INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2023, VOL. 69, NO. 2, PP. 341-352
Manuscript received March 6, 2022; revised April, 2023. DOI: 10.24425/ijet.2023.144370

Resource Management Techniques in Cloud-Fog for IoT and Mobile Crowdsensing Environments

Abbas M. Ali Al-muqarm, and Naseer Ali Hussien

Abstract—**The unpredictable and huge data generation nowadays by smart devices from IoT and mobile Crowd Sensing applications like (Sensors, smartphones, Wi-Fi routers) need processing power and storage. Cloud provides these capabilities to serve organizations and customers, but when using cloud appear some limitations, the most important of these limitations are Resource Allocation and Task Scheduling. The resource allocation process is a mechanism that ensures allocation virtual machine when there are multiple applications that require various resources such as CPU and I/O memory. Whereas scheduling is the process of determining the sequence in which these tasks come and depart the resources in order to maximize efficiency. In this paper we tried to highlight the most relevant difficulties that cloud computing is now facing. We presented a comprehensive review of resource allocation and scheduling techniques to overcome these limitations. Finally, the previous techniques and strategies for allocation and scheduling have been compared in a table with their drawbacks.**

Keywords—**IoT; Fog; Cloud; Resource Allocation; Task Scheduling**

I. Introduction

THE term "Cloud Computing" indicate to the transfer of storage and processing of PCs and other intelligent devices to what is called Cloud. A Cloud server is accessed through the internet to take advantage of the services provided by these servers. Cloud Computing enables users to have data storage outside their personal computer, storing other files and any data on Cloud Computing servers where the users can access to servers from anywhere in the world when connected to the internet. Recently, global investments in the field of cloud and cloud storage have increased, this in turn greatly supports the idea of the IoT, connecting all of the world's devices to the Internet and accessing their data in the cloud. In the future, researchers want to build smart cities that are connected to the Internet and have data stored in the cloud that can be managed digitally. This technology will be employed in factories, hospitals, power plants, trains, and planes, as well as household products like lighting and televisions, air conditioners, and others [1].

Cloud Computing provides information systems infrastructure based on virtualization and provides paid services as per usage. Whereas, the cloud is a distributed system that consists of a large group of computers connected to each other and that appears to the user due to the virtual techniques applied in the computing environment Cloud is a unified and huge group of computing resources that aims to share data and computing resources for all users over the Internet. Cloud computing provides application sharing services to users over the Internet, as well as various computing services for them, while it allows to its users to reserve the resources they need to carry out some tasks for the period they want, through the Pay as you use or Pay as you go system.

The dynamic allocation of resources and the allocation of resources to applications in this way the time required to carry out tasks is reduced and the consumption of energy consumed, so that the quality of services is preserved and thus the implementation of the terms of the contract between the customer and the server Service Level Agreements (SLA). In the process of resource allocation, we are faced with a large number of tasks and resources that need to be allocated and therefore there must be a smart strategy that works to make a quick decision in the scheduling process without scanning the entire solution space because in case of scanning all the solution space, an excellent scheduling will be obtained, but the acquisition time on this scheduling would be very bad. With the increase in users of Cloud Computing services, the tasks required to be scheduled by the cloud are steadily increasing, but the main problem arises is how these servers will provide and allocate the necessary resources for each user and how will the users' work be scheduled. Therefore, there is an increasing need to find the best scheduling algorithm in the Cloud system, these algorithms must optimally schedule tasks within the constraints imposed in different environments [2]. The resource allocation process is a mechanism that ensures virtual machine allocation when there are multiple applications that require various resources such as CPU and I/O memory. There are two technological constraints in cloud: First, machinery capacity is physically limited; Second, task execution priorities must be aligned with maximizing resource efficiency. In the end, the waiting and completion time should be reduced to lower the cost of implementing the system. Traditional ways of improving a solution are time-consuming and, in some situations, unattainable. Inaccurate for addressing optimization issues and often trapped in the local optimum level [3]. The task scheduling problem was solved through the use of reinforcement learning by considering the task order, heterogeneity nodes, and grid disposition. This provides effectiveness with better implementation time [4]. M. Shojafar et al. Proposed a new technique, where they used a Genetic Algorithm (GA) with fuzzy model They represented tasks as chromosomes and genes and utilized fuzzy logic to get the suitable task. [5].

First Author is with University of Kufa, Iraq; and with Computer Technical Engineering Department, The Islamic University, Iraq (e-mail: abbas.almuqarm@gmail.com).

Second Author is with Alayen University, Iraq (e-mail: naseerali@alayen.edu.iq).

II. IoT, Fog and Cloud

This section will present the theoretical background of IoT architecture, Mobile Crowdsensing (MCS), Fog and Cloud Computing.

A. *IoT*

Every day, the number of smart devices linked to the Internet grows. The purpose of expanding the number of smart devices is to provide consumers with more convenience. According to [6], there are around 42 billion devices linked to the Internet in 2019 and 50 billion by 2020. From 2012 to 2020, the number of IoT devices linked to the Internet is shown in Figure 1.

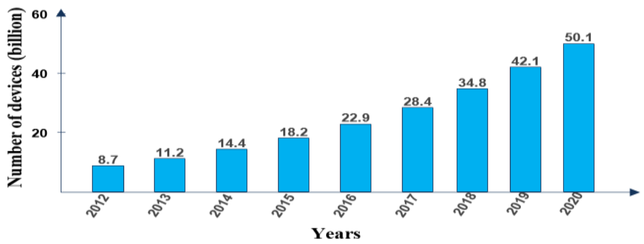

Fig. 1. The Number of Internet-Linked IoT Devices [6].

The basic architecture of IoT in its beginning consists of three layers:

Things layer, gateway, and application layer, horizontally from the bottom to the top [7-9].

Later, middleware and business layers were added as two additional layers for the three basic layers [10-12]. Figure 2 displays three and five layers.

IoT is a combination of smart-devices which communicate with each other's without direct human intervention. The devices communicate with each other and connect through a gateway which may be a router, switch, smartphone, etc. to connect to a server on the other side by using different communication technologies such as Bluetooth, ZigBee, Wi-Fi and Z-Wave which are used to save energy and cost. Non-physical protocols, on the other hand, are used to interface with the communication technologies indicated above for the same goal, such as MQTT, CoAP, etc.

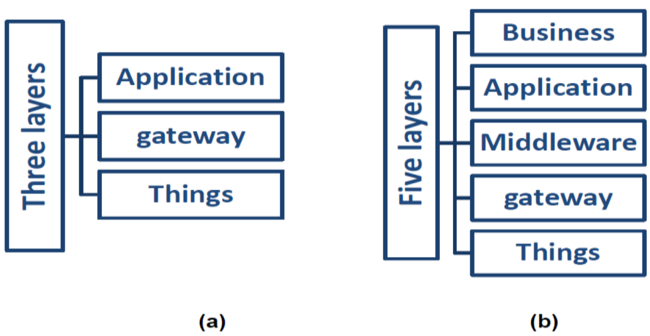

Fig. 2. IoT Layers (a- Three Layers b- Five Layers)

The IoT Architecture and layers are shown in Figure 3.

