Scheduling Independent Parallel Jobs in Cloud Computing: A Survey

Samaneh Abdolhosseini, Mohammad Taghi Kheirabadi
Department of Computer, Gorgan Branch, Islamic Azad University, Gorgan, Iran
m.kheirabadi@gorganiau.ac.ir

***Abstract*— The impressive and rapid development of the internet and wireless networks leads to growing of users in the last decade. Therefore, the limited resources of these systems are now more evident than in the past. Cloud computing is the latest technology to handle the limitation of resources for users. Type of jobs play the main role in the design of scheduling algorithms. A job can be run simultaneously by multi-processor called parallel job, while the job can run by a single processor called serial job. In addition, based on dependency of jobs to each other, the jobs can be divided into dependent and independent jobs. Scheduling the independent parallel jobs is one of important challenges in cloud computing. Hence, in this paper, we classified the existing algorithms of scheduling independent parallel jobs into two main categories including Non-Layer and Two-Layer. This division is performed based on the number of jobs running on a processor simultaneously. Furthermore, the existing scheduling algorithms belong to each categories are divided into two subcategories based on their solving techniques including heuristic and meta-heuristic. Then, the algorithms belong to each category are described in detail. After that, these algorithms are compared to each other based on their different attributes. Our analysis show that the existing Two-Layer scheduling algorithms focus on cost parameter to increase the performance of scheduling algorithms by reducing the waste time of CPU through simultaneous assigning more than one job to each physical machine, while Non-Layer scheduling algorithms didn't pay attention to this issue and only employ techniques to manage the scheduling queue in order to improve the different parameters such as cost, energy, load balancing and deadline.**

***Index Terms*— Cloud Computing; Independent Jobs; Resource Allocation; Scheduling Parallel Job.**

I. INTRODUCTION

Cloud computing is the latest technology employed by end users via Internet to confront with limited resources[1, 2]. The services in cloud computing are classified into Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [3, 4]. Since many jobs need to use limited available resources, scheduling algorithms plays main role in cloud computing [5-7].

In general, the scheduling job algorithms can be divided into two classes including independent and dependent. In addition, based on the number of CPU that is simultaneously used by the job, the scheduling algorithms can be divided into two categories, including serial and parallel. Scheduling parallel independent jobs is one of fundamental issues in cloud computing.

Different studies have been performed on the job scheduling in cloud computing. Authors in [8] examined the scheduling algorithms for distributed systems, such as cloud computing. In this study, scheduling algorithms were categorized based on system scheduling, input and output data, running frequency, program model, system purpose, and scheduling objectives. In [9], authors proposed a taxonomy for different scheduling techniques based on their features, such as static or dynamic, dependency of jobs, and etc. In [10] , a review on the scheduling by work stealing is provided from the perspective of scheduling algorithms, optimization of algorithm implementation and processor architecture oriented optimization. In [11], authors prepared a comparative study of static and dynamic task scheduling schemes. In addition, they compared different scheduling methods based on their advantages and limitations. In [12], authors carried out a comprehensive review of job scheduling algorithms in the cloud computing environment. They investigated the disadvantages and advantages of each of the conventional algorithms, including RR, SJF and min-min, and compared these in terms of their key functions, such as their application in different loading modes in the cloud computing system. In prior studies, little attention has been paid to a discussion of the independence or dependence of the duties and the types of relationship between jobs.

In this paper, we classified the existing parallel independent jobs scheduling algorithms into two categories including non-layer and two-layer algorithms. Furthermore, these categories are divided into two subcategories based on their solution method including heuristic and meta-heuristic. After that, the existing algorithms belong to each category is described in detail and finally compared with each other based on different attributes such as purpose, method, evaluation parameters, and results.

The rest of this paper is organized as follows. Section II describes the resources allocation in cloud computing. In Section III, the existing scheduling schemes for independent parallel jobs are classified and explained. Section IV explores the open issues and discussion related to scheduling parallel jobs, and finally, in Section V, conclusions and future works are presented.

II. RESOURCES ALLOCATION IN CLOUD COMPUTING

In this section, we first define cloud computing. Then, we discuss the resource allocation techniques in cloud computing.

A. Cloud Computing

Cloud computing is the latest technology for providing different remote services for low resource systems in order to cope with the limitation of resources, such as CPU, memory, and applications [13, 14]. In fact, cloud is the new model of services provided in IT-based systems. In this regard, the user employs the service only and there is no need to know anything about the technology or implementation of it[3]. The use of cloud computing services can lead to the access to many efficient and economical infrastructures such as servers, networks, storages, and platforms, [9].

Figure 1 shows a general overview of cloud computing. As shown in this Figure, the services in cloud computing can be divided into three groups, including the infrastructure as a service (IaaS), the platforms as a service (PaaS), and the software as a service (SaaS) [4, 15].

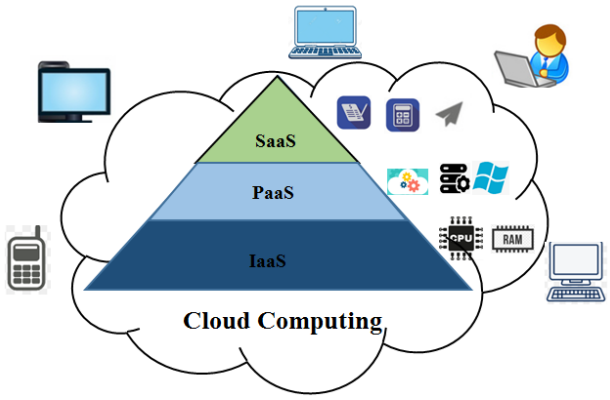

Figure 1: Overview of cloud computing

B. Resource Allocation

Cloud computing are faced with different challenges including resource allocation, scheduling, pricing, and off-loading[16]. The resources allocation is one of the most important challenges in computational systems, including cloud computing because end users may access the resource at any time[17]. For example, the cloud data centers receive frequent offers for resources from cloud computing models. However, the abundance of resources may lead to the allocation of undesirable resources [18, 19].

The resources in the cloud environments can be divided into two groups including physical and logical resources.

- Physical resources: These include the set of memory, network, processor and I/O devices in the cloud environment [20].
- Logical resources: These are the same as virtual resources, which include the abstract mode of the system which can temporarily control physical resources such as bandwidth, storage and energy [20, 21].

Virtualization is one of ways to increase the efficiency of physical machines. Cloud computing is based on virtualization technology; this allows users to send their requests to the cloud in the form of a virtual machine [22]. The ultimate goal is to allocate resources in the cloud environment, which involves finding the right hosts for virtual machines containing user requests. Figure 2 shows an overview of a resource allocation system in which the implementation of several different operating systems is carried out in a physical machine using the concept of virtualization.

