tasks onto clusters processor to reduce overhead of communication Triplet three phases: Task Clustering phase, Processor clustering phase, Mapping phase.	Heterogeneous system
Ilavarasan et al.(2005)[14]	Performance Effective Task Scheduling (PETS)	List Scheduling	Minimize the scheduling length	Construct a priority queue with computed ranks, Meanwhile allot the unscheduled task to processor with which has minimum EFT calculated by insertion based scheduling property. The goal of task scheduling is reduce the length of scheduling or makespan time. The complexity is O (v + e) (p+ log v).	Heterogeneous system
Andreev et al. 2007 [15]	Longest-path Layering algorithm (LP)	Layering scheduling	Minimum layering area. minimizing the width and height	Placing all the sink nodes or vertex in single layer and remaining is placed in Next layer The main advantage of this method is that it has linear time complexity, and it uses minimum no of layer. The disadvantage of the LPL method is that its layering will remain too wide.	-
Coffman and Graham 1972 [17]	Coffman–Graham algorithm (CGA)	Layering scheduling	Minimal length	Layer path layering algorithm It makes sure that a layering at-most m nodes per layer and height of layering may double the optimal height. The time complexity of this algorithm is $O(V^2)$.	-
Sih and E. A. Lee 1993 [20]	Dynamic-Level Scheduling (DLS)	Level Scheduling	Makespan	Algorithm at each select the available processor and ready nodes pair which boost the significance of dynamic level is same as rank minus execution starting time The Time Complexity of DLS algorithm is an $O(v^3 \times q)$.	Heterogeneous system
Lee et al., 1997 [21]	Duplication based State Transition (DST)	Task Duplication	Makespan Time or minimize the schedule length	Duplication-based State Transition (DST) method has two phases Task Manipulation and Task Refinement. “state manipulation” in DST is equal to State Transition operate as a traditional meta-heuristics. In Task refinement (TR) each manipulation state is refined to observe whether there is any wastage duplicated tasks or not and when it found any becomes useless.	Heterogeneous System
Ahmad and Kwok, 1994 [24]	Critical Path fast Duplication (CPFD)	Task Duplication	Makespan	Duplication to schedule CPNs efficiently. The algorithm uses two procedures: Attempt Duplication, and Trace Ancestor. Former store all the unused processor and all processors containing the parent node in a stack.	Heterogeneous System
Kruatrachue and Lewis, 1988 [25]	Duplication Scheduling Heuristic (DSH)	Task Duplication	Makespan	Duplicates immediate predecessor tasks when duplication reduces . In this algorithm, the time slot which is idle take place between finish time slot of happen to be last node and schedule to processor with initial time slot of current , while scheduling although the scale of duplication is very less.	Homogeneous Processors
Chung and Ranka, 1992 [26]	Bottom-Up Top Down Duplication Heuristic (DBUS)	Task Duplication	Makespan Time	BTDH is DSH algorithm extension. Two particular main differences are it do not specify any priority about parent node for duplication and process of duplication won't stop until all idle time slot get filled up Higher complexity and proved better than HLFET and ETF. The complexity of algorithm is $0(n^4)$	Heterogeneous System

2.3.2 Meta Heuristic Scheduling

Recently, meta-heuristics have gained popularity due to its efficiency and effectiveness to solve large and complex problems. Meta-heuristic techniques are to attain close to exact solution or near optimal solutions for NP-Hard scheduling problems. It provides good solutions with less computational effort than simple heuristic methods. They perform better and converge faster to optimal solution for large-scale input data. Meta-heuristic techniques are namely Ant Colony Optimization (ACO) [30, 31], Genetic Algorithm (GA) [32-35], Particle

Swarm Optimization (PSO) [27, 28], Cuckoo Optimization Algorithm (COA) [36, 37], Honey Bee (HBB) [38], LBMPSO [29] etc.