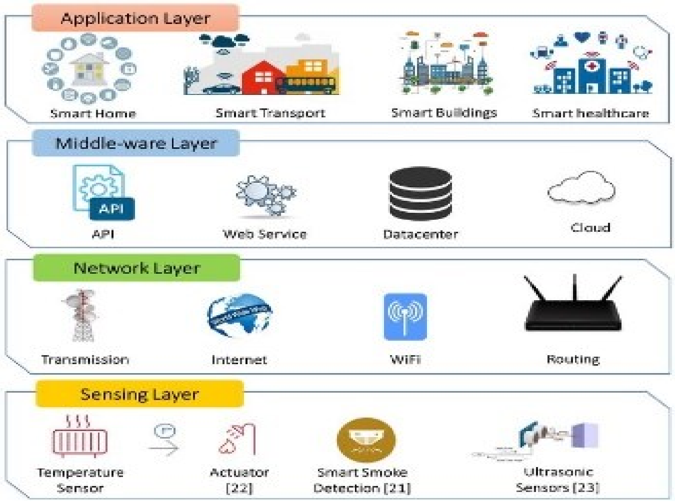

Fig. 3. Layers in IoT System.

B. *Mobile Crowdsensing Overview*

In this day, everyone begins to use technology and modern devices to make their new life easier and more comfortable. The market for mobile devices such as smartphones and other mobile devices has been spreading rapidly over the last few years, where these devices are characterized by their computing capabilities and their capabilities in communication, in addition to having embedded sensors such as compass, microphone, GPS, camera and accelerometer. MCS requires a large number of participants to perform sensing process through sensors in their phones where they can share information related to the location, share pictures, sound, environmental monitoring as well as health care and monitoring of traffic [13]. The process of sensing done by using wireless- technologies like Bluetooth, 3G-4G, LTE, and Wi-Fi to connect smart devices with each other and connect them to the Internet [14].

- MCS Stages and Resources Needed

This subsection discusses the life cycle of MCS (data sensing, computing, and data uploading) [15, 16] as in Figure 4.

.

Sensing or Data Collection	Computing and Storage Data	Data Uploading

Fig. 4. Primary Phases in MCS Life Cycle

1) *Data Sensing*

The first stage in MCS is (Sensing): It means the process of obtaining data from the environment through the sensors of smartphone; all these sensors collect data from the environment such as humidity, temperature, location and pictures where the wireless low-power communication technologies are used to connect devices to gateway (eg. Bluetooth and Wi-Fi) [16].

2) *Computing and Data Storage*

The sensors are collecting data continuously. This leads to the large size of raw data that needs processing and storage. Some sensor data are small in size (e.g. numeric values and locations) that need to use normal databases and some are very large (e.g. picture and audios) that need to use storage system such as Amazon S3. There are devices with limited resources in the computing, data storage or processing, where can be collecting data on the sensor itself before sending it to the server or send data in real time to the server. Sensors generate a big amount of data due to consuming resource of sensor such as bandwidth when transfer data and memory when data stored in local memory which waits transmission for another user [15].

3) Data Uploading

There are many sensors deployed in smart homes or smart cities such as health care, temperature, light, locks, etc., all these devices generate different data and need to store them to be processed and analyzed well. After data collection by the sensors, the last step is to upload data to the fog or cloud computing by using a suitable communication technique such as Wi-Fi or 3G connection [16]. Figure 5 illustrates the MCS paradigm.

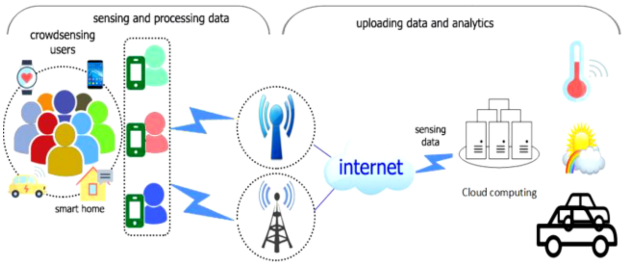

Fig. 5. MCS Paradigm

C. Cloud Computing

Cloud Computing has been clearly used after the appearance of Internet of Things, where there are a lot of widespread devices which generate a huge amount of data and that need processing, analysis and conservation as used in most applications of smart cities and smart homes applications. The data transfer to the Cloud has many negative aspects and challenges due to the limited bandwidth, direct dependence on the Internet, security threats, power consumption and delay, as some applications need fast response by Cloud as in health applications, smart grids, etc., the Fog computing is an appropriate solution to overcome most Cloud problems, where data can be processed near data sources [17, 18].

Cloud computing is classified into two types: The first type is depending on the location; The second type is depending on the type of service provided by the Cloud to users [1].

1) Cloud-Based on Location

a) Public Cloud: The whole infrastructure is located in the building for Cloud computing company that offer Cloud services for those who want to benefit from their services. In this type applications of users can be shared on Cloud servers.

b) Private Cloud: Create private Cloud Computing for exclusive use by a single person or company, where in this type there is full control over the data by the person or company in addition to being more secure.

c) Hybrid Cloud: In this type it is possible to use the public Cloud to host secondary data, but important data can be hosted on the private Cloud.

2) Cloud-Based on Service

a) Infrastructure as a service (IaaS): This type is the main category of the services of Cloud computing that allows renting servers for a fee. This includes virtual machines with high processing and storage instead of buying servers and software, customers buy these resources as their own.

b) Platform as a service (PaaS): PaaS is comparable to IaaS, but it encompasses a wider range of operating systems and services for a particular application. PaaS is a collection of software, consisting mainly of libraries, software and tools needed by developers to be used in the design or creation of web applications and mobile without concern about the storage and databases.

c) Software as a service (SaaS): It is a layer of Cloud layers that is more interested with end-user applications such as e-mail and business management systems.
Figure 6 shows the architecture of the three layers.

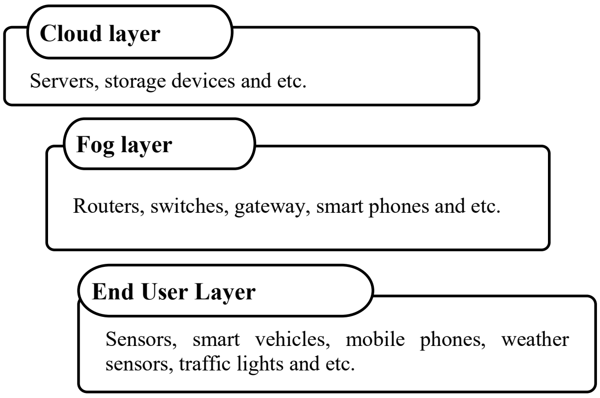

Fig. 6. The Three Layers Architecture

III. Resources Management in IoT, Fog and Cloud

This section will present the details of resources management in IoT, Fog and Cloud.