The resource allocation schemes should be taken into account the following situations and avoid them as far as possible:

- Competition over resources: This situation occurs when multiple users and applications are simultaneously trying to allocate resources.
- Shredded resources: This occurs when there are sufficient resources. However, they cannot be assigned to the requested application because they are split and separated,

Lack of resources: This situation occurs when there are multiple jobs waiting for resources and limited resources are available, for example the requests for memory, I/O devices and processors.

Figure 2: An overview of the resource allocation system [23, 24]

- Assigning too many resources: This occurs when the program is allocated with resources that are greater than necessary.
- Assigning too few resources: This occurs when the program is allocated with insufficient resources.

From the perspectives of cloud users, resources allocation should be carried out at a low cost and within a short period of time[25]. From the cloud provider's viewpoint, the dynamic prediction of user demand, the nature of users, and the demands of the software is a difficult job. Due to the diversity of resource constraints and site constraints, the dynamic nature of resource demands, and the environmental requirements, a strategy is required for allocating resources efficiently and dynamically in the cloud environment [26].

In this case, research into resource allocation is important. Some of the challenges involved in allocating resources are resource pricing, over-reservation of resources, flexibility and scheduling parallel jobs. Among the challenges posed by the resources allocation in cloud computing, the scheduling parallel of jobs is a particularly important issue that seeks to identify an optimal scheduler to perform jobs and allocate an optimal quantity of resources. In the following section, the existing scheduling parallel jobs are classified and explained in detail.

C. Scheduling Parallel Job

The concept of scheduling parallel jobs refers to multiple jobs that can be run simultaneously on multiple processors. When jobs are complex, involve numerous

calculations and have time limits, scheduling parallel can divide the jobs into smaller jobs to give fast results. Scheduling in multiprocessor systems deals with the optimal assignment of a set of jobs and their execution order to minimize total runtime in a parallel multiprocessor system [23]. The scheduling parallel jobs are divided into two categories based on the types of job, i.e. independent and dependent jobs, and these will be described below.

D. Scheduling Dependent Parallel Jobs

In this category, the jobs are dependent to each other. In other words, scheduling scheme should take into account the dependency of jobs [27]. For example, if job *A* is dependent to job *B*, then the priority of job B is more than job *A* and job *B* must run sooner than job *A*.

E. Scheduling Independent Parallel Jobs

In contrast to the last category, in this category the jobs are independent from each other and there is no priority between jobs based on their dependency. Hence, the scheduling scheme can freely schedule the jobs [28].

III. SCHEDULING INDEPENDENT PARALLEL JOBS

The scheduling independent parallel jobs is one of the main challenges in cloud computing. Many studies have been carried out on the scheduling parallel jobs. In this paper, as shown in Figure 3, the existing independent scheduling parallel jobs are classified into two main categories including non-layer and two-layer. In addition, existing schemes that belong to each category is divided into two sub-categories including heuristic and meta-heuristic. These categories and existing works belong to each category are explained in detail in the following.

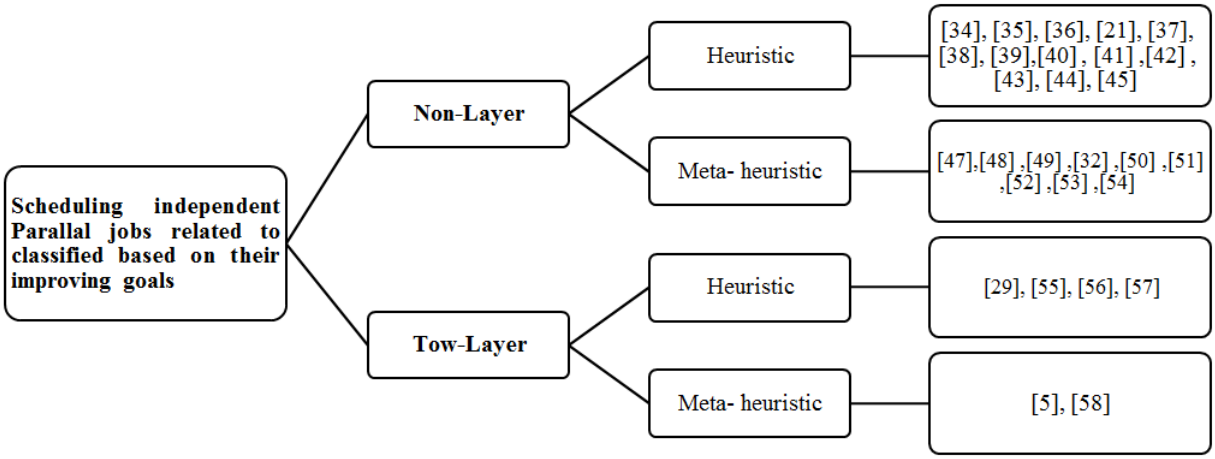

Figure 3: Classification of scheduling parallel algorithms for independent jobs in cloud computing

A. Non-Layered Independent Scheduling Parallel Jobs

Cloud service providers are looking for appropriate scheduling algorithms to run jobs in the cloud so that they can significantly reduce CPU's idle cycles[29]. For this purpose, non-layered scheduling algorithms such as FCFS were proposed for scheduling parallel jobs. In these algorithms, Jobs enter the waiting queue for resource allocation. Then, they are assigned to physical processors for execution. For example, as shown in Figure 4, there are 5 physical processors in the system and two independent jobs J1 and J2, which are run by these processors so that J1 and J2 need 2 and 4 processors to run respectively. Since the FCFS algorithm is employed for scheduling and J1 arrives earlier than j2, the job J1 will be run first by getting the two processors. Considering J2 needs 4 processors to run, J2 has to wait because there are only 3 available free processors. After running J1, it releases its processors and then the job J2 can be run by getting 4 processors.

The existing non-layer independent parallel jobs algorithms is divided into two sub-categories based on their solution techniques including: heuristic and meta-heuristic. Each of these subcategories is discussed as follows.

1. Heuristic Non-Layer Scheduling Algorithms

Heuristic algorithms are problem-oriented algorithms designed for a particular problem[30]. Heuristic search methods can provide an optimal solution for a problem within a limited time. Their solutions are based on learning and exploration and seek to find the desired answer and speed up the response process [31-33]. In the following, some of the non-layer heuristic algorithms for scheduling independent parallel jobs are explained.

- Durga et al. [34] proposed a context-aware job scheduling algorithm. The proposed algorithm is called A Two-Stage Queue Model (ATSQM). The proposed method employs a multi-level queue scheduling algorithm based on the heuristic resource allocation approach. This algorithm contains two queues: request handler queue (RH) and media processing servers queue (MS). Request handler queue is used for the scheduling of the requests to the corresponding serves, while MS queue is employed for optimal resource allocation. The proposed method examined the relationship between response time and cost. The simulation results showed that the proposed algorithm reduces the deadline compared to the existing methods by 30% -40%.
- Li [35] investigated the scheduling parallel jobs. In the proposed method, two main issues were discussed including: energy constraint and time constraint. Two algorithms including pre-power and post-power measurement were proposed on multi-core processors. The main strategy of this paper was the inclusion of an equal-speed method in the proposed algorithm, which leads to an increased performance for the proposed algorithm. The results showed an

improvement in the energy variable compared to the previous methods.