2.3.3 Hybridized Heuristic and Meta- Heuristic Based Scheduling

In Hybrid scheduling, two or more heuristics or Meta-Heuristic scheduling techniques have been merged together. Researchers working on meta-heuristic and heuristic scheduling found that by using the qualities of both the scheduling techniques, we could improve the quality solutions and the speed of convergence. It solves many issues by modifying the input population or transition operator and avoiding saturation of solution in large iteration. For workflow scheduling, hybrid approaches significantly improve the optimal solutions. Some of the hybrid scheduling algorithms are DPSO [39, 40], ACOPS [41], Max-Min Ant [42, 43], GA-ACO [44-47], GHPSO [48] etc; Detailed study of Meta-Heuristic and Hybrid Scheduling is shown in table 2.

TABLE 2: CLASSIFICATION OF META-HEURISTIC AND HYBRID SCHEDULING ALGORITHMS

Author, year	Algorithm	Scheduling	Objectives	Description	Environment
Kabir et al., 2015 [31]	ACOFS, Ant colony with feature selection	Meta-heuristic	Reduced size	In ACOFS technique merges the wrapper and filter approaches on the bases of their advantages. It provides an effective balance between exploration and exploitation of ants in the search, but also intensify the global search capability of ACO for a high-quality solution in FS.	Neural Network
Nishant et al. 2012)[30]	Ant colony optimization ACO	Meta-heuristic	Load balancing	In Nishant's algorithm, Some modification has been done on basic ACO algorithm for considering load balancing as objective function. Instead of makeing final solution directly, here each individual ant fisrt constrcut its solution then construct it into complete soltuion and the resluts improve as the iterations of the algorithm increases.	Cloud computing
Dasgupta et al., (2013)[33]	Genetic Algorithm	Meta-heuristic	Load balancing, Minimize makespan time	In this algorithm Load balancing strategies are used, which out-perfrom the FCFS algorithm. Enhance the System performance , , Improve resource utilization , Reducing job time span. It not consider job priorties, low throughout etc.	Cloud computing
Salimi et al.(2014)[34]	Fuzzy NSGA-II	Multi-objective Meta-heuristic	Minimize Makespan time , Load balancing, Cost	Fuzzy method to propose the adaptive mutation rate in order to get a better Pareto front. It use the fuzzy adaptive rate for crossover and Makespan objective optimization algorithm was compared with the NSPSO and proved that variance-based fuzzy NSGAII converges to Pareto-optimal solutions faster and with more quality.	Grid computing
Yakhchi et al. (2015)[36]	COA , Cuckoo Optimization Algorithm	Meta-heuristic	Load balancing, Reduce energy consumption	This technique applied to detect over-utilized hosts. One or more VMs are selected to transfer the heavy loaded host to less-loaded hosts. For this, they only consider less- loaded host instead of heavily- loaded and transfer all their VMs to other hosts.	Cloud Computing
Babu et al., (2013)[38]	HBB-LB ,Honeybee	Meta-heuristic	Minimize the Waiting time, Execution time	Inspired by the honeybee foraging behaviour, the removal of tasks from overloaded nodes as the honeybees do.	Cloud computing
Zhang et al, 2006[39,40]	DPSO PSO and level scehduling	Hybrid Scheduling	Makespan and Cost	Mostly PSO considered to solve the optimization problems for better optimal solutions, when problem is continuous instead of discrete. Therefore to solve the discrete problems author proposed an algorithm called DPSO for the workflow scheduling problem.	Grid computing
Keng-Mao et al. 2015[41]	ACOPS	Hybrid Scheduling	Makespan Time and Load balancing	They combines the ACO and PSO to solve the Virtual Machine scheduling problem. ACOPS ant colony optimization with particle swarm outperform other scheduling approaches.	Cloud Computing
Stutzle et al,2000[42]	MAX MIN ANT	Hybrid Scheduling	Balance the load, Makespan, cost	Main idea of hybridization is to gather the advantage of both that of max-min at initial stage and ant colony algorithm in later stage	Heterogeneous System
Yangyang Dai, 2015[46]	GA-ACO	Hybrid Scheduling	Security, Time-consuming, Reliability and Budget	integration of two meta heuristic optimization algorithm with Multi-Qos Constraints. Main fitness functions used in algorithm like Security, Time-consuming, Reliability and Budget in process of integrating ACO with GA algorithm addition to that they also take care of balancing the resources	Cloud Computing
Jun Xue ,2012[48]	GHPSO	Hybrid Scheduling	Minimize the cost and Execution time.	It is a QoS based genetic Hybridized Particle-Swarm Optimization scheduling algorithm in cloud infrastructure. In GHPSO, Operator of Genetic algorithm like Crossover , mutation is embed inside the main scheduling algorithm PSO.	Cloud Computing