A. Resources Management in IoT

IoT systems are complicated environments because that include numerous heterogeneous components. In an IoT ecosystem, the large volume of data created by real-world sensor-equipped items will impose a great demand on processing and storage resources in order to turn it into meaningful information or services. Some applications will be latency-sensitive, while others will need complicated processing, such as historical data and time series analysis. so, the resources limitation of IoT nodes, it is difficult to imaging a large-real-world IoT system without using a cloud platform, or some strong hardware like, Smart Gateways or edge, fog nodes. The resource definition in this complicated IoT edge cloud scenario might range from physical resources like memory (storage), network bandwidth, CPU, energy, and so on to software resources. The resource management process is officially defined using a model based on the definition of the resource itself. As a result, we will choose a wide, generic, and complete definition of a resource, similar to how we approach the resource management process: a resource is every object that may be assigned inside a system [20]. Figure 7 shows the resources in an IoT ecosystem.

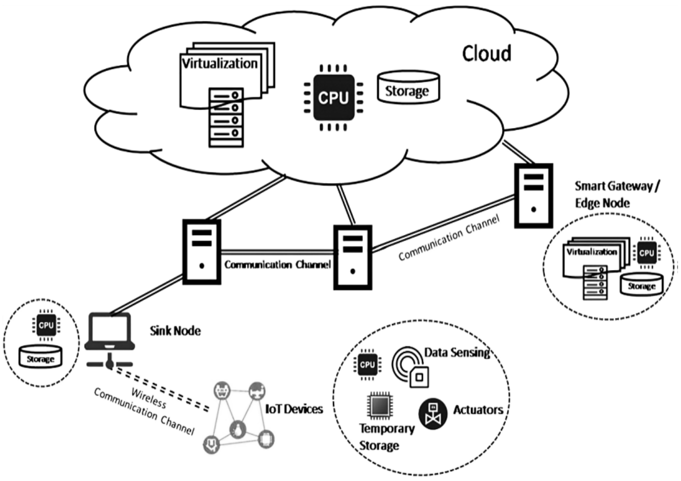

Fig. 7. IoT Ecosystem Resources Example

B. *Resources Management in Fog*

Fog computing, as previously said, is a distributed computing platform that allows IoT services to be performed locally rather than sending data to the cloud. Low-latency computing services are provided through Fog Computing at edge of the network. In regional contexts, fog-enabled network-architecture and services may efficiently exploit such resources in local to serve time-sensitive IoT applications. It minimizes backhaul-traffic transfers in addition to computing centralized demands, hence improving overall network throughput and user satisfaction. Fog, often known as "clouds at edge" is a new computer technique that places services close to devices to improve service quality (QoS). It delivers storage, processing, and networking resources in the same way as Cloud Computing does, because of the Fog environment's dynamicity, unpredictability and heterogeneity, a resource management system is required to make fog computing a reality [2].

As indicated in Figure 8, resource management techniques are divided into six categories:

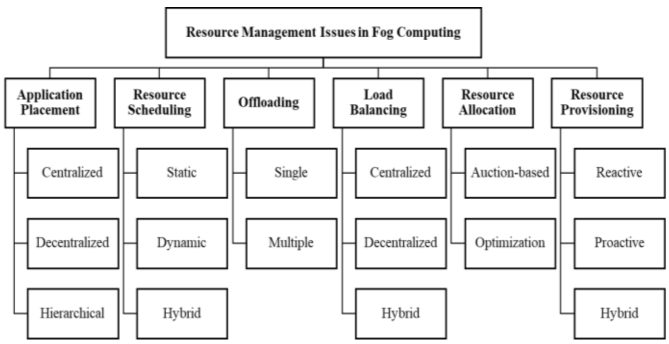

Fig. 8. Taxonomy of Fog Resource Management Methodologies

It's worth noting that the management of resources challenge in fog encompasses a number of issues and limitations that can't all be addressed as a single issue.

The first category of the taxonomy, as shown in Figure 8, placement of the application, which describes how and where applications are placed. The method for deploying applications has a direct influence on network and hardware efficiency. For example, in a decentralized fog architecture, incompatible application spread for a huge quantity of data can cause a congestion in network. One of most important categories is we need to manage resources of fog is how the application deployment and development strategy may influence how the application is put on the resource. Following application deployment, the next category in the fog resource management scenario is scheduling resources or application services. Based on the services requested by the applications, resource scheduling is determine where every fog service can run. Standard scheduling alone may not always be the best way to make the greatest use of fog resources. To accomplish so, a collaboration and offloading mechanism are used, in which duties are delegated to other active fog resources to attain the ultimate optimality in terms of ensuring expected requirements. In addition to resource usage efficiency, for time-sensitive applications, a technique that distributes incoming loads through all resources that available is required. Task scheduling applying by use load balancing, task offloading and placement of an application. To make the greatest use of all sorts of accessible resources, resource allocation is the idea of identifying a resource for services of IoT using available cloud resources, ready for fog node resources or any nearby active fog resources. Resource provisioning is the final major category in this taxonomy, which is concerned with scaling up or down available fog resources to enhancement fog resource management in terms of power, cost, response time, and other aspects [21].

Shah-M. and Wong (2018) offer an IoT system with structure of hierarchical computing and a collection of IoT users. Fog not only minimizes the volume data sent to the cloud across the network, but it also reduces latency for time-critical IoT applications in this system. It decreases the relatively long latency of remote cloud by offloading computational activities to nearby fog nodes. IoT users in close proximity can benefit from fog node computing services. Fog technology is getting more popular to the IoT applications distinct by geo-distribution, real-time, latency sensitivity and high elasticity. Optimizing the process for allocating resources, cloud/fog resource scheduling, enhancing the efficiency of resource consumption, satisfying the users' QoS needs, as well as maximizing profits of resource suppliers and consumers are all challenges related with fog computing [2].

C. *Resources Management in Cloud*

Resource management contains: resource-discovery, resource-scheduling, resource-allocation, and resource monitoring. Resource discovery mean is identifying the physical resources required to create VMs that meet user's requirements. Through resource scheduling, the better resource is selection among the matching physical resources. It actually identifies the physical resource, where the VMs are to be created to provide the resources from cloud infrastructure. Resource allocation is the process of allocating required/available resources to tasks, whereas scheduling is the process of determining the sequence in which these tasks come and depart the resources in order to maximize efficiency. There should be no overcrowding in cloud node, and all resources that available in cloud should not be wasted like (bandwidth and memory). Load balancing in cloud is an effective solution to a diversity of challenges that appear during the setup and use of a cloud system. In a distributed environment, load balancing there are two main duties to consider: resource allocation, and job scheduling [22].

1) Resource Allocation in Cloud

Allocation of resources or resource provisioning technology is a significant process of allocating resources based on the application requirements of the user to achieve the optimum number of servers. Nowadays cloud environments are heterogeneous and they have servers from different companies as consumers use a variety of resources.

There are two levels to the mapping of resources to cloud entities: [22]

a) Mapping VMs to Host

VMs residing on the host (physical servers). Multiple VMs can be mapped to a one host, based on its capabilities and availability. The host is accountable for allocating processor cores to each VM. The basis for assigning processor cores to virtual machines on-demand is defined by provisioning policies. The allocation technique or mechanism must guarantee that important host, and VM characteristics do not match.

b) Mapping Application/Task to VMs

Applications or tasks are run on the VM. To be performed, each application needs a certain level of processing power. to be done. VM should provide the required processing capability to the tasks specific to it. The tasks must be assigned to the correct virtual machine depending on its setup and availability. Tasks must be assigned to the appropriate VM depending on its configuration and availability.