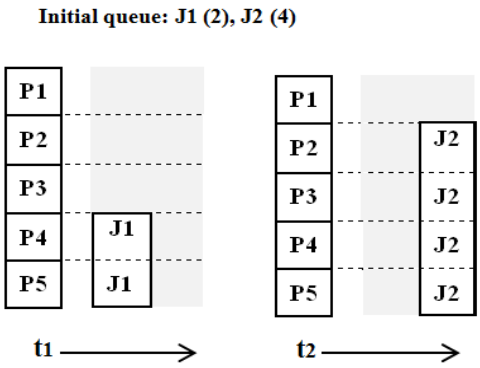

Figure 4: Examples of No.n-Layered scheduling parallel Jobs

- Chen et al.[36] proposed an efficient algorithm to minimize the schedule length using the budget level ,called MSLBL, for selecting a processor. This algorithm satisfied budget constraints and minimised the length of parallel applications. Cloud-based services were used in this paper for the analysis of the proposed method and for the directed acyclic graph (DAG) scheduling model. The result of this simulation showed a decrease in the duration of the programs runtime while meeting resource constraints.
- Shi et al. [21] presented a way to improve resource allocation based on the scheduling parallel jobs in the cloud environment. The purpose of this research is to schedule resource allocation for the parallel execution of jobs in a way that minimises job completion time (JCT). In this paper, the TaPRA algorithm and its simpler version were presented to solve the problem of scheduling parallel jobs. A comparison of this algorithm with previous ones (min-min, GA, MM-GA) showed that the TaPRA algorithm can reduce JCT by 40–45% compared to other algorithms, and by 28–60% percent on average. The TaPRA-fast algorithm was 10 times faster than the TaPRA algorithm, and this led to better and more suitable functioning of this algorithm.
- Cao et al. [37] proposed the Energy-aware resource efficient scheduling (EARES) scheduling algorithm. They explored the scheduling parallel jobs to reduce energy consumption. Dynamic voltage and frequency scaling (DVFS) was used to reduce the energy consumption to an acceptable level. The proposed algorithm minimized the resource allocation and execution time for jobs. It also reduced the load on VMs, which in turn reduced energy consumption and uses more resources. The main disadvantage of the proposed algorithm was the lack of attention to VM allocation strategies for fixing jobs and improving migration, without affecting response time for large clusters.
- Zhang et al. [38] proposed a strategy for optimising the scheduling of jobs and combined resources based on fuzzy clustering. The proposed algorithm is called Algorithm Based on Fuzzy Clustering (ABFC) .This process composed of two parts including; fuzzy clustering with dynamic combination of node information and Bayesian classification strategies. To prioritise each job, the relationship between jobs and real-time node load was utilised. The experiment results indicated that the proposed scheduling algorithm gives improved parallel execution of jobs and resource allocation compared to the classic scheduling algorithms.
- Domanal et al. [39] proposed a load balancing approach with the goal of local optimisation in terms of distributing incoming jobs uniformly across servers or VMs. The proposed algorithm was a modified throttled algorithm in which there is a table of virtual machines. In the proposed method, efforts were made to improve response time and access to efficient VMs. VMs that are in a better position, based on customer demand for processing, were selected from the table and were used to process the customer's request. The simulation results showed that the proposed algorithm distributes the load more equally among virtual machines than the throttled algorithm and the Round Robin (RR). The disadvantages of the proposed method include a lack of attention to the distance between the sources and the data centres.
- Raghu et al. [40] proposed an energy-conscious algorithm for the scheduling parallel jobs in high performance computing. Energy consumption is one of the most important concerns in high performance systems, and the frequency or voltage used to perform a job has a key effect on overall energy consumption. They proposed a Power Aware Algorithm for Scheduling **(**PAAS) algorithm for scheduling jobs that focused on energy reduction in High Performance Computing (HPC) systems. This algorithm directs job scheduling for intelligent decision making, based on information in the database. The knowledge base provides a suitable frequency and voltage that minimises energy consumption for a particular job. Dynamic voltage frequency scaling (DVFS) is optimized on all nodes, based on the frequency and voltage. The results of the implementation of this method indicate that on average it can saves energy by 12.64%. Lack of attention to the heterogeneity of the computing environment and to the implementation in large HPC clusters are the two main disadvantages of the proposed method.
- Bhoi et al. [41] addressed the concept of scheduling parallel jobs. There are many jobs to be performed by existing resources in the cloud computing environment to achieve the minimum completion and response times. With this in mind, they designed a scheduling algorithm that can provide an effective mapping for assigning jobs to resources. The proposed method is a unique modification of the max-min algorithm, and is based on an extensive review of the max-min scheduling algorithm in the cloud computing environment. The proposed max-min algorithm used expected execution time rather than completion time, reducing the overall burden of resources. The implementation of the proposed algorithm showed that it has a better performance than the existing max-min algorithms and other common scheduling algorithms.
- Zhang et al. [42] suggested policies for the scheduling parallel jobs based on collaboration in mobile cloud computing called Lagrangian Relaxation Based Aggregated Cost (LARAC). Mobile applications comprise a sequence of small jobs that form a linear topology. The aim of the proposed algorithm is to optimise energy usage. The shortest path routing algorithm is used for routing on a linear graph. Experiment results showed that implementing collaborative requests between mobile devices and job migrating to the cloud with collaborative approach significantly improves the energy consumption of mobile devices.
- Chen et al. [43] examined the scheduling parallel jobs in the cloud computing environment and showed that to fully utilise cloud-computing power, there is a need for an efficient

job-scheduling algorithm. The min-min algorithm is a simple and efficient algorithm that gives better results than other algorithms in terms of the completion time of jobs. However, the biggest disadvantage of this algorithm is the lack of a load balancing mechanism. To achieve this objective, an improved load-balancing algorithm called Load balancing in min-min (LBIMM), based on the min-min algorithm, was introduced to increase the utilisation of resources. The simulation results of the proposed algorithm showed that it gives significant increases in the performance, access to resources, and user satisfaction.