3. OTHERS SCHEDULING ALGORITHMS IN HCS

In addition to the classification of scheduling algorithms into heuristic, meta-heuristic and hybrid, some are remain uncategorized, we classify them under fields such as Hadoop-based [49-52], Map-reduced [53-54], Agent based [55]etc.

3.1 Hadoop based scheduling

Hadoop framework has been widely used to process large-scale datasets on computing clusters. It is widely used for storing large datasets and processing them efficiently under distributed computing environment. Algorithms are developed as a plug-in to standard MapReduce component of Hadoop such as FIFO scheduling [49] [50], Delay Scheduler [51], LATE [52].

3.2 Map reduce Based scheduling

MapReduce is an important programming model for heterogeneous applications. Hadoop is an open-source implementation of MapReduce. Map and Reduce are two important functions, Map collects set of data file and indexed them with key-value pairs. The mapper loads this data onto machines for processing. MapReduce library combine all the value belong to same key and send these key-value pair to reduce function. Large set of data in divided into smaller data sets by using reduce function. Output file generated by reducer kept in HDFS format for performing MapReduce task. There are some issues in scheduling Map Reduce and to handle these issue some author proposed algorithm such as Cogset [53], Tarazu [54].

3.3 Agent based scheduling

In agent-based techniques, an agent is a part of software, its functionality is to continuously decides for itself and find out how to satisfy the objectives. A multi-agent system includes many agents, which work together with each other. To be successful, the agents have to co-operate and coordinate with each other. Authors [55] proposed an agent-based load balancer for multi-cloud environments. The proposed architecture consists of three agents: an executor agent, a provisioner agent, a monitor agent. The proposed algorithm overcame the provider locking challenge in the cloud and it was flexible to exploit the extreme elasticity.

TABLE 3: CLASSIFICATION OF SCHEDULING ALGORITHMS BASED ON HADOOP, MAPREDUCE AND AGENT BASED

Author, year	Algorithm	Scheduling	Objectives	Description	Environment
Shaikh et al., 2017[49]	FIFO	Hadoop Based	Execution time	It is simple and efficient Algorithm has work on First come frist serve basis , only consider one job at a times. Low data locality and No considering priority or job Size.	Hadoop Based architecture
Zaharia et al., 2010[51]	Delay Scheduler	Hadoop Based	Load balancing	The delay scheduler is an optimization of the fair scheduler, which eliminates its locality issues. State task which is in ready state, is delayed for somethime untile there is free slot in a node with needed data. To avoid starvation condithion, with long enough delayed time, the non-local task is allowed to schedule.	Heterogeneous system
Lee et al., 2011[52]	LATE	Hadoop Based	Load balancing	LATE improve the perfromance by reducing response timefor Heterogeneous environment. It use the past information. It does not ensure reliability.	Hadoop in heterogeneous environments
Valvag et al, 2013[53]	Cogset	Map reduce Based	Increasing the system efficiency, decreasing the request response time	Cogset is one mapreduce based algorithm for static load balancing. The loose coupling between HDFS and MapReduce engine is the cause of poor data locality for many applications. Basically, Cogset combines file system and map reduce engine closely.	Heterogeneous Computing system
Ahmad et al, 2012[54]	Tarazu	Map Reduce Based	Load balancing	Tarazu, is map reduced based optimizations algorithms in heterogeneous clusters to overome the low performance of mapreduce. Authors showed by simulation that using Tarazu significantly improves the performance by eliminating map phase ans bottleneck due to shuffle over a traditional MapReduce in heterogeneous clusters.	Heterogeneous Computing system
Tasquier et al, 2015[55]	Agent-load balancer	Agent Based	Load balancing	It uses a agent-based paradigm to load balance, with improve the lock-in challenge in cloud.	Cloud computing