Figure 9 shows where resources are allocated in Cloud systems, Figure 10 shows the categories of resource-allocation in Cloud.

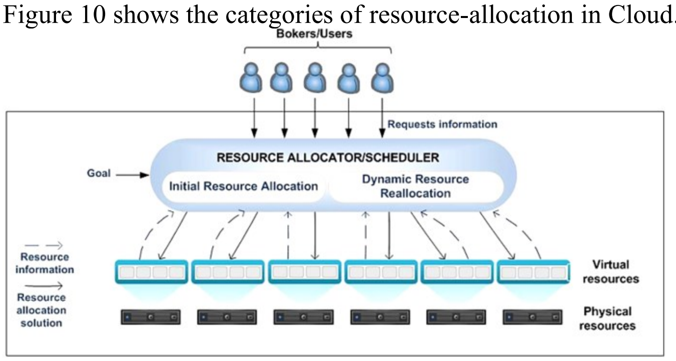

Fig. 9. Resource Allocator Level in Cloud System Strategic Based Resource Allocation

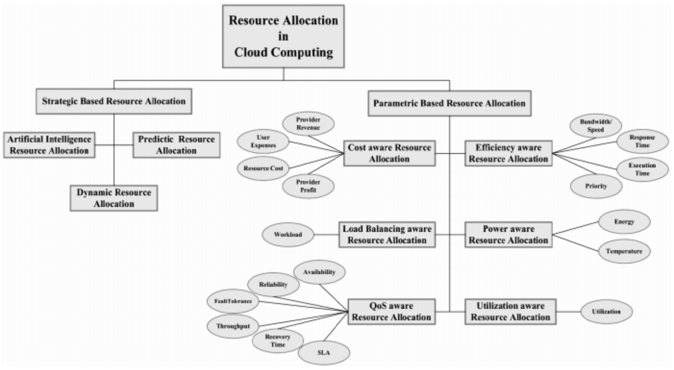

Fig. 10. Categories of Resource-Allocation in Cloud Environment [23]

2) Task Scheduling in Cloud

Task scheduling is be after allocated the resources to each and every cloud entities [22]. The efficiency of resources will be improved in cloud computing data centers. Each datacenter runs a large number of VMs (virtual machines) to maximize resource utilization. Cloud resources are sometimes underused as a result of bad scheduling task, or applications in datacenter. several algorithms schedule tasks depending on one parameter such execution time. But, in a cloud context, numerous parameters such as execution time, cost, and user bandwidth must be considered. Cloud technology is evolving at a rapid pace, and it is confronted with several issues, one of which is scheduling. The term "scheduling" indicates to a collection of policies. that a computer system uses to regulate the order in which tasks are executed. A perfect scheduler conforms to its scheduling strategy according to the type of task and changing environment [24].

There are two ways to schedule tasks: space sharing and time sharing. Users can be provided with hosts, and VMs in either space or time-shared mode. When using the space sharing option, resources are assigned until the work is completed, in this mode resources are not preempted. When using time-sharing, resources are continually preempted until the work is completed [22].

One of the essential aspects of cloud is task-scheduling. To improve system efficiency, cloud providers should use effective scheduling to decrease cost and avoid overburdening resources. In cloud user's side, the makespan and the average time it takes to complete all tasks should be decreased [25].

In cloud computing, the tasks scheduling mechanism goes through three stages: [26].

a) The first level of task includes a collection of tasks “Cloudlets” which are sent by cloud users that must be executed.
b) The second is scheduling level: is for mapping tasks to appropriate resources to obtain the best resource usage with minimum amount of makespan. The makespan is the total time it takes to complete all tasks from start to end.
c) The third VMs level: is a collection of (VMs) that are needed to execute the tasks as in Figure 11.

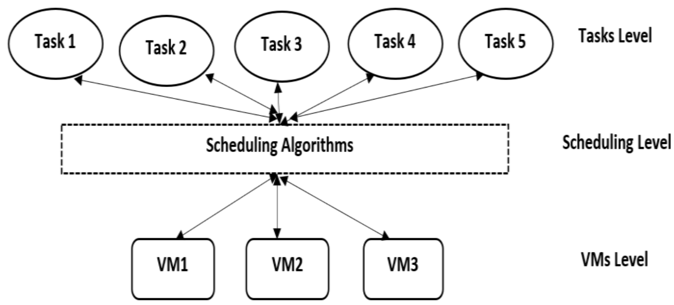

Fig. 11. System of Task Scheduling

1) Task Scheduling Approaches

Task-scheduling approaches in the cloud environment are classification as shown in Figure 12. Task scheduling approaches classify to three major environments which are single-cloud, multi-cloud, and mobile-cloud, Figure 12 shows the subclass in each one of the three environments [27].

a) Single Cloud Environment

Task scheduling used in this category in a single cloud environment. Consumers are served by a single cloud provider who offers his services. Enterprise systems can move process duties to the cloud provider to multiply the number of VMs in the single cloud environment. This class is classified into four subclasses that are energy-aware, cost-aware, QoS-aware and multi-objective.

b) Multi Cloud Environment

Different cloud providers are active in the multicloud system to obtain specific workflow-based tasks demands. To obtain the best possible performance and efficiency, the multicloud system spreads cloud applications amongst multiple cloud service providers. This class is classified into four subclasses that are multi-objective, QoS-aware and cost-aware.

c) Mobile Cloud Environment

Mobile cloud environment extends suitable services in order to provide service applications to (vehicular and mobile devices) and systems. By increasing smartphones and tablets, mobile cloud environment providing portability and elasticity to improve processing and sharing efficiency of data. This class is classified into main two subclasses that are QoS-aware task scheduling and energy-aware.

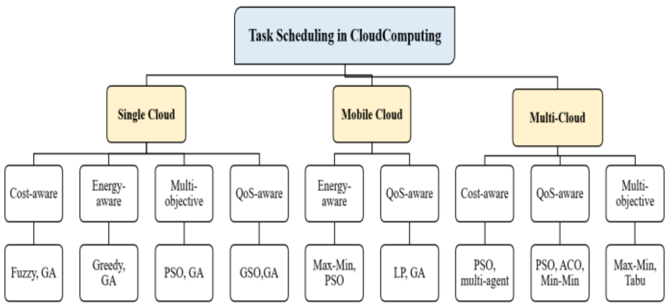

Fig. 12. The Taxonomy of Task Scheduling in Cloud

2) *Scheduling Types [28]*

a) Preemptive scheduling: It is possible to interrupt the execution of a certain task at any time, even after assigning it to the virtual machine in order to release the resource and assign it to another task.

b) Non-preemptive scheduling: When any task starts executing on a virtual machine or resource, the execution of this task will be continued until the end and will not be interrupted regardless of what happened.

c) Static scheduling: where scheduling decisions are taken based on static variables, and these decisions are taken before sending any task for execution.

d) Dynamic scheduling: where scheduling decisions are taken based on the dynamic variables that may change at any moment during the implementation, and this will increase the efficiency of the optimal use of resources.

e) Offline scheduling: where all tasks are assigned and their execution time is calculated before actually starting to implement these tasks, as the tasks are stored in a table before starting their implementation.

f) Online scheduling: Scheduling decisions are taken at the moment the task starts executing, and this may cause interruption of the execution of some tasks when new tasks enter the system.

g) Optimal scheduling: Algorithms that seek to find the optimal solution in terms of reducing costs and reducing waiting time for tasks requested by customers.