- Koneru et al. [44] believed that the processing of specific jobs can be allocated to various types of virtual machines that are automatically introduced and interrupted during execution. The method proposed in this paper focuses on the effectiveness of scheduling algorithms for real-time cloud computing services. In the algorithm, jobs with an earlier end time, which are less likely to be aborted, have a higher priority in the resources allocation. The proposed algorithm is based on the RR algorithm. The disadvantage of the proposed algorithm is that it is only discussed theoretically.
- Kumar et al. [45] proposed an algorithm to handle the issue of scheduling parallel jobs called Algorithm utilizing the turnaround time utility (ATU). The proposed method in this paper utilizes the Turnaround time Utility which employs a gain function and a loss function for each single job. The proposed algorithm also performs high-priority jobs faster than those with low priority, to give early completion of each job. The proposed algorithm is implemented in both preventive and non-preventive ways. The experimental results showed that the output performance of the proposed scheduling algorithm is better than both preventive and non-preventive previous scheduling methods.

2. Meta-Heuristic Non-Layer Scheduling Algorithms

In the case of complex NP-hard problems, the designs of heuristic algorithms are more sophisticated. Hence, the meta-heuristic algorithms are proposed to handle this wide range of problems[46]. In fact, the meta-heuristic algorithms are employed to find, generate, and select the best solution for a problem to cope with the complexity of the NP-hard problems. The meta-heuristic algorithms such as PSO, ant colony, and bee-colony take into account a population and by employing a fitness function try to find the better population. By this method, these algorithms can solve very complex problems. Since the scheduling independent parallel job is a complex NP-hard problem, some of researchers apply this approach to handle this problem. In the following, some of the studies carried out with the meta-heuristic algorithms are investigated.

- Jana et al. [47] claimed that given the inherent nature of cloud computing, the allocation of dynamic resources and the appropriate distribution of load on a cloud server is a challenging job. Therefore, job scheduling is a necessary step to increase the performance of cloud computing. Hence, they proposed a method to handle this problem. They proposed a new method by employing a modified PSO algorithm called MPSO. This algorithm focuses on two basic parameters in cloud scheduling: the average time for scheduling and the successful execution rate. The simulation results showed that the MPSO algorithm performs better than the MAX-min and min-min algorithms.
- Kumar et al. [48] proposed an efficient algorithm to improve scheduling parallel jobs using meta-heuristic techniques. In the proposed method, jobs are stored in the management queue. Relevant resources are assigned to jobs according to their priority. In addition, the new jobs are analyzed and stored in the demand queue. In proposed method, combining of two meta-heuristic algorithms including genetic algorithm (GA) and Particle Swarm Optimization (PSO) are employed. So, the proposed algorithm is called Hybrid Genetic-Particle Swarm Optimization (HGPSO). The proposed algorithm are evaluated by simulation. The experiment results showed that the proposed HGPSO reduces runtime compared to the GA and PSO algorithms. It also increases the access and scalability metrics compared to existing algorithms.
- Laha et al. [49] proposed the ICSA algorithm to minimize Makespan in the scheduling parallel jobs using Cuckoo algorithm. The proposed method, begin with the production of the initial population for scheduling, the law of the longest processing time (LPT) and the job exchange mechanism was used. Then, based on the ICSA algorithm, the best population was selected and implemented. In the proposed approach, an exploratory approach was first proposed using a modular operator to turn the CSA's continuous position into scheduling job to produce a Cuckoo. Then a location-based exploratory exchange approach was proposed to generate smart invoices in ICSA algorithm. The simulation results showed that the proposed algorithm has a better performance compared to existing algorithms.
- Abdullahi et al. [32] presented the DSOS algorithm, which is based on a discrete search mechanism for optimal scheduling of jobs and resources. The DSOS algorithm can be faster than the PSO algorithm, and is therefore more suitable for a large-scale scheduling problem. In a larger search space, the DSOS algorithm performs better than the PSO algorithm, and requires fewer parameters for implementation, making it the most suitable choice for searching in large spaces.
- Wen et al. [50] suggested the artificial bee colony (ABC) algorithm for the multi-purpose optimization of scheduling parallel jobs. The ABC algorithm is a solving algorithm used to find, generate, and select an exploratory solution, which gives better solution overall [31-33]. The main purpose of the paper is dealing with how to reduce energy consumption by providing a timetable for reasonable resources. The two main variables discussed in this article are energy consumption and the implementation time for parallel jobs. The proposed method is based on the Bee colony algorithm and a mathematical model for optimising resource scheduling within a cloud-based environment. To illustrate the improvement in the variables examined, the results of the proposed algorithm are compared with the GAP and NSGAII resource allocation algorithms, showing that the proposed algorithm gives better results. One of the advantages of this algorithm is that it can find all Pareto optimal solutions for small problems. The main disadvantage of this method is that for complex problems of increasing complexity, the algorithm requires more time to reach a solution; this is not cost-effective, and the overall cost of implementing the proposed method increases.
- Hung et al. [51] investigated the scheduling parallel jobs. They believed that despite the increasing use of mobile computing, it is difficult to exploit its full potential due to its inherent problems. In this paper, an innovative genealogy-

based approach is proposed for scheduling parallel jobs in which the processing time of each job is minimised by considering the competition between the network and the operating cost. The results of the implementation of the proposed method showed that there is a trade-off between cost and runtime when the numbers of generations increase. In general, the results of this simulation indicated an improvement in the results and the parameters studied compared to previous methods, which is the main advantage of this approach. Among its disadvantages, it should be noted that the steps involved in the proposed genetic algorithm are not very clear.

- Lin et al. [52] focused on the scheduling parallel jobs in cloud computing and proposed a prediction-based task scheduling (PTS) strategy. The PTS algorithm takes into account the cost of network communication and the scheduling of predefined jobs, and can be combined with complementary algorithms such as ACO to generate the Pre-allocation ant colony optimization algorithm (PACO) algorithm. The experiment results of the PACO algorithm indicated that the proposed algorithm has a better performance in terms of job scheduling than algorithms such as min-min and GA. Failure to implement the algorithm in real-time cloud computing systems is one of the main disadvantages of the proposed algorithm.
- Zhan et al. [53] used meta-heuristic algorithms to schedule independent jobs. The effective use of computing resources to gain maximum profit through the job scheduling system is one of the ultimate goals of computing service providers. One of these algorithms is the Particle swarm optimization (PSO) algorithm for large-scale optimisation. Using the PSO algorithm to optimise the scheduling parallel jobs reduces the response time and increases the amount of access to resources. The PSO algorithm works well on cluster formation and general optimisation issues. This algorithm can rapidly converge with a dynamic change in weight and produce a desirable solution. Fast synchronisation, reliability and high accuracy are the main advantages of the PSO algorithm.
- Li et al. [54] proposed a job scheduling algorithm for load balancing based on an optimised ACO algorithm called Load-balancing ant colony optimization (LBACO). The main purpose of the proposed method is scheduling based on the load balancing of the system and to minimising the set of jobs. The combination of the proposed method with the optimised ACO algorithm results in better performance than the FCFS and ACO algorithms. However, too much attention to the independence of resources results in a lack of dependency of resources. Lack of attention to the heterogeneity of resources and the mobility of the environment are other disadvantages of the proposed method.