4. ISSUES WITH SCHEDULING

In this section, we address some of the major issues that have not been stated above in scheduling algorithms. From the above literature review, many techniques exist for improving the entire scheduling algorithm but no technique is perfect. Like, a few techniques considered response time, resource utilization or migration time whereas others might not consider or ignored these metrics. However, it seems that some metrics are mutually exclusive. For example, relying on VM migration for load balancing may cause an increase in the response time. Service cost is another metric, resource utilization depends on the execution time of task and other. Therefore, there is strong need to present a comprehensive technique to many metrics improvement at a same time. Further research works, find that the energy consumption and carbon emission are two important drawbacks due to the high growth of data-centers number. Only few articles discuss these two drawbacks. Each of these issues is critically important. Therefore, providing complete scheduling strategies in a heterogeneous environment system servicing various issues like scalability, multi-objectivity, security, etc needs to be worked upon in future research work.

5. CONCLUSION

A workflow set of logical series of dependent tasks. Several studies had done in past for workflow scheduling and task scheduling in various distributed system whether it is heuristic and meta-heuristic or hybrid algorithms. Our paper provides a comprehensive analysis on workflow scheduling methods about heterogeneous computing system. Workflow structures classified into DAGs and Non-DAGs based patterns. Workflow scheduling strategies classified into three different model, Performance-based, market-based and trust-based models. These schemes are categorized objectives like makespan, load balancing, energy efficiency, security and cost into these three different categories. Literature survey shows that many author focused on reduction makespan\execution time whereas very few have covered optimization objectives as load-balancing, energy-efficiency, fault tolerant, cost, and average resource utilization. In future there is huge possibilities to perform optimization based these objectives.

The paper widely reviews the application of heuristic, meta-heuristic and hybrid techniques in the area of scheduling in heterogeneous system. Study shows that meta-heuristic technique has slower convergence rate and quality of solution, therefore to improve the problem many measure have been propose such as modification of operator in transition rules, change the pre-processing input population and hybridizing algorithms. Comparative analysis on each algorithm classifies these algorithms into heuristic, meta-heuristic and hybrid technique. Some other classifications based on Hadoop, MapReduce, Agent and others are mention. In the end, current issues regarding scheduling algorithms, objective and future possibilities in heterogeneous system are stated. Possibilities future research work is huge in these areas, the security, energy-aware concept should be integrate into cloud scheduling to resist by the attackers, un-trusted cloud users and reduce the CO^2 emissions. A variety of open issues has been also examined in our paper, which can be the focus for future research.

REFERENCE

[1] Casavant Casavant, Thomas L., and Jon G. Kuhl. "A taxonomy of scheduling in general-purpose distributed computing systems." IEEE Transactions on software engineering 14, no. 2 (1988): 141-154.

[2] van Der Aalst, Wil MP, Arthur HM Ter Hofstede, Bartek Kiepuszewski, and Alistair P. Barros. "Workflow patterns." Distributed and parallel databases 14, no. 1 (2003): 5-51.

[3] van der Aalst, Wil MP, Alistair P. Barros, Arthur HM ter Hofstede, and Bartek Kiepuszewski. "Advanced workflow patterns." In CoopIS, vol. 1901, pp. 18-29. 2000.

[4] Sakellariou, Rizos, and Henan Zhao. "A low-cost rescheduling policy for efficient mapping of workflows on grid systems." Scientific Programming 12, no. 4 (2004): 253-262.

[5] M. Wu, D. Gajski, "Hypertool: A Programming Aid for Message Passing Systems", IEEE Trans. Parallel and Distributed Systems, vol. 1, pp. 330-343, July 1990.