3) *Tasks Scheduling Steps (Procedure) [29]*

a) A CC user sends a task to a scheduler.

b) To receive information about resources, a scheduler connects with the Cloud Information System (CIS).

c) The scheduler received the resource information from CIS.

d) The scheduling algorithm maps tasks to appropriate resources and assigns them to the resource that is the most suited (decision process for allocating a resource).

e) The user obtains the resource's identity (id) and uses it via the cloud interface.

f) According to the schedule, the user delivers the input data to the resource.

g) To control the schedule, the scheduler receives updated information about the condition of a cloud over time.

h) The information is sent to the user.

Figure 13 shows the Tasks Scheduling Procedure.

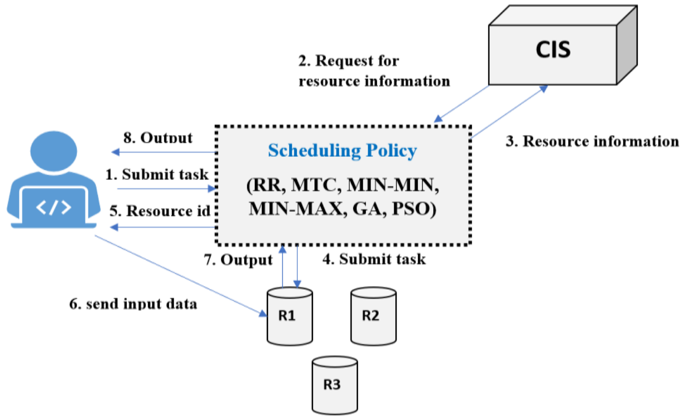

Fig.13. Task Scheduling Procedure in Cloud Computing

4) *Task Scheduling Algorithms in Cloud Computing*

In a distributed computer system, there are several different types of scheduling algorithms, and one of them is job scheduling. The primary benefit of a job scheduling algorithm is that it allows for high-performance computing and optimal system throughput. The job scheduling algorithms are First-Come-First-Serve (FCFS), Shortest-Job-First (SJF), and Round-Robin (RR) scheduling algorithms. In FCFS, the CPU is allocated in the order in which the processes arrive. It is assumed that the ready queue is managed on a first-in-first-out basis, which implies that the first job in the queue will be processed first, regardless of other considerations. SJF is a scheduling approach that chooses the job that takes the least amount of time to execution. Jobs are placed in queue according to the time of execution, the job with the shortest time to execution put first, whereas the job with longest execution time put last and given the lower priority. In this scheduling method the CPU is allotted to the process with the shortest burst time. RR is intended specially for time-sharing systems. A time slice, sometimes known as a quantum, is a tiny unit of time is defined in RR. All processes that can be executed are maintained in a circular queue. The CPU scheduler goes about this queue, allocating the CPU to every process for a time period of one quantum. New processes are inserted into the end of the queue [24].

In a Cloud context, there are three types of algorithms for task scheduling, as shown in Figure 14 [30].

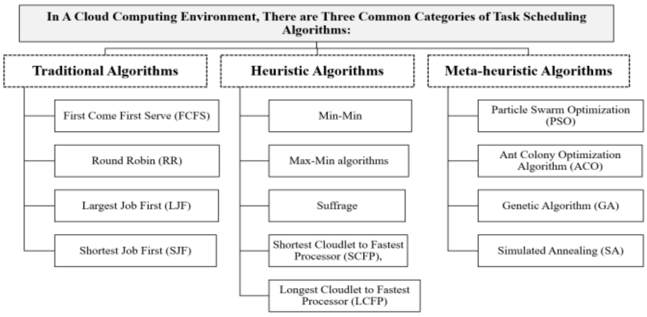

Fig. 14. Task Scheduling Algorithms

A comparison of resource allocation and task scheduling is shown in Table I [22].

Table I
Comparison between Allocation of Resources and Task Scheduling

Task	Sub-Category	Issues Resolved	Provider - Oriented	Customer- Oriented
Resource Allocation	• At host level. • At VM level.	• Ensure Availability. • Efficient Utilization. • Minimize Makespan.	Yes	Yes
Task Scheduling	• Space-sharing • Time-sharing	Decrease total response time.	No	Yes

IV. RELATED WORK

This section of paper includes various comparisons in resource management issues and technologies in different environments like IoT, Fog, and Cloud. Some benefits and drawbacks of each related study are discussed and analyzed.

Task scheduling is one of the most difficult aspects of designing cloud computing systems. It plays a critical role to get high performance and throughput by having the greatest features from the resources. This paper's key contribution is to present a novel approach called DRRHA that focuses on the classic RR algorithm's drawbacks via optimizing the metrics of performance by reducing the average of response time, waiting time, and turn-around time. But It contains a lot of calculations, in addition to their use of the FCFS and SJR algorithms [30].

This research offered a novel scheduling method for dealing with an unlimited number of incoming requests while maintaining high service quality. The allocated resource is responsible for managing incoming job requests, dispatch requests to the server, and response acknowledgment. The major goal of this research is to provide an AI approach namely GA and ANN. for allocation the resources in a cloud environment by investigating and scheduling the parallel and distributed request/response processing. A priority value is fixed depending on suitability value and jobs are prioritized and sorted in a queue and transferred to resource allocation. Resources are allocated depending on fitness value. calculating the values of attributes assigned to each coming job to optimize the Jobs. The results showed that ANN-GA is better compared to ACO of fault detection. But in this work the proposed algorithms are not handling such errors dynamically [4].

The authors of this paper introduced (SRDQ), a novel hybrid task scheduling algorithm that combined SJF and RR schedulers. The suggested methods are depending on two factors, the first of which is a dynamic task-quantum that is used to balance the time for waiting between small and large tasks. The second includes dividing ready-queue to be two sub-queues, one for short tasks and the other for long ones. Two tasks from Q1 and one from Q2 are alternately assigned to resources from Q1 or Q2. According to the results, the suggested technique improved the time of waiting, time to response, and the starvation of long jobs. But their work doesn't balance between the static/dynamic quantum values [31].

In this study, GA is used to schedule tasks in multi-cloud computing. The purpose of their study is to map the tasks to VMs in order to get maximum customer consent and the shortest time that needs to complete the final task. They first use GA to map tasks to the VMs and then schedule tasks by using the SJF algorithm. The findings reveal that the suggested method outperforms current algorithms in terms of efficiency. But they did not take into account latency arbitration, energy consumption, and running cost of a multi-cloud environment [32].

This paper presented a fog computing resource allocation technique based on priced timed Petri nets (PTPNs) to decrease the load on service provider data centers, response time, processing time, and overall cost. But they omitted fairness and correctness evaluations also did not analyze time [33].