B. Two–Layer independent Scheduling Parallel Jobs

Non-layered scheduling algorithms have a lot of waste of CPU resources. For example, the FCFS algorithm suffers from the fragmentation problem of the CPU. As seen in Figure 4, jobs J1 and J2 cannot run simultaneously, because there is not enough CPU to assign them separately. Hence, they run one by one based on their priority. As a result, a number of processors cannot take part in processing that lead to waste of resources. Given that, multi-processors jobs cannot use all of the CPU's space, so a percentage of CPU space is left unused. Therefore, the two-layered scheduling algorithm was proposed to efficiently use the CPU and increase the parallel execution of jobs and reduce the loss of CPU's resources. These algorithms work based on the structure of multi virtual machines on each physical machine in order to assign more than one job to each physical machine. As seen in Figure 5, jobs J1 and J2 need 2 and 4 processor respectively. In the two-layer architecture, both of these jobs can run simultaneously by employing several virtual machines in each physical machine.

The existing two-layer scheduling algorithms based on their approach of solving the problem are divided into two categories including heuristic and meta-heuristic. The existing algorithms belong to each category is described in the following.

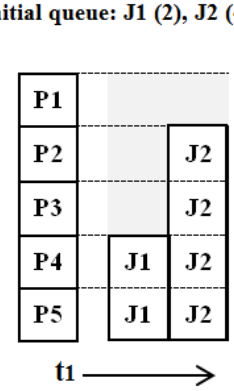

Figure 5: Examples of Tow-Layered scheduling parallel Jobs

1. Heuristic Two -Layer Scheduling Algorithms

In the following, the heuristic two-layer algorithms for scheduling independent parallel jobs are explained.

- Liu et al. [29] employed the architecture of the virtual machine has two prioritised layers that organise the VMs for parallel execution of jobs. Each virtual machine is partitioned into two background (Bg) and foreground (Fg) layers. The CPU in the Fg layer has a higher priority than that in the Bg layer. The performance of the single-processor jobs is roughly related to the total space allocated for the processor. The purpose of the proposed method is to reduce the idle cycle and make optimal use of it. The algorithm is called ACFCFS, and it has all the advantages of scheduling parallel algorithms in the cloud (e.g. the FCFS and EASY algorithms), such as a lack of hunger, a lack of need to estimate the execution time of jobs, easy implementation, and non-migration of jobs. Another of its advantages is the possibility of estimating the inappropriate consumption of parallel processes in the CPU, as well as the loss of CPU utilisation.
- Lin et al. [55] proposed scheduling parallel jobs methods for providing software as a service (SaaS) in the cloud environments. The proposed method is called Two-Tier Strict Backfilling - slack factor (2TFB-SF). This approach relies on the flexibility of the computational power of the cloud. This algorithm is based on a two-layered architecture. In this method, for reducing the runtime of the programs, prioritizing is used.
- Liu et al. [56] presented the conservative consolidation-based first-come first-serve (CCFCFS) algorithm in the cloud computing environment. In this algorithm, each layer has particular CPU priorities. VMs with high priority are called foreground VMs. Low-priority VMs are called background VMs. Background VMs perform jobs using a small percentage of CPU's space. Jobs are prioritized based on their

entry time, and they are placed on the processors of the foreground and background layers. The jobs in the initial queue and running jobs on the processors are compared. The virtual machines of the foreground layer are assigned to high priority jobs. Jobs are moved between layers based on their priorities, and temporarily free up the processor and leave the processors. Therefore, at a time the number of jobs is high, this method has little efficiency.

- Liu et al. [57] believed that much of the computing power of the cloud goes to waste in the parallel processing of applications. With this view, the authors of the paper presented a scheduling parallel algorithm called MEASY, which uses migration and fixation to increase the effectiveness of the EASY scheduling algorithm. The results of the simulation and an evaluation of the proposed algorithm indicate that the algorithm can provide better service quality, can reduce average response time by 23.1%, and can improve the average speed reduction by 63.3% compared to the EASY algorithm.

2. Meta-heuristic Two -Layer Scheduling Algorithms

In the following the meta-heuristic two-layer algorithms for scheduling independent parallel jobs are explained.

- Komarasamy et al. [5] argued that scheduling plays an important role in processing large jobs. The parallel execution of jobs may result into inefficient CPU use due to the synchronization and communication between parallel processes. The proposed approach is the EMSA (Effective Multiphase Scheduling Approach) algorithm. In the proposed method, jobs are pre-processed to prevent starvation and reduce unwanted delay. Next, a prioritization approach is proposed that categorizes jobs according to their priority so that the number of migrations can be minimized. Finally, prioritized jobs are assigned to resources based on the Back filing algorithm. In addition, using virtualization technology, computational capacity is divided into two background and background virtual machines. Experimental results show that the performance of the proposed algorithm is better than other existing algorithms and can improve the use of CPU resources by 8%.
- Ashouraie et al. [58] examined a new class of cloud computing systems for providing users with services. The proposed method is called the Genetic algorithm - directed acyclic graph (GA-DGA) algorithm. In the proposed method, jobs are prioritized based on the genetic algorithm. To increase the efficiency of the proposed method, the genetic algorithm is combined with existing scheduling algorithms such as FIFO. The results show that the proposed algorithm has less time complexity than similar algorithms. But this algorithm is designed for heterogeneous computing environments.

IV. DISCUSSION AND OPEN ISSUES

In this section, the existing independent scheduling parallel jobs that are expressed in last section are compared with each other. As shown in Table 1 and Table 2, a detail information about existing independent non-layer and two-layer scheduling parallel jobs in term of purpose, method, results, evaluation parameters, and simulation tools are expressed respectively. The existing works are shown in the both tables are sorted in descending order based on their publishing date.

As shown in these tables, the CloudSim is the most popular simulation tool is employed by non-layer scheduling algorithms, while trace-driven and CloudSim are two popular simulation tools are used by two-layer scheduling algorithms. In addition, existing non-layer and two-layer scheduling parallel jobs algorithms employ different heuristic or meta-heuristic techniques in scheduling process to improve different parameter.

As shown in the following tables, four parameters deadline, cost, energy, and load balancing is the most popular parameters take into account in the existing works in non-layer scheduling algorithms while two-layer scheduling algorithms only focus on cost parameter. Moreover, comparison between non-layer and two-layer scheduling algorithms show that non-layer algorithms employ heuristic and meta-heuristic techniques in order to manage the scheduling queue to increase the performance or other parameters such as energy, deadline, and load balancing, while two-layer scheduling algorithms employ heuristic and meta-heuristic techniques to increase the performance of algorithms through reducing the waste time of CPU by simultaneous assigning more than one job to each physical machine.