[6] T. Yang, A. Gerasoulis, "DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors", IEEE Trans. Parallel and Distributed Systems, vol. 5, no. 9, pp. 951-967, Sept. 1994.

[7] Kim, S. J., and J. C. Browne. "A general approach to mapping of parallel computation upon multiprocessor architectures." In International conference on parallel processing, vol. 3, no. 1, p. 8. 1988.

[8] Kwok, Yu-Kwong, and Ishfaq Ahmad. "Dynamic critical-path scheduling: An effective technique for allocating task graphs to multiprocessors." IEEE transactions on parallel and distributed systems 7, no. 5 (1996): 506-521.

[9] Graham, Ronald L. "Bounds for certain multiprocessing anomalies." Bell Labs Technical Journal 45, no. 9 (1966): 1563-1581.

[10] Topcuoglu, Haluk, Salim Hariri, and Min-You Wu. "Task scheduling algorithms for heterogeneous processors." In Heterogeneous Computing Workshop, 1999.(HCW'99) Proceedings. Eighth, pp. 3-14. IEEE, 1999.

[11] Braun, Tracy D., Howard Jay Siegel, Noah Beck, Ladislau Boloni, Muthucumaru Maheswaran, Albert I. Reuther, James P. Robertson, Mitchell D. Theys, and Bin Yao. "A taxonomy for describing matching and scheduling heuristics for mixed-machine heterogeneous computing systems." In Reliable Distributed Systems, 1998. Proceedings. Seventeenth IEEE Symposium on, pp. 330-335. IEEE, 1998.

[12] Cirou, Bertrand, and Emmanuel Jeannot. "Triplet: a clustering scheduling algorithm for heterogeneous systems." In Parallel Processing Workshops, 2001. International Conference on, pp. 231-236. IEEE, 2001.

[13] Bozdag, Doruk, Umit Catalyurek, and Fuisun Ozguner. "A task duplication based bottom-up scheduling algorithm for heterogeneous environments." In Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International, pp. 12-pp. IEEE, 2006.

[14] Ilavarasan, E., P. Thambidurai, and R. Mahilmannan. "Performance effective task scheduling algorithm for heterogeneous computing system." In Parallel and Distributed Computing, 2005. ISPDC 2005. The 4th International Symposium on, pp. 28-38. IEEE, 2005.

[15] Andreev, Radoslav, Patrick Healy, and Nikola S. Nikolov. "Applying ant colony optimization meta-heuristic to the DAG layering problem." In Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International, pp. 1-9. IEEE, 2007

[16] Healy, Patrick, and Nikola S. Nikolov. "How to layer a directed acyclic graph." In International Symposium on Graph Drawing, pp. 16-30. Springer Berlin Heidelberg, 2001.

[17] Coffman, Edward G., and Ronald L. Graham. "Optimal scheduling for two-processor systems." Acta informatica 1, no. 3 (1972): 200-213.

[18] Nikolov, Nikola S., and Alexandre Tarassov. " Graph layering by promotion of nodes." Discrete Applied Mathematics 154, no. 5 (2006): 848-860.

[19] Ma, Yan, and Bin Gong. "Double-layer scheduling strategy of load balancing in scientific workflow." In Parallel and Distributed Systems (ICPADS), 2009 15th International Conference on, pp. 671-678. IEEE, 2009.

[20] G. C. Sih and E. A. Lee, A Compile Time Scheduling Heuristic for Interconnection Constrained Heterogeneous Processor Architectures," IEEE Transactions on Parallel and Distributed Systems, vol. 4, pp. 175{186, Feb. 1993

[21] Lee, Young-Choon, and Albert Zomaya. "A novel state transition method for metaheuristic-based scheduling in heterogeneous computing systems." IEEE Transactions on Parallel and Distributed Systems 19, no. 9 (2008): 1215-1223.