In this paper, they introduced a Dynamic Resource Allocation (DRA) method with efficient task-scheduling, also improve the modules for power management based on PBTS, DR table updating module, and energy enhancement module. An effective resource allocation is achieved by enhanced task scheduling technique and a reduced energy consumption approach. Compared with: FCFS + Round Robin >> for Resource utilization and EERAPF + ENREAL >> for Power consumption, but they did not discuss if there were tasks having same size and arrival-time [34].

Table II summarizes the previous works on tasks allocation and scheduling in the Cloud. However, each of these methods has its own set of drawbacks and limitations.

Table II
Previous Works of Tasks Allocation and Scheduling in The Cloud

Year [Ref]	Problem	The Proposed	Technique or Algorithm	Environment	Advantage	Testing Tool	Limitation	Compared with
[35] 2010	Cost and distribution of workload into resources.	schedule tasks, and task-allocation.	Swarm Optimization (PSO) Based Heuristic	Cloud Computing	• less time consuming. • effective workload allocation onto resources.	JSwarm6 package	It does not take into account real applications.	Best Resource Selection (BRS) algorithm
[36] 2015	Mapping tasks to a Vms	presented a multiple purposes tasks scheduling	Introduce algorithm	Cloud Computing	• Better performance and improved throughput. • Improve data center efficiency, and minimize costs.	Cloud Sim	Other QoS factors can be incorporated into the suggested method to improve it.	priority scheduling and FCFS algorithm
[37] 2011	Task scheduling	presented a method called (LBACO).	ACO algorithm for load balancing.	Cloud Computing	• Equilibrated the whole system's load. • Minimizing the time takes to finish the last task.	CloudSim toolkit package	not consider for energy	FCFS, and basic ACO.
[38] 2019	Task scheduling	proposed a hybrid approach (FMPSO).	(PSO) algorithm + fuzzy system	Cloud Computing	• improved performance in terms such as: • degree of unbalance. • make-span.	CloudSim toolkit	• Task priority and load balance were not considered. • A real-world cloud environment is required to test the suggested technique.	SGA, MGA, FUGE, SPSO, and MPSO
[32] 2018	Resource Allocation and Task Scheduling	• Customer-Conscious based on Genetic Algorithm • Allocation of Resources and Scheduling the Task	• Genetic algorithm to map tasks VMs. • shortest job algorithm to schedule tasks.	Multi-Cloud Computing	• reduce makespan time. • highest level of customer satisfaction.	MATLAB	They did not consider latency, energy consumption, and the cost of running a multi-cloud environment.	GACCRATS, COTS TLBO
[39] 2017	• Resource allocation. • scheduling. • load balancing.	ABC-SA, a hybrid optimization algorithm.	Simulated Annealing (SA) + Artificial Bee Colony (ABC).	Cloud Computing	The suggested method is effective in: • Minimize makespan. • Efficiency in searching optimum resource time.	CloudSim	not tested in a (real-case study).	MFCFS, SJF, LJF, ABC_LJF, ABC_SJF
[30] 2021	Task scheduling	The novel suggested is focuses on the standard-RR drawbacks.	Enhanced Round-Robin Algorithm	Cloud Computing	optimizes the traditional RR algorithm for scheduling tasks by lowering the average of • waiting-time. • turnaround-time. • response-time.	CloudSim	It contains a lot of calculations, in addition to their use of the FCFS and SJR algorithms.	• Dynamic time slice. round-robin. • IRRVQ. • improved RR. • SRDQ.
[40] 2019	RR is suffering from some shortcomings in an average of (turnaround time, waiting time)	Proposed an approach Eighty-Five-Percentile R-R (EFPRR).	Round-Robin	Cloud Computing	Minimize average of: • waiting time. • turnaround time. • response time.	RR scheduling simulator	Algorithms need to compare using: • real-world datasets • high-volume datasets.	RR, IRR, AAIRR, ERR, ARR, RRRT, IRRVQ, AMRR and Yosef RR

[41] 2019	Process scheduling	A RR-based approach was proposed.	SJF + RR	Multiprogramming OS	Reducing average of: • Waiting time. • Number of context switches (CSs). • Turnaround time.	no	They did not use simulations and did not take many cases for experimentation.	Simple RR.
[42] 2018	Resource allocation in auctions	presented two resource allocation algorithms.	machine learning + linear regression + logistic regression.	Cloud Computing	The proposed has good effect on resource allocation in cloud.	• GNU Octave 4.2.1 • C++ • IBM CPLEX 12	did not discuss whether the resource-allocation based on machine learning satisfies the strategy proof of the auction mechanism.	greedy-based allocation algorithm G-VMPAC-II-ALLOC. Linear-ALLOC Logistic-ALLOC OPT_ALLOC
[34] 2020	Task scheduling and power minimization	A Dynamic Resource Allocation technique was presented.	PBTS + DRA Updating + Power enhancement module.	Cloud Computing	An improved resource allocation is done by improving: • task scheduling. • reduced power consumption.	no	They did not discuss if there were tasks having: same (size and inter arrival time).	• FCFS + Round Robin >> for Resource utilization • EERAPF + ENREAL >> for Power consumption
[43] 2022	Energy consumption + thermal hotspots.	Propose approach called (HUNTER) uses GGCN.	HUNTER = GGCN + Holistic/Resource Management.	Cloud Computing	Optimize five parameters: energy consumption, temperature, cost, time, and SLA violation.	•CloudSim. •COSCO.	They did not consider how to enhance the cooling management by tactics of cooling.	• PADQN. • CRUZE. • MITEC. • ANN. • SDAE-MMQ. • HDIC.
[4] 2021	Resource allocation	novel scheduling ANN-GA	ANN + GA.	Cloud Computing	• minimized fault when using intelligence techniques. • Provides less time for fault detection.	no	They could have increased the robustness of their algorithms by handling such errors dynamically.	Ant colony optimization (ACO).
[44] 2017	Latency in Smart City and IoT.	architecture for service placement was presented.	Integer Linear Programming (ILP)	Fog Computing + IoT	Good performance in throughput, delay, outage probability, and energy consumption.	Prototype	• not evaluated the power consumption. • There isn't a suitable simulation.	no
[45] 2017	communication-delay between (mobile users), and (cloud).	a heuristic algorithm with minimal complexity.	Heuristic-based	Fog Computing	• The computational complexity is low. • Cloudlet heterogeneity is taken into consideration. • Considering the user's mobility.	Simulation (NA)	• Energy is not discussed. • and not tested in a (real-case study).	Best-Fit Rand OPT
[46] 2018	data-intensive, delay-sensitive, and real-time in IoT	introduce FOGPLAN as a framework	Heuristic-based	Fog Computing	minimize the: • service latency. • computational complexity.	Java	• Energy consumption has not evaluated.	no