Research on resource allocation based on the scheduling parallel of jobs is still in its early stages. Various issues are still emerging as new challenges, as described below:

- Virtual machine migration: This occurs due to the user's need to switch to another provider in order to achieve better resource. Hence, new job migration algorithms should be designedly to deal with this problem.
- Energy efficiency: Due to emergence of large data centres for saving the data in cloud computing, the amount of energy consumption significantly increased. This growth of energy consumption leads to increase the speed of the global warming. Hence, designing new energy efficient and green algorithms for cloud computing is vital.
- Accessibility**:** Cloud resource availability must be guaranteed to ensure that a user's application can run for long enough to complete the required calculations. Therefore, new techniques should be proposed to handle the lack of availability.
- Flexibility**:** In the cloud, flexibility refers to how far the scheduling of the required resources can be done dynamically. The demand for resources may increase over time, and the cloud should automatically identify and measure the size of these requests and provide the resources required.

V. CONCLUSIONS AND FUTURE WORK

Cloud computing is one of the promising technology for the cope with limitation of resources in future. Scheduling independent parallel job is a crucial problem in the cloud computing, because it has a direct impact in response time, delay, and performance of network. In this paper, at first, we classified the existing algorithms of scheduling independent parallel jobs into two categories including non-layer and two-layer. After that each class is divided into two subcategories including heuristic and meta-heuristic based on the behaviour of their solution. Finally, the existing scheduling algorithms belong to each category are explained and compared with each other based on their several features. As a future work, we will propose a new two-layer scheduling algorithm for independent parallel jobs in cloud environment.

Table 1
Overview of existing works for Non-Layer scheduling parallel Jobs

Algorithms	Purpose	The Poposed Method	Evaluation Parameter				Results			Simulation Tools
			Dead Line	Cost	Load Balance	Energy		Heuristic	Metal-Heuristic	
ATSQM (2018)	provide better performance as compared to the state-of-the-art approaches	propose a context-aware job scheduling algorithm	—	—	—	✓	propose a context-aware job scheduling algorithm	–	✓	NetBeans
HGPSO(2018)	Hybrid Genetic-Particle Swarm Optimization (HGPSO) algorithm	suitable resources for the user jobs which are in the on demand queue	—	—	✓	—	To implement genetic algorithm and particle swarm optimization algorithm are combined and used	✓	–	CloudSim
MPSO(2018)	Modified Particle Swarm Optimisation (MPSO)	Modified Particle Swarm Optimisation (MPSO) technique	—	—	✓	—	better performance than Min-Min, Max-Min, and Standard PSO	✓	–	CloudSim
ICSA(2018)	schedule from this population and then execute the proposed ICSA	propose a modulus operator and a heuristic procedure based on the pairwise exchange neighborhood	—	—	✓	—	the proposed ICSA produces better solutions than the six state-of-the-art existing algorithms	✓	–	CloudSim
MSLBL (2017)	To set the start time of parallel jobs on the processors	Choosing a processor to minimise the length of applications	—	—	✓	✓	Reduces the duration of the program	–	✓	Java
Pre-power & post-power (2017)	To optimise system performance and minimise energy consumption and deadline	Using the equal-speed method in the proposed algorithm	✓	—	—	✓	Enhances energy variability compared to previous methods	–	✓	
TaPRA-fast (2016)	To run parallel implementation jobs to reduce the completion time of a process	Using the TaPRA algorithm for scheduling parallel jobs	—	—	—	✓	Decrease completion time of jobs by 40-45% compared to other algorithms	–	✓	CloudSim
ABC (2015)	Optimising resource allocation in the cloud computing environment	Improving the bee colony algorithm	✓	—	—	—	Better performance than the GAP and NSGAII algorithms	✓	–	Or-Library
DSOS (2015)	A discrete search mechanism for optimal scheduling of jobs on resources	DSOS algorithm	—	—	—	✓	Better performance than the PSO algorithm when searching larger spaces	✓	–	CloudSim
GA (2015)	Scheduling parallel of jobs and assigning resources to a set of jobs running on processors.	Innovative genetic algorithm for scheduling parallel jobs	—	—	✓	—	Creates trade-off between cost and runtime by increasing the number of generations	✓	–	Numerical Simulations
EARES (2014)	Multi-stage utilisation of required resources during the transfer and energy reduction stages	EARES algorithm	✓	—	—	—	Optimises energy consumption compared to previous methods	–	✓	CloudSim
ABFC (2014)	To optimise the scheduling issue by creating a common relationship between jobs and node load	Cluster hybrid job scheduling optimisation strategy based on fuzzy clustering	—	✓	—	—	Better performance time of the proposed algorithm	–	✓	CloudSim
IDEA (2013)	Scheduling parallel jobs, using the improved differential evolution algorithm (IDEA)	Deployment and combination of the taquchi method and the DEA algorithm	—	—	✓	—	Makespn smaller and lower cost in parallel execution jobs	✓	–	CloudSim
PAAS (2013)	scheduling Parallel jobs in high performance computing considering energy	Scheduling jobs for smart decisions based on information in the database	✓	—	—	—	An average of 12.64% and a maximum of 13.5% in energy savings	–	✓	.NET
PACO (2013)	To reduce the cost of network communications and increase access to resources	Combining PTS algorithm and ACO algorithm	—	—	✓	—	The ability to perform jobs parallel to the min-min and GA algorithms	✓	–	CloudSim

Modified throttled (2013)	Local optimisation for the distribution of input jobs with a load balancing approach	Modified throttled algorithm	—	✓	—	—	Improved distribution of jobs with the aim of load balancing over the throttled algorithm	–	✓	CloudAnalyst
Modified max-min (2013)	To achieve best knowledge, minimum completion time and shortest response time	Improving the max-min algorithm based on the expected execution time rather than the completion time	—	—	—	✓	Improved max-min modified algorithm performance compared to other scheduling algorithms	–	✓	CloudSim
LARAC(2013)	Scheduling parallel of jobs with the goal of energy optimization	Implementing joint requests and migrating them to the cloud in a collaborative way	✓	—	—	—	Optimises energy consumption on mobile devices	–	✓	Numerical simulation
LBIMM (2013)	Scheduling of jobs with the goal of load balancing in the system	The improved load balancing algorithm is based on the min-min algorithm	—	✓	—	—	Increases user satisfaction by 20% and optimises use of resources	–	✓	Matlab
PSO (2012)	Reducing response time, increasing access to resources	PSO algorithm to optimise scheduling parallel jobs	—	—	—	✓	Generates an optimal solution with dynamic weight change	✓	–	CloudSim
Modified RR (2012)	To review the challenges and opportunities for parallel processing of data efficiently	Assigning resources to jobs with less time and ability to abort	—	—	✓	—	Higher priority in assigning resources	–	✓	Nephele
ATU(2012)	implemented faster high-priority jobs	utilizes the Turnaround time Utility	—	—	✓	—	Performance better than previous methods.	–	✓	Nephele
LBACO (2011)	Scheduling of jobs with the goal of load balancing	Job scheduling algorithm for load balance based on ACO optimised algorithm	—	✓	—	—	Better performance than FCFS and ACO algorithms	✓	–	CloudSim