[22] S. Darbha and D.P. Agrawal, "Optimal Scheduling Algorithm for Distributed-Memory Machines," IEEE Trans. Parallel and Distributed Systems, vol. 9, no. 1, pp. 87-95, Jan. 1998.

[23] Mingyu, Kun and Chang-kwanchen, 2008. An Adaptive Scheduling Algorithm for Scheduling Tasks in Computational Grid "Grid Publication Date: pp: 24-26.

[24] I. Ahmad, Y. Kwok, "A New Approach to Scheduling Parallel Programs Using Task Duplication", Proc. Int'l Conf. Parallel Processing, vol. 2, pp. 47-51, 1994.

[25] B. Kruatrachue, T.G. Lewis, "Grain Size Determination for Parallel Processing", IEEE Software, pp. 23-32, Jan. 1988.

[26] Y. Chung, S. Ranka, "Applications and Performance Analysis of a Compile-Time Optimization Approach for List Scheduling Algorithms on Distributed Memory Multiprocessors", Proc. Supercomputing, pp. 512-521, Nov. 1992.

[27] Awad, A. I., N. A. El-Hefnawy, and H. M. Abdel_kader. "Enhanced particle swarm optimization for task scheduling in cloud computing environments." Procedia Computer Science 65 (2015): 920-929.

[28] Guo, Lizheng, Shuguang Zhao, Shigen Shen, and Changyuan Jiang. "Task scheduling optimization in cloud computing based on heuristic algorithm." JNW 7, no. 3 (2012): 547-553.

[29] Pandey, Suraj, Linlin Wu, Siddeswara Mayura Guru, and Rajkumar Buyya. "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments." In Advanced information networking and applications (AINA), 2010 24th IEEE international conference on, pp. 400-407. IEEE, 2010

[30] Nishant, Kumar, Pratik Sharma, Vishal Krishna, Chhavi Gupta, Kuwar Pratap Singh, and Ravi Rastogi. "Load balancing of nodes in cloud using ant colony optimization." In Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on, pp. 3-8. IEEE, 2012.

[31] Kabir, Md Monirul, Md Shahjahan, and Kazuyuki Murase. "A new hybrid ant colony optimization algorithm for feature selection." Expert Systems with Applications 39, no. 3 (2012): 3747-3763.

[32] Singh, Harmel Karam, and Abdou Youssef. "Mapping and scheduling heterogeneous task graphs using genetic algorithms." Master's thesis, George Washington University, 1995.

[33] Dasgupta, Kousik, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mandal, and Santanu Dam. "A genetic algorithm (ga) based load balancing strategy for cloud computing." Procedia Technology 10 (2013): 340-347.

[34] Salimi, Reza, Homayun Motameni, and Hesam Omranpour. "Task scheduling using NSGA II with fuzzy adaptive operators for computational grids." Journal of Parallel and Distributed Computing 74, no. 5 (2014): 2333-2350.

[35] Afzalirad, Mojtaba, and Javad Rezaeian. "A realistic variant of bi-objective unrelated parallel machine scheduling problem: NSGA-II and MOACO approaches." Applied Soft Computing 50 (2017): 109-123.

[36] Yakhchi, Moona, Seyed Mohssen Ghafari, Shahpar Yakhchi, Mahdi Fazeli, and Ahmad Patooghi. "Proposing a load balancing method based on Cuckoo Optimization Algorithm for energy management in cloud computing infrastructures." In Modeling, Simulation, and Applied Optimization (ICMSAO), 2015 6th International Conference on, pp. 1-5. IEEE, 2015.

[37] Rajabioun, Ramin. "Cuckoo optimization algorithm." Applied soft computing11, no. 8 (2011): 5508-5518.

[38] LD, Dhinesh Babu, and P. Venkata Krishna. "Honey bee behavior inspired load balancing of tasks in cloud computing environments." Applied Soft Computing 13, no. 5 (2013): 2292-2303

[39] Zhang, X. D., X. P. Li, Qian Wang, and Y. C. Yuan. "Hybrid particle swarm optimization algorithm for cost minimization in service-workflows with due dates." Journal on Communications 29, no. 8 (2008): 87-93

[40] Zhang, Lei, Yuehui Chen, and Bo Yang. "Task scheduling based on PSO algorithm in computational grid." In Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on, vol. 2, pp. 696-704. IEEE, 2006.