[47] 2018	Delay and energy consumption in IoT	presented (energy-aware) strategy.	IRR + DVFS Heuristic-based.	Fog + (IoT – Cloud)	Low in: • energy consumption. • latency.	iFogSim	• The patient's movement has not been taken into account. • High complexity in computation.	• Cloud-only. • Fog-default.
[48] 2019	challenging of put app. in Fog.	QoE-aware placement policy.	Fuzzy logic	Fog + IoT	Improves: • Time of data processing. • Network-congestion. • Resource-affordability. • Service-quality.	iFogSim	• not tested in a real case. • The approach's overhead not been explored.	MeFoRE, QoS-aware and CloudFog.
[49] 2018	Computation-offloading in fog.	present a strategy for dynamic computation offloading.	(Queuing + Game) theory.	Fog computing	Improves: • Energy. • Delay. • cost. • Group-execution cost.	Simulation (NA)	• Overhead not investigated. • Low scalability.	with schemes: SCA and LODCO.
[33] 2017	Resource scheduling	resource allocation strategy (PTPNs).	Priced Timed Petri nets (PTPNs).	Fog/cloud	Minimize: • load on datacenters. • processing-time. • overall-cost. • response-time.	Simulation (NA)	• Leaving out evaluations of (correctness) and (fairness). • time Not analyzed.	• Weight-Based Datacenter Selection. • Priority Based R-R Service Broker • Priority-Based Service-Broker

CONCLUSION

Resource allocation and scheduling are crucial features of the Cloud-based IoT ecosystem for the innovative paradigm's efficient operation. In this paper, the details presented for Resource allocation and scheduling in Mobile Crowd Sensing and discussed the drawbacks of the previous techniques that are used in allocation and scheduling tasks. We reviewed systematically all these techniques and detect the constraints and improvement of these algorithms. This paper is observed that there is a need for enhancing the performance of the cloud and we concluded that the traditional scheduling algorithms such as FCFS, SJF, and RR are impractical within a small space, so if this type of method is used to schedule tasks within a single computer, it will give excellent results, but if it is used at the cloud level, there will be big problems in time and energy because of the huge volume of possible solutions. That is why all efforts have been directed to searching for Meta-Heuristic algorithms like GA, TS, PSO, etc. that are nearly to the optimal solution and not the optimal solution, and this is what led to the emergence of Meta-Heuristic algorithms. In this study, the challenges and directions have been discussed. In the future, we plan to work on a comparison between all these algorithms in CloudSim simulator.

REFERENCES

[1] Bokhari, M. U., Makki, Q., & Tamandani, Y. K. (2018). A survey on cloud computing. In Big Data Analytics (pp. 149-164). Springer, Singapore. https://doi.org/10.1007/978-981-10-6620-7_16

[2] X. Li and L. Da Xu, "A Review of Internet of Things - Resource Allocation," IEEE Internet Things J., vol. 8, no. 11, pp. 8657–8666, 2021,. https://doi.org/10.1109/JIOT.2020.3035542

[3] S. Mousavi, A. Mosavi, A. R. Várkonyi-Kóczy, and G. Fazekas, "Dynamic resource allocation in cloud computing," Acta Polytech. Hungarica, vol. 14, no. 4, pp. 83–104, 2017, https://doi.org/10.12700/APH.14.4.2017.4.5.

[4] R. Geetha and V. Parthasarathy, "An advanced artificial intelligence technique for resource allocation by investigating and scheduling parallel-distributed request/response handling," J. Ambient Intell. Humaniz. Comput., no. 0123456789, 2020, https://doi.org/10.1007/s12652-020-02334-y.

[5] M. F. Manzoor, A. Abid, M. S. Farooq, N. A. Nawaz, and U. Farooq, "Resource allocation techniques in cloud computing: A review and future directions," Elektron. ir Elektrotechnika, vol. 26, no. 6, pp. 40–51, 2020, https://doi.org/10.5755/j01.eie.26.6.25865

[6] Burhan, M., Rehman, R., Khan, B., & Kim, B. S. (2018). IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey. Sensors, 18(9), 2796. https://doi.org/10.3390/s18092796

[7] The Intel IoT Platform, "Reference Architecture for IoT Infrastructure" Intel White paper, 2015.

[8] Mahmoud, R., Yousuf, T., Aloul, F., & Zualkernan, I. (2015, December). Internet of things (IoT) security: Current status, challenges and prospective measures. In 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST) (pp. 336-341). IEEE. https://doi.org/10.1109/ICITST.2015.7412116

[9] Atzori, L., Iera, A., Morabito, G., & Nitti, M. (2012). The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. Computer networks, 56(16), 3594-3608. https://doi.org/10.1016/j.comnet.2012.07.010

[10] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. IEEE communications surveys & tutorials, 17(4), 2347-2376 https://doi.org/10.1109/COMST.2015.2444095

[11] Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A literature review. Journal of Computer and Communications, 3(05), 164. https://doi.org/10.4236/jcc.2015.35021

[12] Sethi, P., & Sarangi, S. R. (2017). Internet of things: architectures, protocols, and applications. Journal of Electrical and Computer Engineering, 2017. https://doi.org/10.1155/2017/9324035

[13] Zhang, X., Yang, Z., Sun, W., Liu, Y., Tang, S., Xing, K., & Mao, X. (2015). Incentives for mobile crowd sensing: A survey. IEEE Communications Surveys & Tutorials, 18(1), 54-67. https://doi.org/10.1109/COMST.2015.2415528

[14] Capponi, A., Fiandrino, C., Kliazovich, D., Bouvry, P., & Giordano, S. (2017). A cost-effective distributed framework for data collection in cloud-based mobile crowd sensing architectures. IEEE Transactions on Sustainable Computing, 2(1), 3-16. https://doi.org/10.1109/TSUSC.2017.2666043

[15] Liu, J., Shen, H., Narman, H. S., Chung, W., & Lin, Z. (2018). A survey of mobile crowdsensing techniques: A critical component for the internet of things. ACM Transactions on Cyber-Physical Systems, 2(3), 1-26. https://doi.org/10.1145/3185504

[16] WANG,L. " Analyzing And Evaluating Network Protocols In Iot", thesis, Doctoral School: Computer Science, Telecommunications and Electronics of Paris, Paris, May, 2016.

[17] Yi, S., Hao, Z., Qin, Z., & Li, Q. (2015, November). Fog computing: Platform and applications. In 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb) (pp. 73-78). IEEE. https://doi.org/10.1109/HotWeb.2015.22

[18] Saharan, K. P., & Kumar, A. (2015). Fog in comparison to cloud: A survey. International Journal of Computer Applications, 122(3).

[19] Hu, P., Dhelim, S., Ning, H., & Qiu, T. (2017). Survey on fog computing: architecture, key technologies, applications and open issues. Journal of network and computer applications, 98, 27-42. https://doi.org/10.1016/j.jnca.2017.09.002

[20] Delicato, F. C., Pires, P. F., & Batista, T. (2017). Resource management for Internet of Things (Vol. 16). Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-319-54247-8

[21] M. Ghobaei-Arani, A. Souri, and A. A. Rahmanian, "Resource Management Approaches in Fog Computing: a Comprehensive Review," J. Grid Comput., vol. 18, no. 1, 2020, https://doi.org/10.1007/s10723-019-09491-1.

[22] R. Rajeshkannan and M. Aramudhan, "Comparative study of load balancing algorithms in cloud computing environment," Indian J. Sci. Technol., vol. 9, no. 20, 2016, https://doi.org/10.17485/ijst/2016/v9i20/85866.