Table 2
Overview of existing works for Two -Layer Scheduling parallel jobs

Algorithms	Purpose	The Poposed Method	Evaluation Parameter				Results			Simulation Tools
			Dead Line	Cost	Load Balanc	Energy		Heuristic	Metal-Heuristic	
ACFCFS (2015)	Parallel processing capabilities to reduce the use of CPU resources	Splitting the processor computational capacity into two rows, with foreground and background layers	—	—	✓	—	Better performance than the FCFS and EASY algorithms	—	✓	trace-driven
2TFB-SF(2015)	The goals are to reduce the project turn-around time and to support priority scheduling by employing suitable scheduling algorithms	propose a set of two-tier backfilling algorithms which extend the well-known conservative backfilling algorithm with project slack-time and priority concepts	—	—	✓	—	reduce the mean turn-around time of high-priority projects by more than 25%	—	✓	CloudSim
GA-DGA(2014)	proposed method, the genetic algorithm is combined with existing scheduling algorithms	jobs are prioritized based on the genetic algorithm	—	—	✓	—	the proposed algorithm has less time complexity than similar algorithms	✓	—	Expert Cloud
CCFCFS(2013)	Parallel processing capabilities to reduce the use of CPU resources	Splitting the processor computational capacity into two rows, with foreground and background layers	—	—	✓	—	Better performance than the GAP and NSGAII algorithms	—	✓	trace-driven
MEASY (2012)	Immigration and fixation to enhance the efficiency of the ESAY scheduling algorithm	Presenting the MEASY algorithm, dividing each processor into two rows: background and foreground	—	—	✓	—	Improves response time by 23.1% and mean slowdown by 63.3% compared to EASY algorithm	—	✓	trace-driven

REFERENCES

[1] H., Luo, "A Distributed Management Method Based on the Artificial Fish-Swarm Model in Cloud Computing Environment, " International Journal of Wireless Information Networks, 2018. 25(3): p. 289-295.

[2] W. and A. van Moorsel, Wongthai, "Logging System Architectures for Infrastructure as a Service Cloud," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 2017. 9(2-4): p. 35-40.

[3] S., Ab, M. Dogan, and E. Alqahtani, "A Survey On Resource Allocation In Cloud Computing, " Vol. 6. 2016.

[4] Z., Li, et al., "Bandwidth-Guaranteed Resource Allocation and Scheduling for Parallel Jobs in Cloud Data Center, " Symmetry, 2018. 10(5): p. 134.

[5] D., Komarasamy, and V. Muthuswamy, "Priority scheduling with consolidation based backfilling algorithm in cloud," World Wide Web, 2018: p. 1-19.

[6] Y., Zhu, and L. Zhou, "An Compression Technology for Effective Data on Cloud Platform," International Journal of Wireless Information Networks, 2018. 25(3): p. 340-347.

[7] H., Althumali , M. Hussin, and Z.M. Hanapi, "Cost Efficient Scheduling Through Auction Mechanism in Cloud Computing, " Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 2017. 9(2-10): p. 65-69.

[8] L.F., Bittencourt, et al., "Scheduling in distributed systems: A cloud computing perspective," Computer Science Review, 2018. 30: p. 31-54.

[9] R. Tyagi, and S.K. Gupta, "A Survey on Scheduling Algorithms for Parallel and Distributed Systems, " in Silicon Photonics & High Performance Computing. 2018, Springer. p. 51-64.

[10] Yang, J. and Q. He, "Scheduling parallel computations by work stealing: a survey. International Journal of Parallel Programming, 2018. 46(2): p. 173-197.

[11] P. Akilandeswari, and H. Srimathi, "Survey and analysis on Task scheduling in Cloud environment, " Indian Journal of Science and Technology, 2016. 9(37).

[12] Meriam, and N. Tabbane. "A Survey on Cloud Computing Scheduling Algorithms, " In: 2016 Global Summit on Computer & Information Technology (GSCIT). 2016. IEEE.

[13] E. Liu, Y., et al. "A Fuzzy-based Approach for MobilePeerDroid System Considering of Peer Communication Cost, " in International Conference on P2P, Parallel, Grid, Cloud and Internet Computing. 2018. Springer. pp 180-191

[14] Chrétienne, P. and A. Quilliot, "A polynomial algorithm for the homogeneously non-idling scheduling problem of unit-time independent jobs on identical parallel machines, " Discrete Applied Mathematics, 2018. 243: p. 132-139.

[15] P. Durgadevi, and S. Srinivasan, "Resource Allocation in Cloud Computing Using SFLA and Cuckoo Search Hybridization, " International Journal of Parallel Programming, 2018: p. 1-17.

[16] X. Zhang, , et al., "Securing elastic applications on mobile devices for cloud computing, " in Proceedings of the 2009 ACM workshop on Cloud computing security. 2009, ACM: Chicago, Illinois, USA. p. 127-134.

[17] Aggarwal, R., "Resource Provisioning and Resource Allocation in Cloud Computing Environment, " Vol. 3 , no. 3, 2018. pp. 1040–1049.

[18] L. Huang, , H.-s. Chen, and T.-t. Hu, "Survey on Resource Allocation Policy and Job Scheduling Algorithms of Cloud Computing, " 1. JSW, 2013. 8(2): p. 480-487.

[19] J.-T. Tsai,., J.-C. Fang, and J.-H. Chou, "Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm, " Computers & Operations Research, 2013. 40(12): p. 3045-3055.

[20] S. Jayanthi, "Literature review: Dynamic resource allocation mechanism in cloud computing environment," In: 2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE). 2014. IEEE.

[21] L. Shi, , Z. Zhang, and T. Robertazzi, "Energy-Aware Scheduling of Embarrassingly Parallel Jobs and Resource Allocation in Cloud, " IEEE Transactions on Parallel and Distributed Systems, 2017. 28(6): p. 1607-1620.

[22] A. Beloglazov , J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing, " Future generation computer systems, 2012. 28(5): p. 755-768.

[23] A. Beloglazov, et al., "A taxonomy and survey of energy-efficient data centers and cloud computing systems," Advances in computers, 2011. 82(2): p. 47-111.

[24] S.T. Selvi, C. Valliyammai, and V.N. Dhatchayani. "Resource allocation issues and challenges in cloud computing, " In 2014 International Conference on Recent Trends in Information Technology. 2014. IEEE.