[41] Cho, Keng-Mao, Pang-Wei Tsai, Chun-Wei Tsai, and Chu-Sing Yang. "A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing." Neural Computing and Applications 26, no. 6 (2015): 1297-1309

[42] Stützle, Thomas, and Holger H. Hoos. "MAX–MIN ant system." Future generation computer systems16, no. 8 (2000): 889-914.

[43] Yassa, Sonia, Rachid Chelouah, Hubert Kadima, and Bertrand Granado. "Multi-objective approach for energy-aware workflow scheduling in cloud computing environments." The Scientific World Journal 2013 (2013).

[44] Lee, Zne-Jung, Shun-Feng Su, Chen-Chia Chuang, and Kuan-Hung Liu. "Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment." Applied Soft Computing 8, no. 1 (2008): 55-78.

[45] Rashidi, Shima, and Saeed Sharifian. "A hybrid heuristic queue based algorithm for task assignment in mobile cloud." Future Generation Computer Systems 68 (2017): 331-345.

[46] Dai, Yangyang, Yuansheng Lou, and Xin Lu. "A task scheduling algorithm based on genetic algorithm and ant colony optimization algorithm with multi-QoS constraints in cloud computing." In Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2015 7th International Conference on, vol. 2, pp. 428-431. IEEE, 2015.

[47] Ragmani, Awatif, Amina El Omri, Noreddine Abghour, Khalid Moussaid, and Mohammed Rida. "A performed load balancing algorithm for public Cloud computing using ant colony optimization." In Cloud Computing Technologies and Applications (CloudTech), 2016 2nd International Conference on, pp. 221-228. IEEE, 2016.

[48] Xue, Sheng-Jun, and Wu Wu. "Scheduling workflow in cloud computing based on hybrid particle swarm algorithm." Indonesian Journal of Electrical Engineering and Computer Science 10, no. 7 (2012): 1560-1566.

[49] Shaikh, Mrs Bareen, Mrs Kavita Shinde, and Mrs Sangeeta Borde. "Challenges of Big Data Processing and Scheduling of Processes Using Various Hadoop Schedulers: A Survey." International Journal of Multifaceted and Multilingual Studies 3, no. 12 (2017).

[50] Li, Zhuozhao, and Haiying Shen. "Designing a hybrid scale-up/out hadoop architecture based on performance measurements for high application performance." In Parallel Processing (ICPP), 2015 44th International Conference on, pp. 21-30. IEEE, 2015.

[51] Zaharia, Matei, Dhruba Borthakur, Joydeep Sen Sarma, Khaled Elmeleegy, Scott Shenker, and Ion Stoica. "Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling." In Proceedings of the 5th European conference on Computer systems, pp. 265-278. ACM, 2010

[52] Lee, Kyong-Ha, Yoon-Joon Lee, Hyunsik Choi, Yon Dohn Chung, and Bongki Moon. "Parallel data processing with MapReduce: a survey." AcM sIGMoD Record 40, no. 4 (2012): 11-20.

[53] Valvåg, Steffen Viken, Dag Johansen, and Åge Kvalnes. "Cogset: a high performance MapReduce engine." Concurrency and Computation: Practice and Experience 25, no. 1 (2013): 2-23.

[54] Ahmad, Faraz, Srimat T. Chakradhar, Anand Raghunathan, and T. N. Vijaykumar. "Tarazu: optimizing mapreduce on heterogeneous clusters." In ACM SIGARCH Computer Architecture News, vol. 40, no. 1, pp. 61-74. ACM, 2012.

[55] Tasquier, L. "Agent based load-balancer for multi-cloud environments, Columbia Int." Publ. J. Cloud Comput. Res 1 (2015): 35-49.