[23] S. H. H. Madni, M. S. A. Latiff, Y. Coulibaly, and S. M. Abdulhamid, "Recent advancements in resource allocation techniques for cloud computing environment: a systematic review," Cluster Comput., vol. 20, no. 3, pp. 2489–2533, 2017, https://doi.org/10.1007/s10586-016-0684-4

[24] D. A. Agarwal and S. Jain, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment," Int. J. Comput. Trends Technol., vol. 9, no. 7, pp. 344–349, 2014, https://doi.org/10.14445/22312803/ijctt-v9p163.

[25] A. Abid, M. F. Manzoor, M. S. Farooq, U. Farooq, and M. Hussain, "Challenges and issues of resource allocation techniques in cloud computing," KSII Trans. Internet Inf. Syst., vol. 14, no. 7, pp. 2815–2839, 2020 https://doi.org/10.3837/tiis.2020.07.005

[26] T. Aladwani, "Types of Task Scheduling Algorithms in Cloud Computing Environment," Sched. Probl. - New Appl. Trends, pp. 1–12, 2020, https://doi.org/10.5772/intechopen.86873

[27] A. Amini Motlagh, A. Movaghar, and A. M. Rahmani, "Task scheduling mechanisms in cloud computing: A systematic review," Int. J. Commun. Syst., vol. 33, no. 6, pp. 1–23, 2020, https://doi.org/10.1002/dac.4302

[28] Panda, S. K., Nanda, S. S., & Bhoi, S. K. (2022). A pair-based task scheduling algorithm for cloud computing environment. Journal of King Saud University-Computer and Information Sciences, 34(1), 1434-1445. https://doi.org/10.1016/j.jksuci.2018.10.001

[29] J. Zhou, S. Bin Dong, and D. Y. Tang, "Task Scheduling Algorithm in Cloud Computing Based on Invasive Tumor Growth Optimization," Jisuanji Xuebao/Chinese J. Comput., vol. 41, no. 6, pp. 1360–1375, 2018, https://doi.org/10.11897/SP.J.1016.2018.01360

[30] Alhaidari, F., & Balharith, T. Z. (2021). Enhanced Round-Robin Algorithm in the Cloud Computing Environment for Optimal Task Scheduling. Computers, 10(5), 63. https://doi.org/10.3390/computers10050063

[31] S. Elmougy, S. Sarhan, and M. Joundy, "A novel hybrid of Shortest job first and round Robin with dynamic variable quantum time task scheduling technique," J. Cloud Comput., vol. 6, no. 1, 2017,

https://doi.org/10.1186/s13677-017-0085-0

[32] Jena T, Mohanty JR (2018) GA-based customer-conscious resource allocation and task scheduling in multi-cloud computing. Arab J Sci Eng 43:4115. https://doi.org/10.1007/s13369-017-2766-x

[33] Ni, L., Zhang, J., Jiang, C., Yan, C., Yu, K.: Resource allocation strategy in fog computing based on priced timed petri nets. IEEE Internet Things J. 4(5), 1216–1228 (2017). https://doi.org/10.1109/JIOT.2017.2709814

[34] J. Praveenchandar and A. Tamilarasi, "Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing," J. Ambient Intell. Humaniz. Comput., vol. 12, no. 3, pp. 4147–4159, 2021, https://doi.org/10.1007/s12652-020-01794-6

[35] Pandey S, Wu L, Guru SM, Buyya R (2010) A particle swarmoptimization- based heuristic for scheduling workflow applications in cloud computing environments. In: 24th IEEE international conference on advanced information networking and applications, Perth, WA, pp 400–407. https://doi.org/10.1109/AINA.2010.31

[36] Lakra AV, Yadav DK (2015) Multi-objective tasks scheduling algorithm for cloud computing throughput optimization. Procedia Comput Sci 48:107–113. https://doi.org/10.1016/j.procs.2015.04.158

[37] Li K, Xu G, Zhao G, Dong Y, Wang D (2011) Cloud task scheduling ased on load balancing ant colony optimization. In: Sixth annual chinagrid conference, Liaoning, pp 3–9. https://doi.org/10.1109/china grid.2011.17

[38] Mansouri N, Zade BMH, Javidi MM (2019) Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory. Comput Ind Eng 130:597–633. https://doi.org/10.1016/j.cie.2019.03.006

[39] Muthulakshmi B, Somasundaram K (2017) A hybrid ABC-SA based optimized scheduling and resource allocation for cloud environment. Clust Comput. https://doi.org/10.1007/s10586-017-1174-z

[40 Jbara, Y. H. (2019, April). A new improved round robin-based scheduling algorithm-a comparative analysis. In *2019 International Conference on Computer and Information Sciences (ICCIS)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICCISci.2019.8716476

[41] Srujana, R., Roopa, Y. M., & Mohan, M. D. S. K. (2019, April). Sorted round robin algorithm. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 968-971). IEEE. https://doi.org/10.1109/ICOEI.2019.8862609

[42] Zhang, J., Xie, N., Zhang, X., Yue, K., Li, W., & Kumar, D. (2018). Machine learning based resource allocation of cloud computing in auction. *Comput. Mater. Continua*, *56*(1), 123-135. https://doi.org/10.3970 /cmc.2018.03728

[43] Tuli, S., Gill, S. S., Xu, M., Garraghan, P., Bahsoon, R., Dustdar, S., ... & Jennings, N. R. (2022). HUNTER: AI based holistic resource management for sustainable cloud computing. *Journal of Systems and Software*, *184*, 111124. https://doi.org/10.1016/j.jss.2021.111124

[44] Velasquez, K., Abreu, D. P., Curado, M., & Monteiro, E. (2017). Service placement for latency reduction in the internet of things. *Annals of Telecommunications*, *72*(1), 105-115. https://doi.org/10.1007/s12243-016-0524-9

[45] Yao, H., Bai, C., Xiong, M., Zeng, D., & Fu, Z. (2017). Heterogeneous cloudlet deployment and user-cloudlet association toward cost effective fog computing. *Concurrency and Computation: Practice and Experience*, *29*(16), e3975. https://doi.org/10.1002/cpe.3975

[46] Yousefpour, A., Patil, A., Ishigaki, G., Kim, I., Wang, X., Cankaya, H. C., ... & Jue, J. P. (2018). Qos-aware dynamic fog service provisioning. *arXiv preprint arXiv:1802.00800*. https://doi.org/10.48550/arXiv.1802.00800

[47] Mahmoud, M. M., Rodrigues, J. J., Saleem, K., Al-Muhtadi, J., Kumar, N., & Korotaev, V. (2018). Towards energy-aware fog-enabled cloud of things for healthcare. *Computers & Electrical Engineering*, *67*, 58-69. https://doi.org/10.1016/j.compeleceng.2018.02.047

[48] Mahmud, R., Srirama, S. N., Ramamohanarao, K., & Buyya, R. (2019). Quality of Experience (QoE)-aware placement of applications in Fog computing environments. *Journal of Parallel and Distributed Computing*, *132*, 190-203. https://doi.org/10.1016/j.jpdc.2018.03.004

[49] Liu, L., Chang, Z., & Guo, X. (2018). Socially aware dynamic computation offloading scheme for fog computing system with energy harvesting devices. *IEEE Internet of Things Journal*, *5*(3), 1869-1879. https://doi.org/10.1109/JIOT.2018.2816682