[25] T. Meng, , et al., "A secure and cost-efficient offloading policy for Mobile Cloud Computing against timing attacks, " Pervasive and Mobile Computing, 2018. 45: p. 4-18.

[26] M.H. Mohamaddiah , et al., "A survey on resource allocation and monitoring in cloud computing. International Journal of Machine Learning and Computing, 2014. 4(1): p. 31.

[27] F. Capobianco, "5 Reasons To Care About Mobile Cloud Computing. International Free and Open Source Software Law Review, " 2010. 1(2): p. 139-142.

[28] N.R. Mohan, and E.B. Raj. "Resource Allocation Techniques in Cloud Computing--Research Challenges for Applications, " In: 2012 Fourth International Conference on Computational Intelligence and Communication Networks (CICN). 2012. IEEE.

[29] X. Liu, et al., "Scheduling parallel jobs with tentative runs and consolidation in the cloud, " Journal of Systems and Software, 2015. 104: p. 141-151.

[30] F. Villa, , E. Vallada, and L. Fanjul-Peyro, "Heuristic algorithms for the unrelated parallel machine scheduling problem with one scarce additional resource, " Expert Systems with Applications, 2018. 93: p. 28-38.

[31] A. Choudhary, , et al. "Workflow scheduling algorithms in cloud environment: A review, taxonomy, and challenges, " In: 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC). 2016. IEEE.

[32] M. Abdullahi, and M.A. Ngadi, "Symbiotic Organism Search optimization based task scheduling in cloud computing environment, " Future Generation Computer Systems, 2016. 56: p. 640-650.

[33] L. Wu, , Y.J. Wang, and C.K. Yan. "Performance comparison of energy-aware task scheduling with GA and CRO algorithms in cloud environment, " in Applied Mechanics and Materials. 2014. Trans Tech Publ.

[34] S. Durga, , S. Mohan, and J.D. Peter, "A Two-Stage Queue Model for Context-Aware Task Scheduling in Mobile Multimedia Cloud Environments, " in Advances in Big Data and Cloud Computing, 2018, Springer. p. 287-297.

[35] K. Li, , "Scheduling parallel tasks with energy and time constraints on multiple Manycore processors In A cloud computing environment, " Future Generation Computer Systems, 2017.

[36] Y. Chen, Z. Yu, and B. Li. "Clockwork: Scheduling Cloud Requests in Mobile Applications, " in 2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). 2017.

[37] F. Cao, M.M. Zhu, and C.Q. Wu. "Energy-efficient resource management for scientific workflows in clouds, " In: 2014 IEEE World Congress on Services (SERVICES). 2014. IEEE.

[38] Q. Zhang, , H. Liang, and Y. Xing, "A parallel task scheduling algorithm based on fuzzy clustering in cloud computing environment, " International Journal of Machine Learning and Computing, 2014. 4(5): p. 437.

[39] S.G. Domanal, and G.R.M. Reddy. "Load balancing in cloud computingusing modified throttled algorithm," In: 2013 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM). 2013. IEEE.

[40] H.V .Raghu, S.K. Saurav, and B.S. Bapu. "PAAS: Power Aware Algorithm for Scheduling in High Performance Computing," in 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing. 2013.

[41] U. Bhoi, and P.N. Ramanuj, "Enhanced max-min task scheduling algorithm in cloud computing," International Journal of Application or Innovation in Engineering and Management (IJAIEM), 2013. 2(4): p. 259-264.

[42] W. Zhang, Y. Wen, and D.O. Wu. "Energy-efficient scheduling policy for collaborative execution in mobile cloud computing," In 2013 Proceedings Ieee Infocom. 2013. IEEE.

[43] H. Chen, et al. "User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing," In: 2013 National Conference on Parallel Computing Technologies (PARCOMPTECH). 2013. IEEE.

[44] S. Koneru, V.R. Uddandi, and S. Kavuri, "Resource Allocation Method using Scheduling methods for Parallel Data Processing in Cloud, " International Journal of Computer Science and Information Technologies [IJCSIT], 2012. 3(4): p. 4625-4628.

[45] V.V. Kumar, and S. Palaniswami, "A dynamic resource allocation method for parallel dataprocessing in cloud computing, " Journal of computer science, 2012. 8(5).

[46] B. Fahimnia, H. Davarzani, and A. Eshragh, "Planning of complex supply chains: A performance comparison of three meta-heuristic algorithms," Computers & Operations Research, 2018. 89: p. 241-252.

[47] B. Jana, M. Chakraborty, and T. Mandal, "A Task Scheduling Technique Based on Particle Swarm Optimization Algorithm in Cloud Environment," in Soft Computing: Theories and Applications. 2019, Springer. p. 525-536.

[48] A.S. Kumar, and M. Venkatesan, "Task scheduling in a cloud computing environment using HGPSO algorithm," Cluster Computing, 2018: p. 1-7.

[49] D. Laha, and J.N. Gupta, "An Improved Cuckoo Search Algorithm for Scheduling Jobs on Identical Parallel Machines," Computers & Industrial Engineering, 2018.Vol. 126. p. 348-360

[50] T. Wen, , Z. Zhang, and M. Wang. "A Parallel Bee Colony Algorithm for Resource Allocation Application in Cloud Computing Environment, " In: 2015 IEEE International Conference on Data Science and Data Intensive Systems. 2015.

[51] P. Phuoc Hung, and E.-N. Huh, "An Adaptive Procedure for Task Scheduling Optimization in Mobile Cloud Computing," Mathematical Problems in Engineering, 2015.Vol. 2015: p. 13.

[52] R. Lin, and Q. Li. "Task scheduling algorithm based on Pre-allocation strategy in cloud computing," In: 2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA). 2016. IEEE.

[53] S. Zhan, and H. Huo, "Improved PSO-based task scheduling algorithm in cloud computing," Journal of Information & Computational Science, 2012. 9(13): p. 3821-3829.

[54] K. Li, et al. "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization," In 2011 Sixth Annual Chinagrid Conference. 2011.

[55] Y.-D. Lin, et al., "Two-tier project and job scheduling for SaaS cloud service providers," Journal of Network and Computer Applications, 2015. 52: p. 26-36.

[56] X. Liu, et al., "Priority-based consolidation of parallel workloads in the cloud. IEEE Transactions on Parallel and Distributed Systems, 2013. 24(9): p. 1874-1883.

[57] X. Liu, et al., "Scheduling Parallel Jobs Using Migration and Consolidation in the Cloud," Mathematical Problems in Engineering, 2012.Vol. 2012: p. 18.

[58] M. Ashouraie, and N. Jafari Navimipour, "Priority-based task scheduling on heterogeneous resources in the Expert Cloud," Kybernetes, 2015. 44(10): p. 1455-1471.